JUNE 8, 2017

PREDATORY BIRD FEEDER

VERSION 0.3.0

ENKHAMGALAN BATERDENE, SANDEEP HEERA, GRIFFIN TOYODA
UNIVERSITY OF WASHINGTON TACOMA
1900 Commerce St, Tacoma, WA

Table of Contents

.

1. PURPOSE

This section introduces the purpose of this system specification document. It also addresses the intended audience.

This system specification document provides both a high level and low level overview of the automated bird feeder. It will describe the hardware, software, user interface design, testing of the system, and a timeline of the development of the project. The intended audience are the customer and anyone who wishes to learn more about the specifics of this project.

2. SCOPE

This section discusses the scope of this specification document. A description of the project is provided along with the objective and the environmental impact.

2.1. PRODUCT DESCRIPTION

The automated bird feeder will be the subject of this system specification document. The automated bird feeder will dispense treats to the birds of the Wild Wonders Outdoor Theater shows.

2.2. OBJECTIVE

Automate the process of dispensing treats to the birds that partake in the Wild Wonders Outdoor Theater shows. Also, allow for manual, ranged dispensing of treats by zoo staff.

2.3. BENEFITS

Improved Show Attendance due to the efficiency of how the treats are provided to the predatory birds. Automating the process of dispensing treats will improve the quality of the show by improving the entertainment value of the event. The automation of this portion of the show will also free up the biologists to interact with the audience and removes a responsibility that they have.

When project is completed and thoroughly tested and prototyped, we will have field tests that will indicate the social impact it can bring.

- Possible positive social impacts: Better performance for “Wild Wonders Outdoor” show. The device will train the show birds to perform different activities and biologists can control the activation. It will also be amusing for the viewers and attendants of the show.
- Negative social impact can be caused from product failures. There’s always a possibility that product could reach edge cases where product fails in some ways, but we have to prevent it from happening in the first place. One case is where, improper rotation of the tray could cause the bird beak to get stuck to the tray.

2.4. ENVIRONMENTAL IMPACT

Our project is composed of variety of materials that will impact environment differently.

- Environmentally hazardous materials that we need to correctly dispose are: batteries, plastic prototype casing, and microcontrollers. After proper usage and failure, the individual components should be disposed of properly.
- Items that need recycling however, but are neutral to the environment: screws, metal sheets, Lazy Susan, and wood casings.

There will be no detrimental emissions from the product that could negatively impact the environment.

3. SYSTEM OVERVIEW

This section will provide the system context. It will also provide context and background for the product.

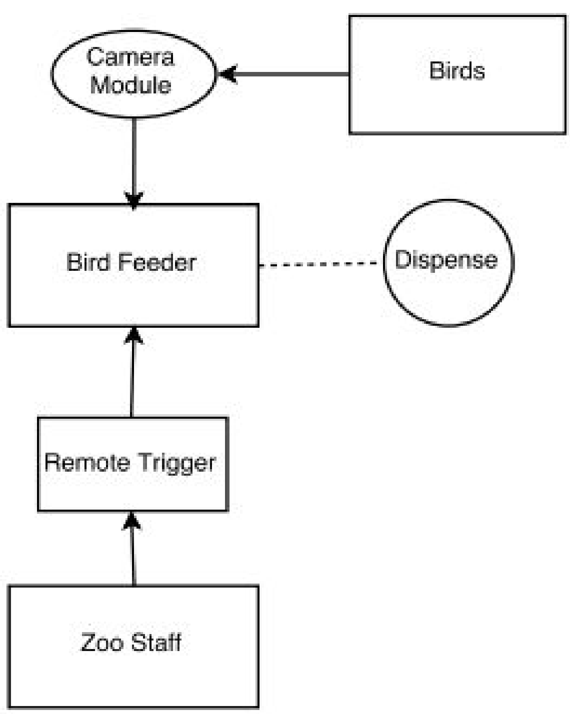
3.1. SYSTEM CONTEXT AND BACKGROUND

The bird feeder will have two modes; automatic and manual. In automatic mode, the bird feeder will determine if a valid bird is on the platform and dispense a treat corresponding to that specific bird. There will also be a remote override for this mode. Manual mode will allow the user to select which treat to disperse without the need for validation from the image recognition system.

The bird feeder will be placed on a raised platform near the rear of the Wild Wonders Outdoor Theater. The birds that are the subject of the show will fly up to the platform during the demonstrations from the stage which is 200 feet away. The bird feeder must detect if a valid bird is on the platform and if so, detect the correct treat. The zoo staff can also remotely trigger the device with a remote.

The hardware aspect of the system will consist of several microcontrollers, a camera, a stepper motor, a servo, and a metal enclosure to house the various components. The software aspect of the system consists of utilizing C (Arduino) to program the microcontrollers and Python to program the Raspberry Pi Zero W. The Raspberry Pi Zero W will execute an image recognition algorithm to determine if a valid bird is on the platform by using OpenCV (an open source computer vision platform). The Moteino microcontrollers will communicate via radio frequency signals and will operate in the 915 MHz range.

3.1.1: System Overview Diagram

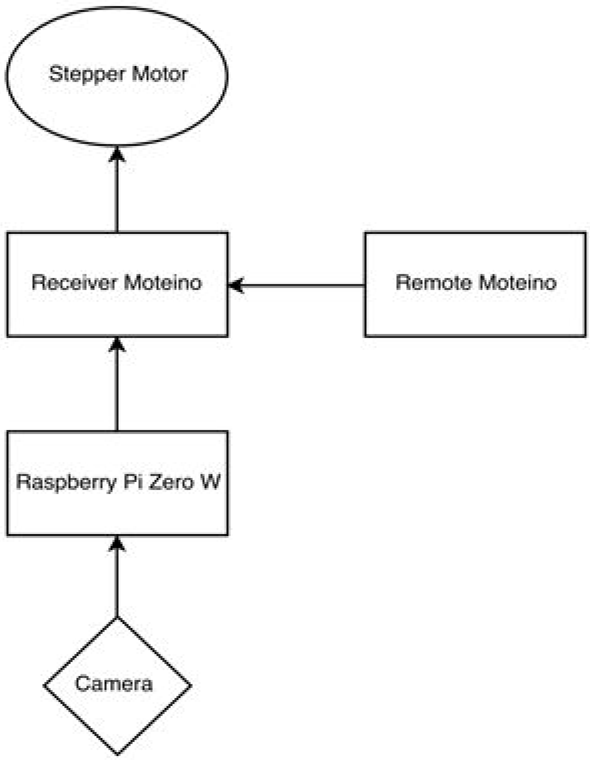


4. SYSTEM ARCHITECTURE

This section will provide both a hardware and software architecture perspective. The rationale behind the architecture and design will also be provided.

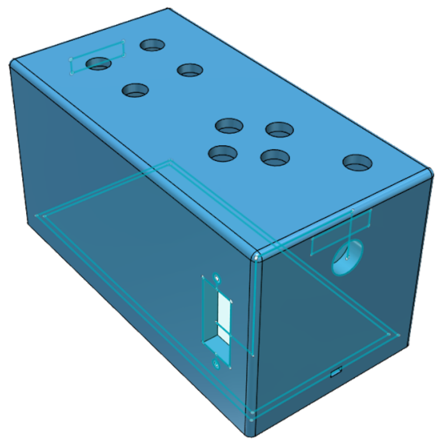
4.1 HARDWARE PERSPECTIVE

4.1.1: Hardware Overview Diagram



The remote Moteino will have four buttons that corresponds to a specific position of the dispenser, a button to open the cover to reveal the treat, a button to close the cover, and forward/backward buttons to rotate the tray one position. When a button is pressed, the remote will send a signal as a string that are unique to each command. The strings will be received by the Receiver Moteino which will determine which operation to execute. If the dispenser needs to be repositioned, the Receiver Moteino will determine which direction to turn the motor and how many steps it must take. It will then open the dispenser to allow for a treat to be retrieved.

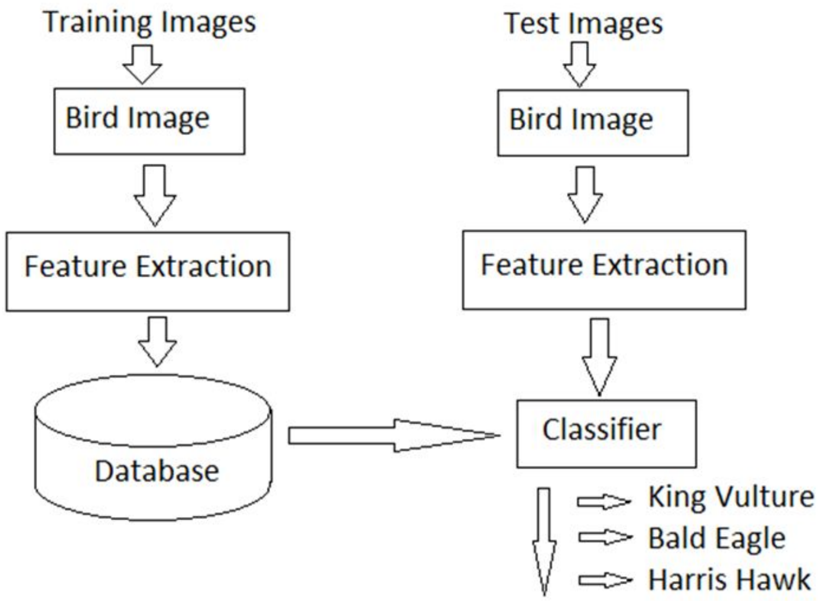
4.1.2: 3D remote casing model



The Raspberry Pi Zero W will constantly be parsing frames from a video stream to determine whether a valid bird is on the platform if the device is in automatic mode. The Pi will be directly connected to the Receiver Moteino and will use the serial interface to convey 5 possible states; don't dispense, dispense position 0, dispense position 1, dispense position 2, and dispense position 3. If the image recognition algorithm executing on the Pi determines that a valid bird is on the platform, it will output date (on the UART pins) which correspond to the unique bird.

4.2 SOFTWARE PERSPECTIVE

4.2.1: Image recognition system



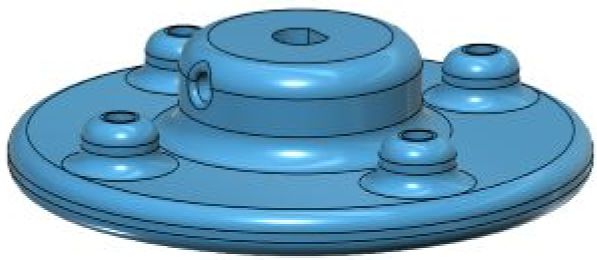
4.3. HARDWARE SPECIFICATIONS

- Moteino:
  - ATmega328p chip
  - 14 I/O pins
  - RFM69HW transceiver
- Raspberry Pi Zero W:
  - 1 GHz single core processor
  - 512 MB RAM
- Remote:
  - Composed entirely of PLA filament

4.4. HARDWARE DESIGN RATIONALE

The receiver Moteino acts as a central controller as it communicates with both the Raspberry Pi Zero W and the remote Moteino. The receiver Moteino determines if the dispenser needs to be turned and if so, the direction and step size. This centralized control will allow for both an automated state and a manual state of control. We decided to have the Pi communicate via UART with the Moteino to avoid situations where we have undefined inputs to the receiver Moteino. The serial interface is also one we are familiar with and is easy to use. The remote Moteino will communicate via radio frequency with the receiver Moteino and will act as a remote override when the device is in automatic mode. The reasoning behind providing an override is if the image recognition system fails, the operator has the option to manually adjust the dispenser.

4.4.1: Stepper Motor D Notch - Tray Actuator



4.5. SOFTWARE DESIGN RATIONALE

The open source OpenCV computer vision platform will be utilized to implement the image recognition algorithm. We have previous experience with this platform and it has a high degree of reliability. The image recognition algorithm will use Haar feature-based cascade classifiers to determine if a valid bird is present on the platform. If a valid bird is detected, the Pi will notify the receiver Moteino via the UART interface if the device is in automatic mode.

The two Moteinos will communicate using the open source RFM69 library. The Low Power library will also be used to put the two microcontrollers in a low power state when they are idle. The receiver and remote Moteinos will be nodes on a network operating in the 915 MHz band. The receiver Moteino will always be listening for the remote Moteino. The remote Moteino will simply determine if a button has been pressed, and if so, send a string corresponding to the button that was pressed to the receiver Moteino. Debouncing protocols will be used to ensure that a command does not get sent twice and an acknowledgment signal will be returned to the remote which will contain essential data relating to the current state of the device (whether it is positioning the dispenser, whether a position has already been exhausted, etc.).

5. SYSTEM DESIGN

This section will provide details regarding the system design. This includes both software and hardware design details.

5.1. CONSTRAINT ANALYSIS

- When the team completes a working prototype, we’re responsible for providing a tutorial on how we designed the project and how to operate the remote triggered food dispensing device.
- The hardware and software have been integrated, it is a cohesive design, and all the design requirements have met the safety and engineering standards.
- Provides and functional alternative for feeding the predatory flying birds.
- Must not harm any of the birds that it feeds due to functional failures.
- Must only be operable on specific scenarios where the correct bird acquires the treat thus eliminating the chance for a crow to acquire the food, instead of the show birds.
- The show birds are friendly with sudden noises and would not be put off by the servo sound or

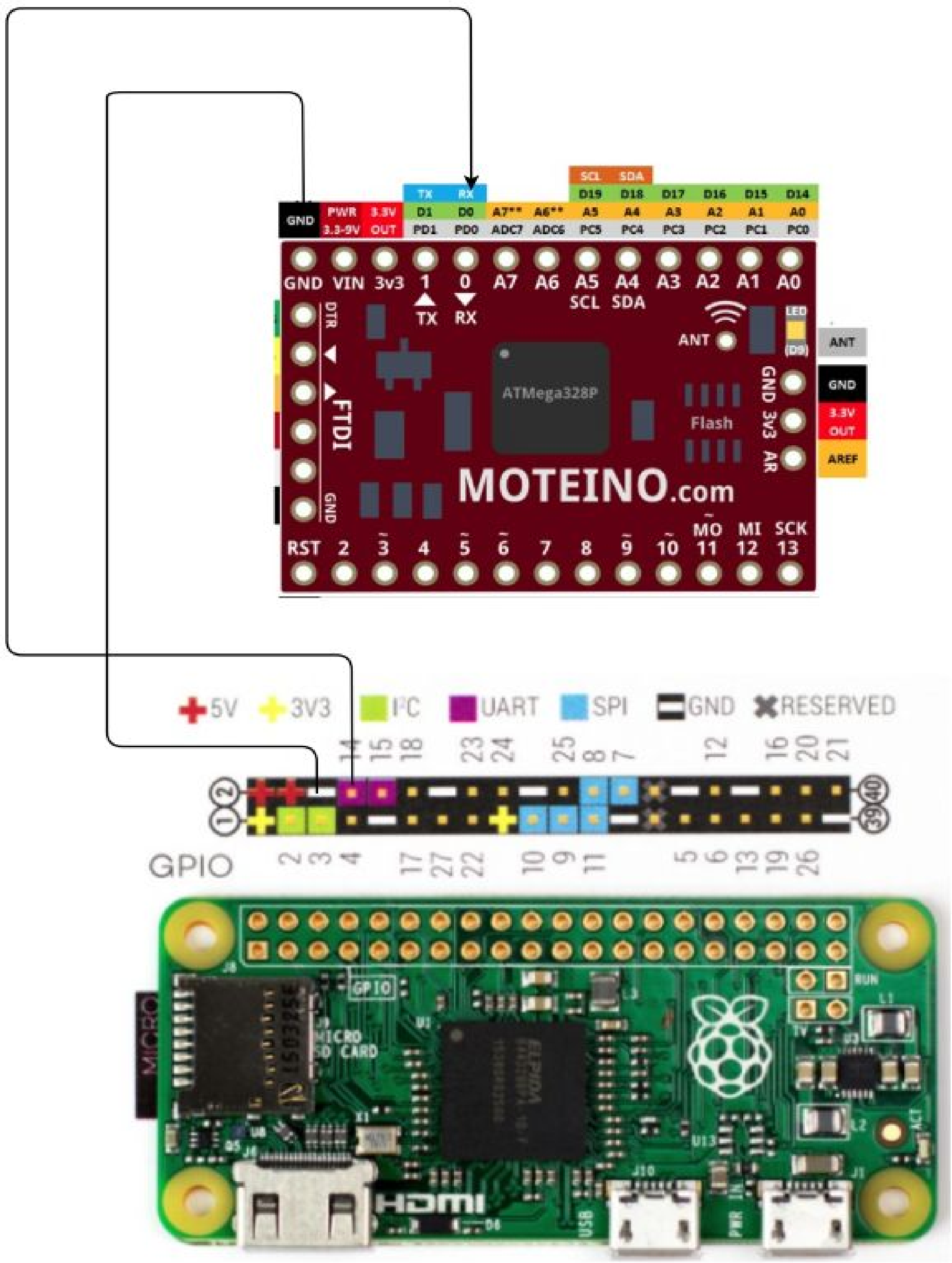
other noises.

- The provided design will be feasible and user friendly that must not distract from show.
- The given design would be portable and has detachable part for maintainability over time. The parts are easily detachable for cleaning the platform holding the lazy susan and the tray sections for food.

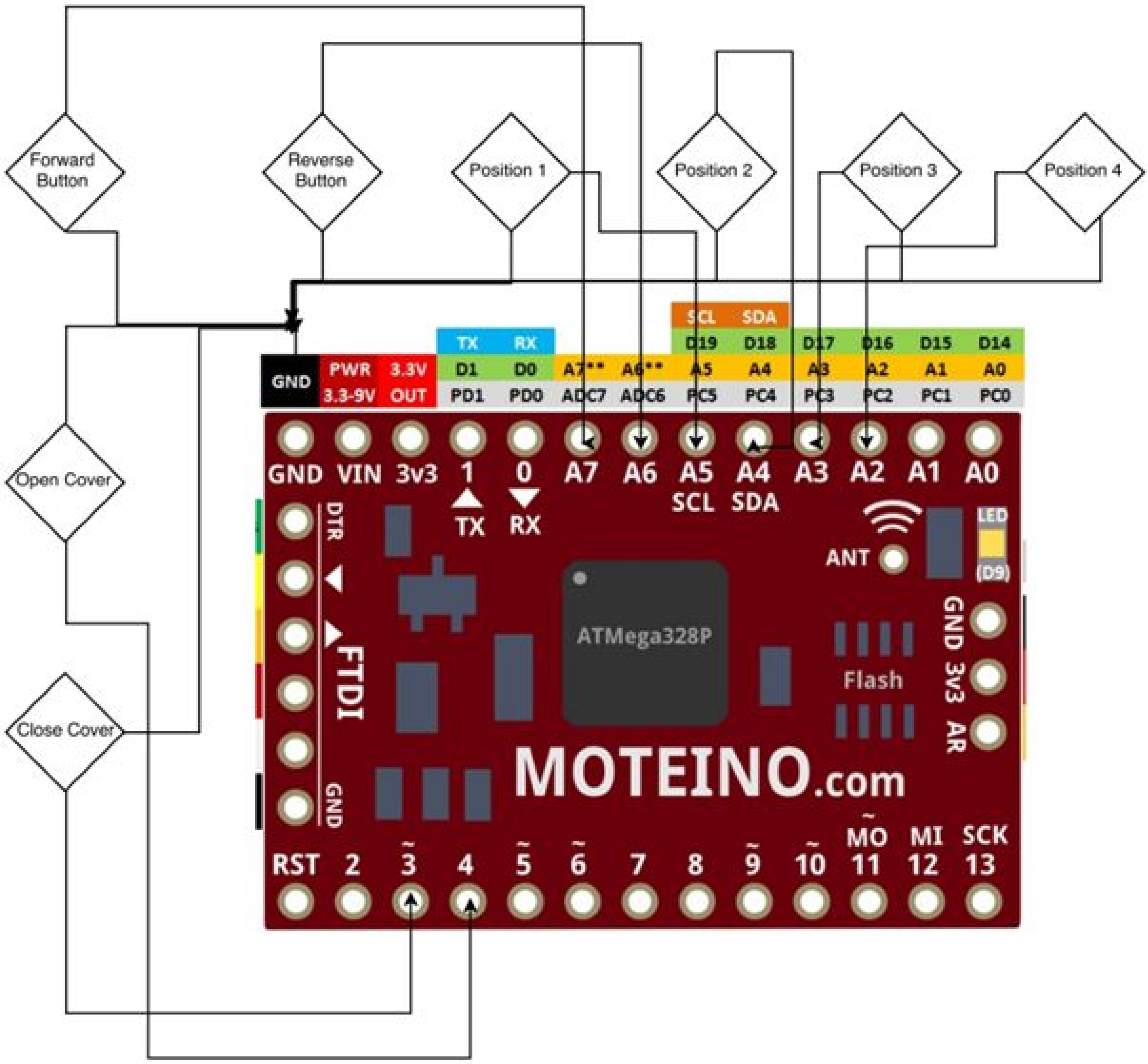
5.2. HARDWARE

The hardware aspect of the system will consist of several microcontrollers, a camera, a stepper motor, a servo, and a metal enclosure to house the various components. The software aspect of the system consists of utilizing C (Arduino) to program the microcontrollers and Python to program the Raspberry Pi Zero W. The Raspberry Pi Zero W will execute an image recognition algorithm to determine if a valid bird is on the platform by using OpenCV (an open source computer vision platform). The Moteino microcontrollers will communicate via radio frequency signals and will operate in the 915 MHz range. The electronic components of the system will be connected via a custom designed and printed PCB board.

5.2.1: Hardware interface between the Raspberry Pi Zero W and the receiver Moteino



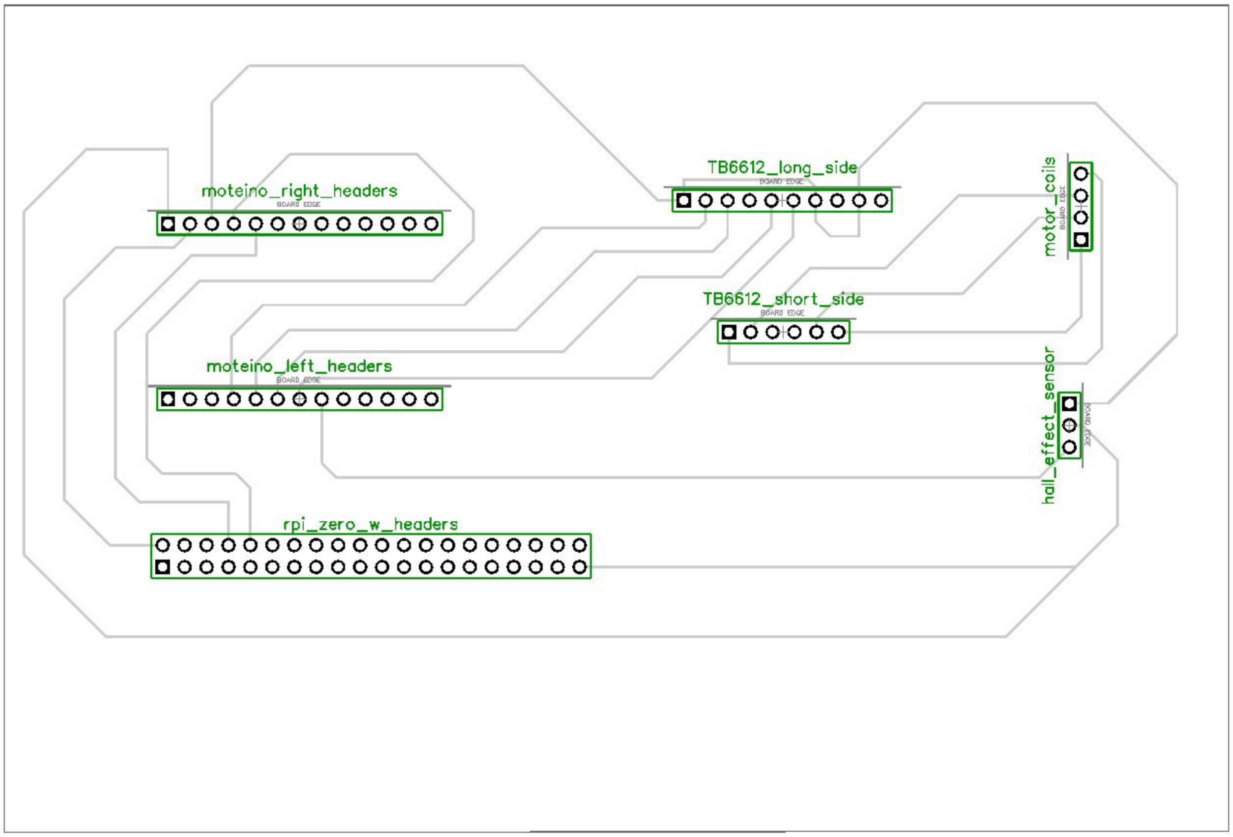
5.2.2: Hardware setup of the remote Moteino



5.3. PCB DESIGN

The PCBs were designed specifically for our system. We required 2 PCBs; one for the receiver components and one for the remote components. The PCB design for the receiver components involved ensuring that all power demands were met and that all components of the system that needed to be connected were connected.

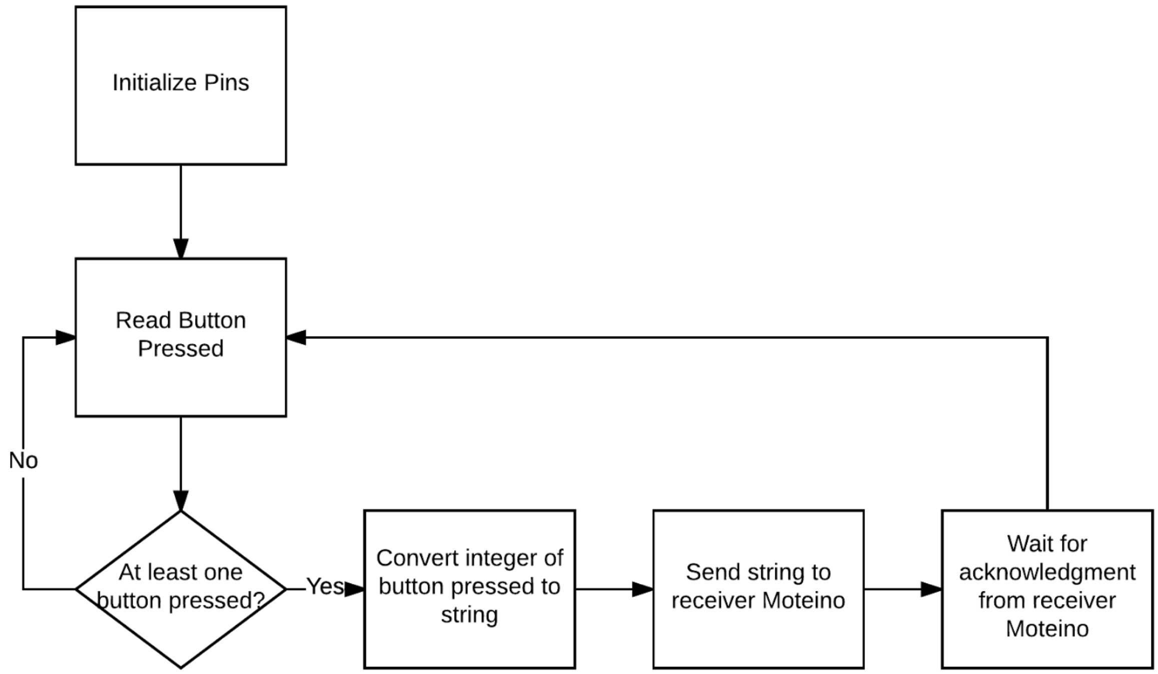
5.3.1: PCB design of the receiver components



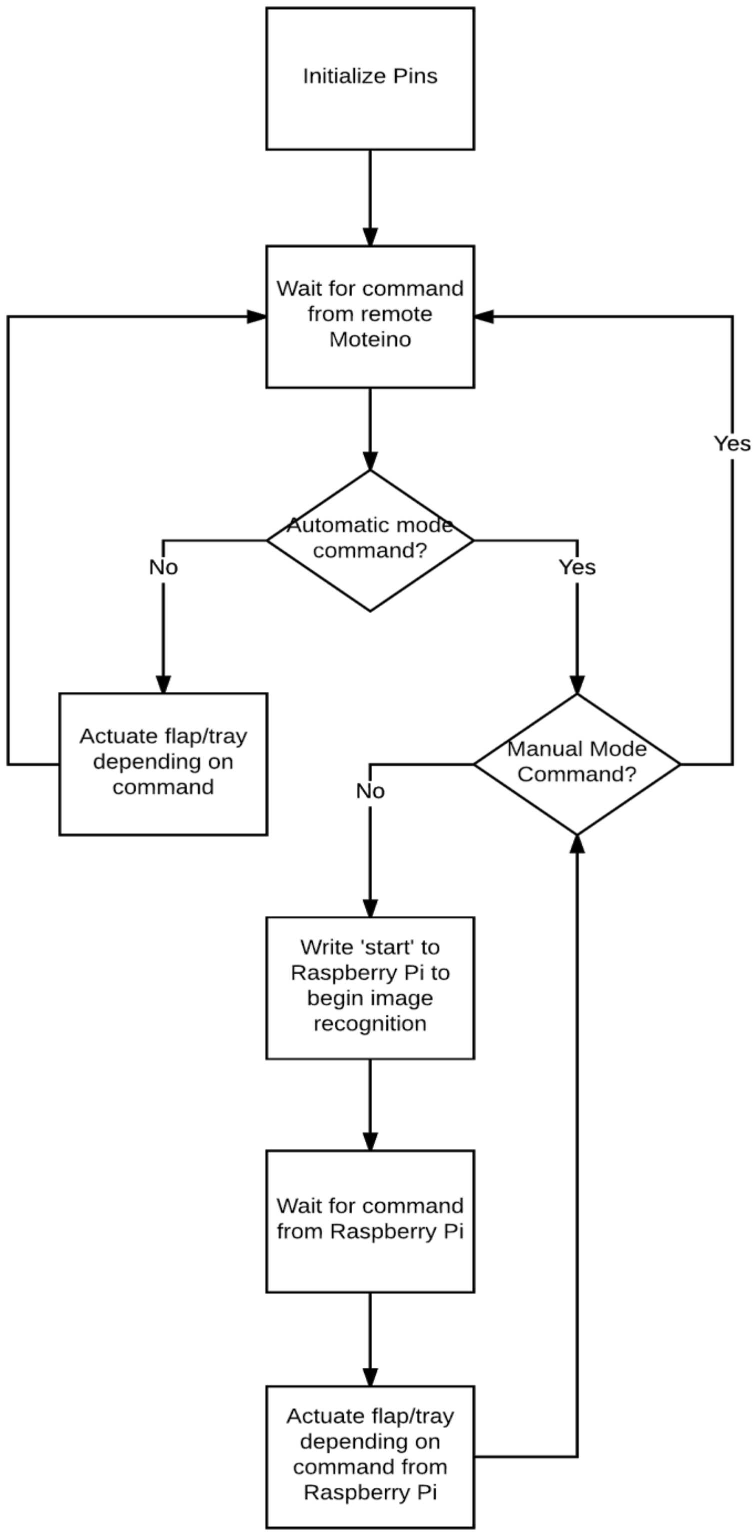
5.4. SOFTWARE

The software design of the system included the creation of Haar classifier cascades for image recognition, several distinct programs to address system functionality, and master programs to consolidate all programs.

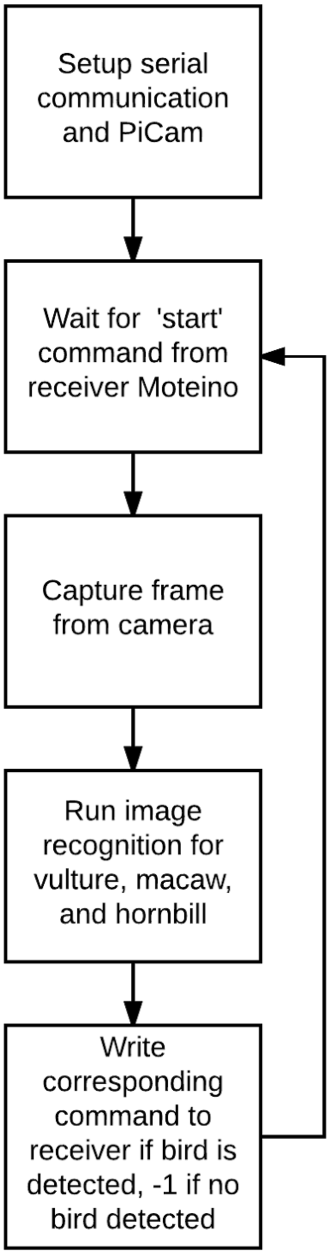
5.4.1: Flow Chart for Remote



5.4.2: Flow Chart for Receiver



5.4.3: Flow Chart for Raspberry Pi



6. HUMAN INTERFACE DESIGN

This section will discuss how this system will interact with the user. It will describe how this interaction will occur and differ between the distinct modes.

Before operating the device, the user must turn the power switch on for both the remote and the bird feeder. Once the power is on for both devices, the user will then be able to control the bird feeder via the remote. Additionally, the user must place the food for the King Volcher, Harris Hawk, Hornbill, and

Eagle Owl in compartments 0, 1, 2, and 3, respectively, for the automatic mode to dispense the proper treat.

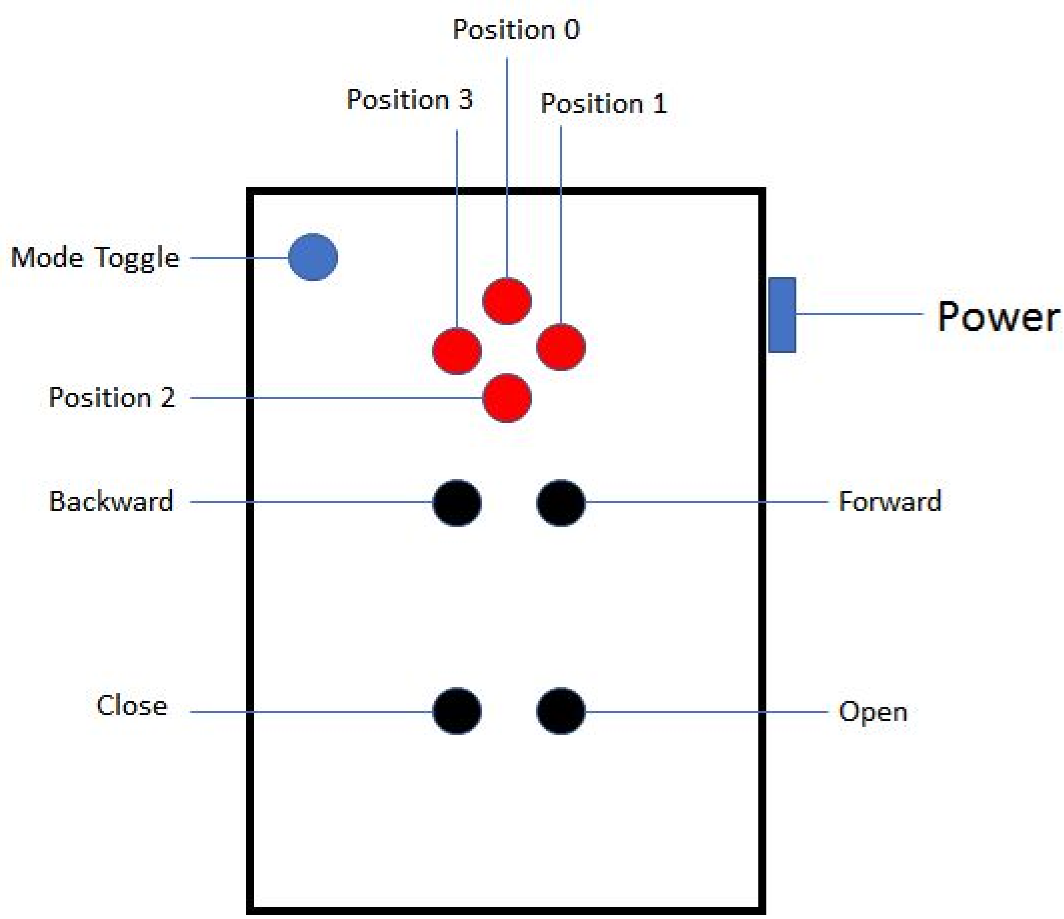
As mentioned in the scope section of this document, the bird feeder operates in two modes: automatic and manual. When the bird feeder first powers on, the compartment will be in position 0 with the flap closed, and will be operating in manual mode.

6.1. MANUAL MODE

When in manual mode, the position buttons will rotate the compartment to the position based on the diagram below. The flap position will not be changed. The “Forward” and “Backward” buttons will rotate the compartments one position forward or backward, depending on the button pressed. Again, the flap position will not change when button is activated. The “Open” and “Close” buttons will change the flap’s position corresponding to the button pressed. To switch from manual to automatic mode, the user must press the “Mode Toggle” button.

6.2. AUTOMATIC MODE

When in automatic mode, the bird feeder will operate as described previously in the scope section of this document. The default start state of this mode is at compartment 0, with the flap closed. To switch from automatic to manual mode, the user must press the “Mode Toggle” button.



6.3. MAINTENANCE

Describe changing the remote battery, bird feeder battery, removing trays for cleaning.

7. TESTING

This section will describe how the system was tested. Each individual testing procedure for all relevant components of the system will be outlined here.

7.1. MANUAL MODE

The following are 10 tables for each of the buttons on the remote. Within each table, it lists each of the initial states for the compartment/flap and the expected ending states for the compartment/flap.

7.1.1. Position 0 Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Position 0	0	Open
1	Open	Position 0	0	Open
2	Open	Position 0	0	Open
3	Open	Position 0	0	Open
0	Closed	Position 0	0	Closed
1	Closed	Position 0	0	Closed
2	Closed	Position 0	0	Closed
3	Closed	Position 0	0	Closed

7.1.2. Position 1 Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Position 1	1	Open
1	Open	Position 1	1	Open
2	Open	Position 1	1	Open
3	Open	Position 1	1	Open
0	Closed	Position 1	1	Closed
1	Closed	Position 1	1	Closed

2	Closed	Position 1	1	Closed
3	Closed	Position 1	1	Closed

7.1.3. Position 2 Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Position 2	2	Open
1	Open	Position 2	2	Open
2	Open	Position 2	2	Open
3	Open	Position 2	2	Open
0	Closed	Position 2	2	Closed
1	Closed	Position 2	2	Closed
2	Closed	Position 2	2	Closed
3	Closed	Position 2	2	Closed

7.1.4. Position 3 Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Position 3	3	Open
1	Open	Position 3	3	Open
2	Open	Position 3	3	Open
3	Open	Position 3	3	Open
0	Closed	Position 3	3	Closed
1	Closed	Position 3	3	Closed
2	Closed	Position 3	3	Closed
3	Closed	Position 3	3	Closed

7.1.5. Forward Button:

Initial Compartment	**Initial Flap Position**	**Button**	**End Compartment**	**End Flap Position**
0	Open	Forward	1	Open
1	Open	Forward	2	Open
2	Open	Forward	3	Open
3	Open	Forward	0	Open
0	Closed	Forward	1	Closed
1	Closed	Forward	2	Closed
2	Closed	Forward	3	Closed
3	Closed	Forward	0	Closed

7.1.6. Backward Button:

Initial Compartment	**Initial Flap Position**	**Button**	**End Compartment**	**End Flap Position**
0	Open	Backward	3	Open
1	Open	Backward	0	Open
2	Open	Backward	1	Open
3	Open	Backward	2	Open
0	Closed	Backward	3	Closed
1	Closed	Backward	0	Closed
2	Closed	Backward	1	Closed
3	Closed	Backward	2	Closed

7.1.7. Open Button:

Initial Compartment	**Initial Flap Position**	**Button**	**End Compartment**	**End Flap Position**
0	Open	Open	0	Open
1	Open	Open	1	Open
2	Open	Open	2	Open
3	Open	Open	3	Open
0	Closed	Open	0	Open
1	Closed	Open	1	Open
2	Closed	Open	2	Open
3	Closed	Open	3	Open

7.1.8. Close Button:

Initial Compartment	**Initial Flap Position**	**Button**	**End Compartment**	**End Flap Position**
0	Open	Close	0	Closed
1	Open	Close	1	Closed
2	Open	Close	2	Closed
3	Open	Close	3	Closed
0	Closed	Close	0	Closed
1	Closed	Close	1	Closed
2	Closed	Close	2	Closed
3	Closed	Close	3	Closed

7.1.9. Operating Mode Toggle Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Operating Mode	0	Closed
1	Open	Operating Mode	0	Closed
2	Open	Operating Mode	0	Closed
3	Open	Operating Mode	0	Closed
0	Closed	Operating Mode	0	Closed
1	Closed	Operating Mode	0	Closed
2	Closed	Operating Mode	0	Closed
3	Closed	Operating Mode	0	Closed

7.1.10. Power Toggle Button:

Initial Compartment	Initial Flap Position	Button	End Compartment	End Flap Position
0	Open	Power Toggle	0	Closed
1	Open	Power Toggle	0	Closed
2	Open	Power Toggle	0	Closed
3	Open	Power Toggle	0	Closed
0	Closed	Power Toggle	0	Closed
1	Closed	Power Toggle	0	Closed
2	Closed	Power Toggle	0	Closed
3	Closed	Power Toggle	0	Closed

7.2. AUTOMATIC MODE

Preferably we would want to test the device with the camera pointed to the actual birds. However, this is not an option, unless we get explicit approval from the zoo. In our final test, we will test the automatic mode at the zoo. In the meantime, we will print pictures of birds to put in front of the camera. The table below is the sample bird and the expected outcome:

Sample Input	Expected Output
START	Compartment 0, Flap Closed
END	Compartment 0, Flap Closed
King Vulture	Compartment 0, Flap Open
Macaw	Compartment 1, Flap Open
Hornbill	Compartment 2, Flap Open
Eagle Owl	Compartment 3, Flap Open
Crow	Compartment X, Flap Closed
Null	Compartment X, Flap Closed

7.3. TESTING OF PCB BOARDS AND CORRESPONDING DESIGNS

The procedure for testing the printed circuit boards (PCB) and their corresponding design was as follows:

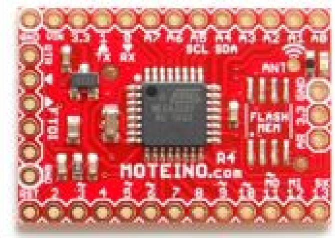
1. Ensure all traces have been properly etched via a visual inspection and improve traces as required.
2. Check each soldering point to ensure contact is made with the relevant traces and check to see if there are any undesired connections that have occurred from the soldering process.
3. Connect each individual pin to 5V and measure the pin with a multimeter to ensure that it is outputting the correct voltage
4. Check the voltage of the traces connected to the pin along with any other pins which are connected to check if they have the correct voltage.
5. Check nearby traces and pins in case an undesired connection has been made.
6. If the the pins are outputting the correct voltage (5V if it is connected to the power source and 0 otherwise), move on to step 7. Else, there is a problem with the board and a new one will be etched.
7. Connect all components and check to see if the system is operating as desired.
8. If the system is not operating as desired, there is a problem with the board or the design and both should be reevaluated.

8. TIMELINE

This section describes the timeline for the development of this project. The order of development is also given along with other relevant information.

8.1. INITIATION

The project was proposed by Point Defiance Zoo for their Outdoor Predatory Bird show. Our group had initial meeting with the Biologist and have taken the basic requirements for the device to accomplish.
Morph Chart

	Option 1	Option 2	Option 2
Remote Triggering and Wireless communication	Bluetooth – JY-MCU 	RF device – NRF24L01 	Moteino transceiver with built-in RF 
Image Recognition	Raspberry Pi 	Rpi Zero 	
Casing – with Laser Cutter	Cardboard	Wood Panel	Acryllic
Tray – With 3D printer	Uniform circular Tray	PLA Octagonal Tray	

8.2. DEVELOPMENT

Our team is following an Agile methodology with Scrum meetings where we split group responsibilities and report to each other on the assignment group meeting times about our progress and our trajectory of what needs to be implemented or accomplished by the next meeting time.

The following is a timeline for project development:

10/30/2016- Initial meeting with Point Defiance Zoo & Aquarium.

1/5/2017- Completed initial communication program between the two transceiver modules.

2/18/2017- Tested range of remote at Point Defiance Zoo & Aquarium.

2/30/2017- Implemented initial image recognition system on the Raspberry Pi Zero W.

3/12/2017- Received financial agreement from Point Defiance Zoo & Aquarium.

4/1/2017- Completed first iteration of remote.

4/2/2017- Completed first iteration of system with additional features including position selection and forward/backward capabilities.

4/13/2047- Finished System Specification Draft #1.

4/24/2017- Implemented core logic for compartmentalized dispensing on the receiver Moteino.

5/2/2017- Completed System Test Plan Draft #1.

5/4/2017- Tested prototype II at Point Defiance Zoo & Aquarium and conducted interview with media specialist from UWT.

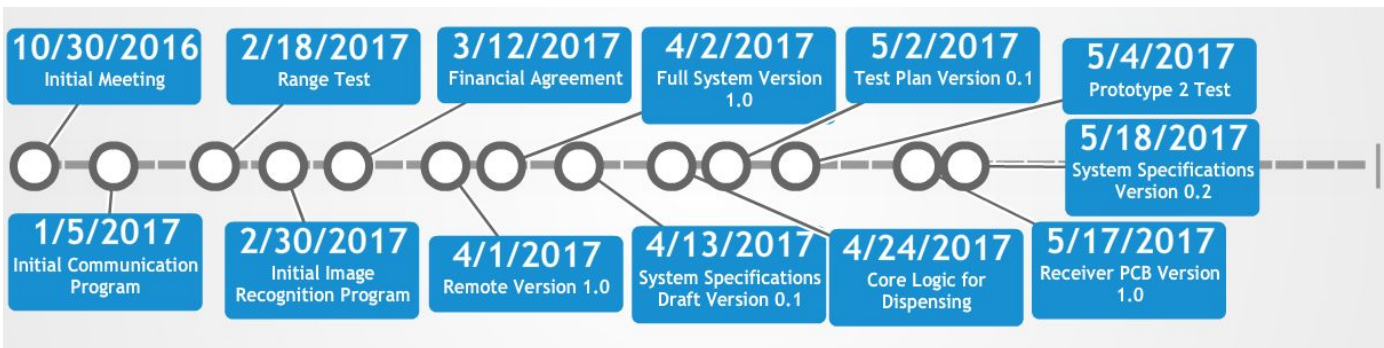
5/17/2017- Completed and printed first iteration of the PCB design for the receiver components of the system.

5/18/2017- Completed System Specification Draft #2.

6/2/2017- Presented functioning project at end of quarter showcase.

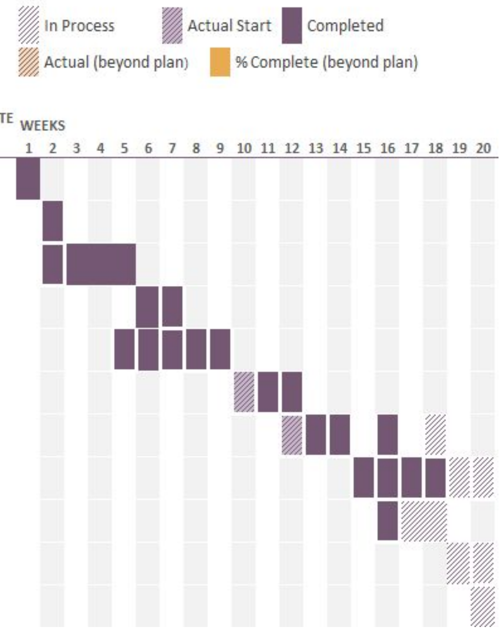
6/8/2017 Completed Final System Specification.

8.2.1: Visual diagram of project development timeline.



8.2.2: Gantt Chart for Project Development

Automated Bird Feeder



ACTIVITY	PLAN START (WEEK)	PLAN DURATION (WEEKS)	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Discuss Plans	1	1	1	1	100%
Purchase Materials (P. 1)	2	1	2	1	100%
Create Prototype I	2	4	2	4	100%
Field Test and Consultation	6	2	6	1	100%
Configure OpenCV	6	4	6	0	100%
Create Prototype II	10	3	0	0	90%
Field Test Prototype II	13	2	0	0	0%
Finalize Design	15	1	0	0	90%
Construct Final Product	16	3	0	0	85%
Test Final Product	19	1	0	0	0%
Present to PDZ&A	20	1	0	0	0%

9. RESEARCH (APPENDIX A)

9.1. REFERENCE:

Moteino image: https://lowpowerlab.com/guide/moteino/out-of-box/

Raspberry Pi Zero W image: http://www.extremeelectronics.co.uk/pi-zero-pinout/

LowPowerLab RFM69 Library: https://github.com/LowPowerLab/RFM69