BADMINTON OPEN SOURCE SYSTEM

Final Project Report

APRIL 11, 2019

ENGINEERING DESIGN – ELECENG 4OI6

Nicholas Amoirm   Dyllan Appelbohm   Allan Hieng   Mackenzie Wiens

Table of Contents

Acknowledgements

We would like to acknowledge Dr. Shirani for all his support, Gavin Crasto for his assistance 3D printing components, and the boys of 26 Haddon for letting us take over their living room during the build. Lastly, thank you to all women in the engineering, for without equal representation we will always fail in our duty to society.

Final Report

Introduction

The BOSS (Badminton Open Source System) is a badminton birdie launching system that reduces training obstacles for players of all socio-economic backgrounds. This system contains the physical launching device and a user-friendly android app that when paired together create a training environment for all! Within the app, user will have the option to select either pre-programmed launching patterns or create their own based on the type of shot, and the lateral location they wish to receive it.

Motivation

Badminton is a growing sport in Canada and is popular worldwide. A common limitation for players is the lack of consistent training scenarios. The BOSS System is an open source programmable Bluetooth device that will act as an accurate launcher. Ideally this device is an affordable option for players of all levels.

Background

In badminton training programs there are very few alternatives to private coaching. One alternative that some have tried to capitalize on are automatic feeding machines. However, these machines are not robust, difficult to purchase, and expensive. One of these machines is the patent-pending *Black Knight Trainer Pro*. This device has basic functionality and is not user-friendly. It requires the user to bring a laptop with a rudimentary desktop application. Furthermore, this device is not even available for purchase. The newest product is the *Baddy*, a badminton "robot" that you can assemble yourself or purchase ready-to-go. This device has a mobile application (although its interface is not intuitive) and does function, but it is very expensive. It cost approximately $600 CAD plus shipping (from Europe). Furthermore, this device is not open source, so any type of improvements would be very difficult to implement. Our objective was to build an open source, user-friendly, economic, automatic shuttlecock launcher. This will ultimately grow the game of badminton through providing better training opportunities to all athletes regardless of economic class.

There are six main types of badminton shots (as seen in Figure 1), the BOSS in its current iteration can hit three: smash (middle back of court), drop (middle of the court), net-drop (front of court) in 3 separation lateral directions (left, center, right) for a total of nine separate shots.

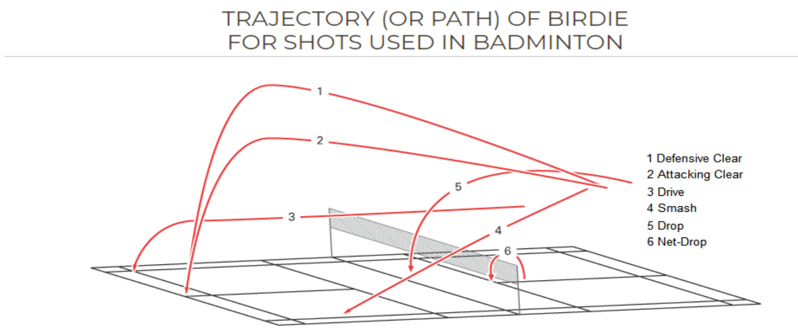


Figure 1 - Types of Badminton Shots

Project Goals & Planned Approach

Our proposed Gold Project" had the following features:

Physical Device

- Lateral movement (left, right, center)
- Linear movement (to hit back court shots)

Android App Capability

- Bluetooth connection
- Pre-programmed shot locations
- Ability to customize patterns for shots

- Battery Powered
- All STL files open source

- Open source app template
- Open source Arduino code

The device that we created included the following features:

Physical Device

- Lateral movement (left, right center)
- Piston to control back, middle, or front shots
- Only Brushless Motors Battery Controlled
- All STL and SVG files open source

Android App Capability

- Bluetooth connection
- Press shot location and device shoots
- Within the app: opportunity for pre-programmed shot locations and ability to customize patterns for shots. But not on the Arduino end.
- Open source app template
- Open source Arduino code

Milestone Descriptions	Start Date	End Date	H
M1: Build launching mechanism	October 2018	December 2018	50
M2: Prototype wheels to reduce slip	October 2018	March 2019	30
M3: connect Arduino to brushless and stepper motors	December 2018	February 2019	40
M4: Android app development	January 2019	March 2019	60
M5: Build launch tube and birdie ramp	March 2019	March 2019	20
M6: Design and testing of piston	March 2019	April 2019	20
M7: Assembly of physical device (laser cut gears, mounts, and housing)	March 2019	April, 2019	10
M8: Integrate the physical design (launching stepper motors, direction stepper motors, and brushless motors) with Arduino	April, 2019	April 7^{th,} 2019	10
M9: Integrate android app with Arduino	April 7^{th,} 2019	April 8th, 2019	5
M10: Final testing and tweaking	April 8th, 2019	April 8th, 2019	10

Table 1 - Timeline of Milestones & Hours Worked

Approach Taken

Design

Basic Technical Approach

1. Develop and build a mechanism using two brushless motors to launch a shuttlecock by applying voltage to the motors.
2. Test different wheel materials and sizes for the best results
3. Introduce secondary stepper motors to move the original mechanism left and right (lateral movement).
4. Laser cut gears to increase torque and direction for the device when moving laterally.
5. Create the wood housing for the device.
6. Design a passive feed tube to hold birdies and use stepper motors to dispense.
7. Design sliding ramp (piston) system to alter shot distance.
8. Begin Arduino code development
    a.
9. Integrate Arduino with device for controlling motor voltages.
10. Attach device to tripod; design and implement attachment.
11. Test and record ramp position based on required distance and lateral direction.
12. Android app development
    a. Sketch app flow
    b. Design background images
    c. Create homepage
    d. Create buttons that open a new activity (new page)
    e. Create an array & variable to store the int that will be sent to the Arduino
    f. Create Bluetooth protocol (open socket to connect with Arduino)
13. Integrate device with app
14. Integrate battery power into the device.
15. Fine tune the aim of device both longitudinally and laterally.

Design Process

The process of designing the device revolved around the goal to shoot one undamaged birdie at a time to consistent locations by controlling the vertical angle, horizontal angle, and power of the shot. Keeping simplicity in mind, a flywheel system was chosen as the best way to launch the birdies. Since the flywheels would damage the feathers significantly, the feeding system was designed to keep the feathers of the birdie upright so they wouldn't get caught. During testing a track was implemented to easily feed the birdie into the wheels and due to pivoting flywheels pivot point was made around the center point of the birdie so that the track could remain stationary (as seen in the below figures). Lastly, two interlocking gears were used to turn the flywheels left and right so the stepper motors did not have to be powerful and to reduce the weight that would be directly resting on one motor.

At this point in the design process the ability to vary the vertical angle of the shot was abounded due to the difficulty it would take to pivot the entire machine. The BOSS was mounted at a constant angle of 12 degrees using a tripod to get the most distance out of the shots.

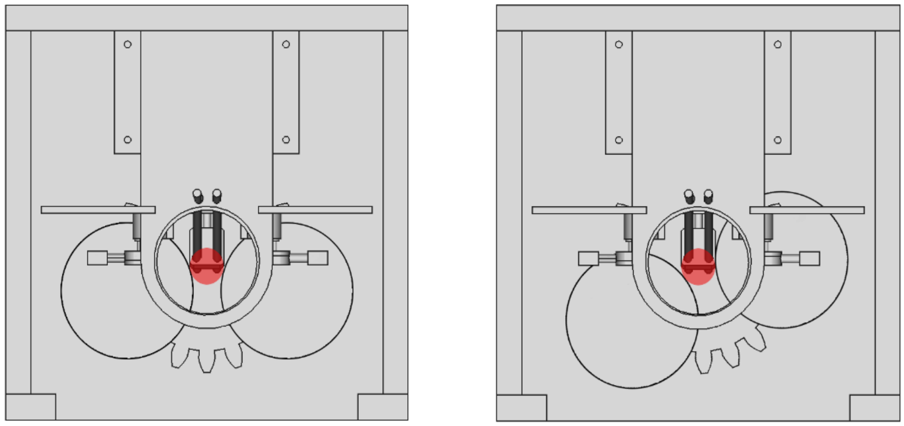


Figure 2 - The track remains stationary while the flywheels rotate around the center point of the birdie's starting position

The next challenge was moving the birdie from the storage tube to the flywheels. A 3-inch mailing tube and the elastic part of a sock was used to hold back the birdies before changing the sock for a foam gasket. The supports for the tube and the dropper were designed to fit together nicely to add additional stability to the dropper arms. The arms were tipped with the same foam material that the tube gasket was made from due to its soft and springy properties.The last challenge was to come up with a way to vary the distance that the birdie was shot. Varying the speed of the brushless motors did not change the distance of the shot significantly since the brushless motors were too weak to deal with the resistance of the birdie. A solution was found while manually feeding birdies through the flywheels, when the cork on the birdy didn't line up properly with the wheels the birdie didn't travel as far. A small piston was designed to adjust the distance between the track and the wheels which would allow us to control how much of the cork would be gripped by the flywheels. This allowed the BOSS to shoot net shots as well as the farthest shots.

Colour	$U_0\ (ms^{-1})$	θ_0 (°)
Blue	19.8	39
Green	24.7	44
Cyan	6.8	18
Yellow	9.7	44
Light Purple	9.5	30
Gray	9.6	18
Black	13.4	58
Red	37.6	38
Dark Purple	32.3	12

Table 2 - Initial Conditions for Figure 4

Figure 3 - BOSS mounted on a tripod at 12 degrees (the optimal angle for trajectory)

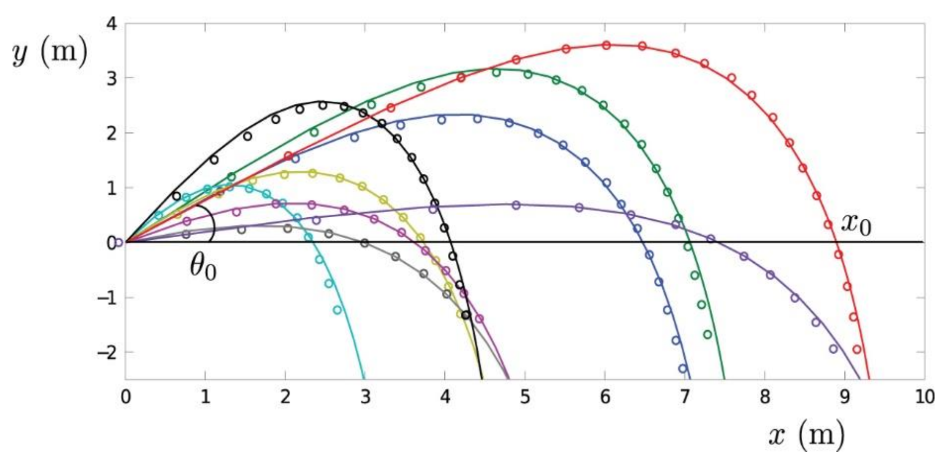
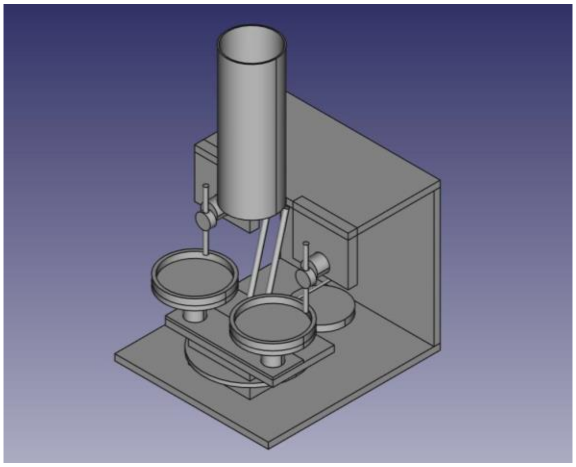
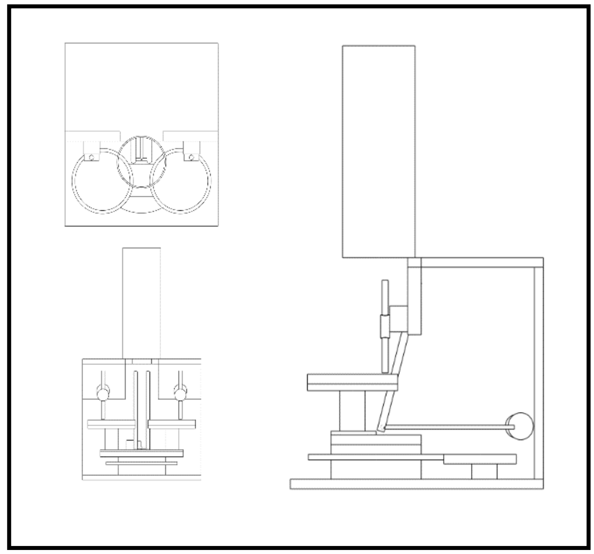


Figure 4 - Comparison between the observed trajectories (circles) and trajectories calculated with a pure drag equation and (solid line) for different initial conditions.

Modeling & Simulation

Device Drawing





Laser Cut Parts

- Pulldown motor hub (2)
- Flywheels (2)
- Piston motor hub
- Feed ramp connector

3D Printed Parts

- Main & Secondary gears
- Brushless motor plate
- Lazy Susan bearing spacer
- Piston motor shelf
- Tube mounting plate
- Pull down motor mounting plate

Implementation

Physical Device

The BOSS device contains five stepper motors controlled by an Arduino which is controlled via a Bluetooth connection from an Android App. Two stepper motors are used to pull one birdie at a time from the gravity tube feeder and load them onto the track. The track position (lateral direction) and launch (linear) distance is determined by two other stepper motors. One of the steppers turns a gear which controls the lateral direction and the other stepper motor is the piston mechanism. Brushless motors are moved using a servomotor connected to a The Arduino is connected to an electronic speed controller to spin two brushless motors which are connected to flywheels that launch the birdies. When the birdie slides down the track, it is launched when it contacts the rotating flywheels

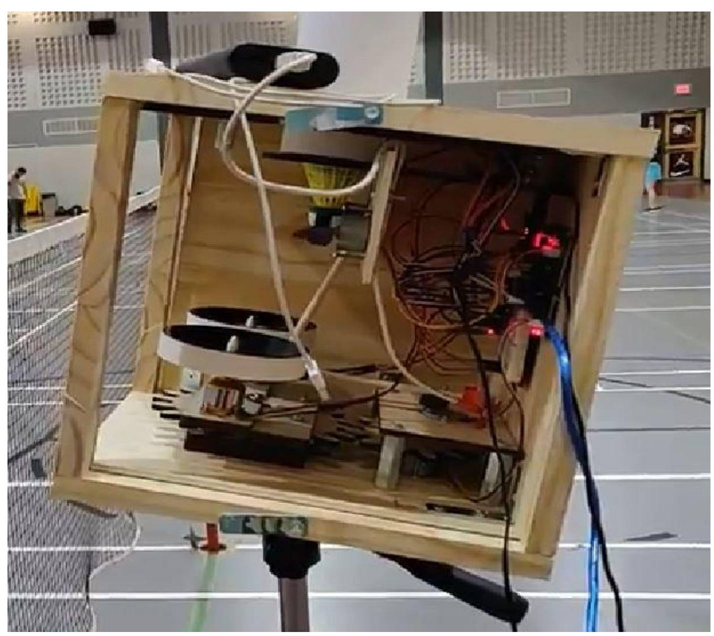


Figure 5 - BOSS Use Flowchart

Arduino

The Arduino is main controller of the entire device. It connects the Android app to the hardware using a Bluetooth module, the HC-06. It controls the 4 stepper motors, and the brushless motors. It provides power for the piston and bearing stepper motors. It does not provide power for the feeder stepper motors as they require more amperage to do their job. It provides the brushless motors with a square wave that is generated through pin 3, which controls the speed of the brushless motors. The AccelStepper and MultiStepper libraries were used, since they abstracted some controls of the stepper motors. These libraries were to used to control the speed and position of the various components.

The Arduino accepts Bluetooth inputs with a baud rate of 9600 since speed is not a constraint and very little data is sent at once. Once it receives data from the HC-06, a protocol designed is used to control the next actions. An integer is received, and a case statements determines the actions based on Table 5. it is ALWAYS required to press reset before powering off the device. Because it moves the device to the 0 position and all the other positions are based of this location being 0.To move each stepper motor, a function stepper.moveTo(num) was used. Where num is an integer value relative to its start position which is 0. Moving clockwise requires a positive integer, while the opposite is true for a negative number. Since the steppers used are 8 steps, the following values are used to move to key positions. Except for the feeder motors, the other motors used some values in between the key values.

The brushless motors are powered separately as they take a burst of current to run, which the Arduino cannot handle. However, the speed can be controlled by the Arduino. Since they are controlled by the duty cycle % of the PWM. Using analogWrite(pwm_pin, duty_cycle), the % duty cycle can be controlled. Anything above 60% turns the motors on. For 100% duty cycle, the value of 255 is used with 0 being 0%. 149(58%) was used for off, while 160(63%) was used was as on. These values were found by testing as the PWM of the Arduino had some noise.

Integer Value	Action
-450	Turn bearing right
450	Turn bearing left
0	Turn bearing straight & move piston to neutral far position
1500	Move piston to middle position
2600	Move piston to close position
1600	Move piston to close position for L&R

Table 4 - Stepper Motor (Bearing & Piston) Positions

Integer Value (+/-)	Position
4096	1 REV
4096/2 = 2048	½ REV
4096/4 = 1024	¼ REV

Table 5 - Key Stepper Motor Positions

Serial Data Read	Action
1	Close left shot
2	Close right shot
3	Close center shot
4	Middle left shot
5	Middle right shot
6	Middle center shot
7	Far left shot
8	Far right shot
9	Far center shot
10	Reset to natural (middle, center)
11	Turn off brushless motors
12	Turn on brushless motors

Table 3 - Arduino Protocol Table

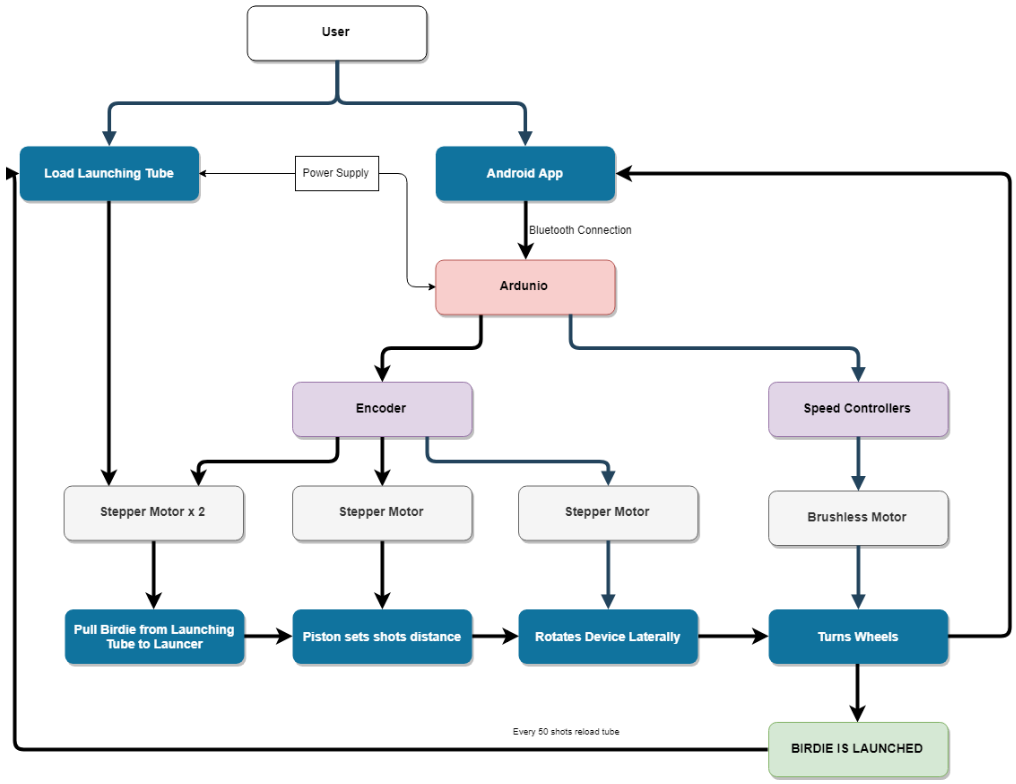


Figure 6 - System Use Flowchart

Android App

During the development of the physical device, a testing app was also developed. This testing app was primarily used to test the Bluetooth connection and basic functionality of the device while being the backbone of the final app. The testing app was very basic as it only featured numerous buttons to test parts of the device such as turning the bearing, adjusting the piston, and turning on and off the brushless motors. To establish connectivity, Bluetooth socket was created. Since a HC-06 was used, it can only accept connections. The socket was created with a generic UUID, which is not an issue if the device does not connect to 2 devices at once with the same UUID and the name HC-06.

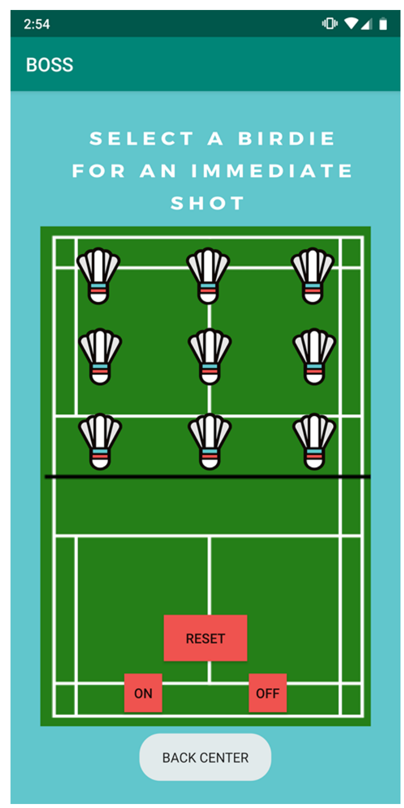


Figure 7 - Interactive Shooting Page; with Toast Message of which button clicked

This approach to the socket will continue to the final version of the app. The app in its current iteration features a homepage and the option to play three separate ways: (1) via a custom shooting pattern chosen by the user, (2) pre-chosen shooting patterns, or (3) an “immediate shot” that when the player selects the location the birdie is shot there. The app is set up for these 3 options, but the Arduino connection only interfaces with the option (3). When the user opens the app and selects “Immediate Shots”, a Bluetooth socket opens (the same type of socket that was used during testing) and the app begins to listen for a button click. The user must first press the “On” button to turn on the brushless motors and then can begin selecting locations. When a location is selected, an integer is sent to the Arduino (based on table 5).

Critical Problems Solved

Launching Mechanism

Launching the birdie to the back of the court without damaging any of its feathers has been the largest issue we faced and where most of our solutioning has come from. To solve this problem the following steps were taken:

1. Brushless motors with flywheels were chosen as the mechanism
2. Shooting tray was utilized to limit feather contact with motors
3. Launching arms act as guides down the tray to ensure that the birdie is launched correctly and without damage.
4. Passive feeding tube with gasket was used to ensure

Changing Shot Distance

Once the we knew that the birdies could be shot, getting different distance became difficult. There was is slippage between the birdie and the wheels which makes it difficult to hit long shots.

1. Grip tape was first used to reduce slippage (but the motors spun to fast for this work around).
2. Plastigrip spray was used on the wheels to manufacture grip
3. Device was tilted at 12 degrees to get optimal results
4. Piston introduced to change distance based on birdie to wheel contact.

Battery Power

The initial goal was to have a battery powered device to eliminate any power obstacles players may face, the below calculations were done to determine power usage.

$$Battery: 4800\ mAh$$

$$Arduino: 45mA \quad Stepper: 70mA \times 4 = 280mA \quad HC06: 8mA \quad Brushless\ Motors: 1A - 20A$$

$$Device\ Operates\ at\ {\sim}1A, therefore\ Total\ Amperage = 1333mA$$

$$\frac{4800mAh}{1333mA} = 3.6\ H$$

The battery should have lasted for more than 3H but based the battery could not handle the 'bust' of power needed to turn the motors as it exceeds 2A (as seen in the Figure 8). To solve this power usage, the Arduino was battery operated but 2 separate "plug in" power sources were used to power the device.

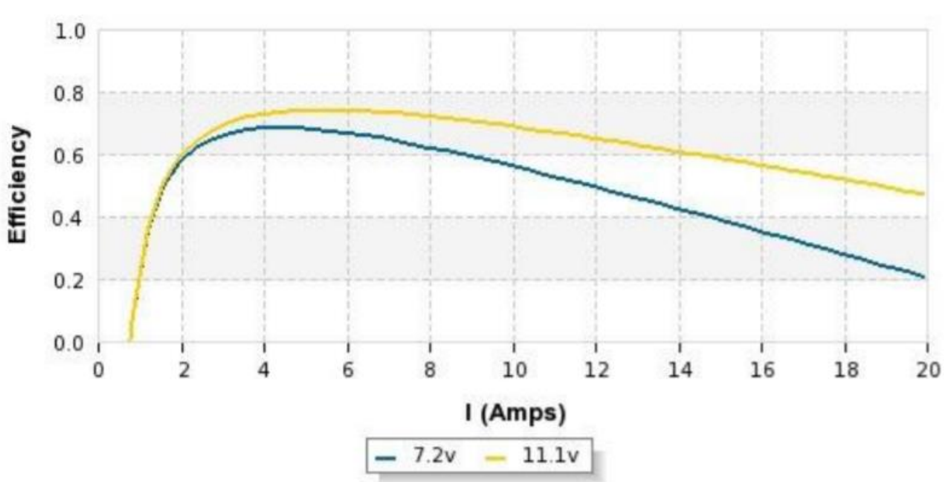


Figure 8 - Brushless Motor Power Graph

Conclusion

Deliverables

1. Physical Badminton Birdie Shooting Device
2. Open Source SVG and STL Files[1]
3. Open Source Arduino Code[2]
4. Open Source Android Studios App Template[3]

Next Steps

Custom Grip

The plastigrip worked a mild solution to the grip problem, but there is still slippage which does not allow the flywheels to get a strong enough grip to launch the birdies to the back of the court. Increasing the voltage of the brushless motors does not correlate to a more powerful shot due to this slippage and to the resistance that the birdie provides to the wheel. The development of a custom circular "stretchy" grip to the wheels would be a next step. If full contact can be without slip, then the birdies can readh the back of the court. Once this issue is resolved, we can have better app to court accuracy. The shots left and right can go further, and the Arduino power settings can be change for an all around more accurate experience.

App & Android Integration

Currently the Arduino only reads in an integer and then launches a birdie. Moving forward, there would be further integration so that the app reads loops through a custom or pre-defined shooting pattern and sends those integers to the Arduino.

App Development

The app has basic functionality with a basic interface that is only set up for a 2XL Pixel smart phone. Moving forward, it would be developed using more relative constraints with relative images (not prefixed size images with fixed constraints). Furthermore, an iOS app would also be developed so iPhone user can also use this device.

Physical Housing

The laser cut logo and pine would looks nice, but we would add a darker finish to the wood and organize the inside of the box. Currently the wires are optimally organized, and it is difficult to reach the Arduino to manually reset it. We would organize the inside of the box and add a compartment for the battery.

Future

As discussed in the background, there are currently two similar devices on the market, the *Black Knight*, which is still unavailable for order and the *Baddy*, an expensive European option. The Baddy is a $600 CND device, our final device cost is $175, and we believe that with some fine tuning we can easily get that cost down to $120. Our system also has a much more intuitive app and is in located Canada. For North American and European badminton players, our device is already more accessible. There are millions of badminton players worldwide and even in Canada, where badminton is not seen as a "popular" sport, almost every city has a badminton club with players who want to train but lack access to consistent training partners. Even at McMaster, the gym is filled daily with badminton players. This device can be used for competitive players

---

[1] Search "Badminton Open Source System Parts" on Thingiverse
[2] Search "Badminton Open Source System" on Github located with Android App Code
[3] Search "Badminton Open Source System" on Github for both java and XML code, located with Arduino Code

looking to get an edge on their competition (or even just have access to quality shots) and for those just a fun and easy way to exercise.

Open Source Kits for Purchase

Our device can be marketed to players of all socioeconomic background. We are currently looking at having three separate price options. For players willing to pay the most, the BOSS can be sent assembled and ready to be used. We think a popular option will also be the BOSS Kit. A ready-to-assemble kit with pre-cut parts, all the motors, and hardware. Lastly, for those looking to keep costs down, we can have a "just the essentials", the player would have to download the SVG and STL files themselves and can choose which hardware pieces they need. If a player already had an Arduino, brushless motors, and wires, they would be able to order a customizable kit that contained the stepper motors and hardware needed.

#	Parts	Cost
2	Brushless Motors	$40
4	Servomotors	$15
1	Ardunio	$15
1	Bluetooth Module	$5
1	3D Printed Device Components – Flywheels (2), Gears (2), Housing Unit	$40
1	Tripod Stand	$25
1	Lazy Susan Bearings	$7
1	Power Supply	$30

Table 6 - Final Device Cost Breakdown

What's Next

Our project is a holistic showcase of five years of engineering education, from first year CAD drawing and second year java and Arduino coding all the way to fourth year socket knowledge. We look forward to making continuous improvements to the physical device, Arduino code, and android app and hope to move from the prototype to market device. We are looking forward to investing the prize money back into the device and looking into entrepreneurial competitions to win more funds to further improve the BOSS.