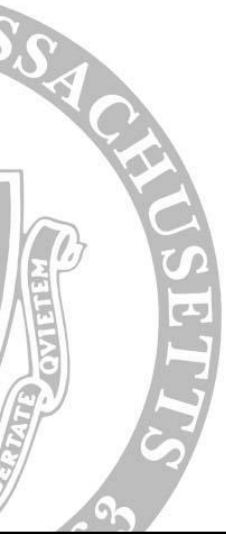
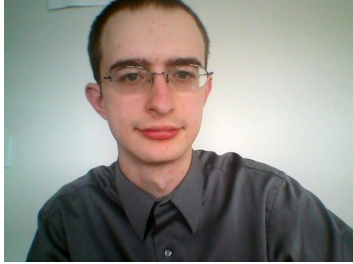


# B.R.O.

Basketball Return Optimizer  
FPR



# Team Members



Derek Foster (EE)



Devon O'Rourke (CSE)



Brian Acker (CSE)



Adam Paranay (EE)



# Project Overview

- Practicing basketball alone is inefficient without someone to return the ball to you
- Even if you make all of your shots, still have to retrieve ball
- Inefficient use of practice time
  - Energy/time lost chasing rebounds
- **Current return systems require manual adjustment**

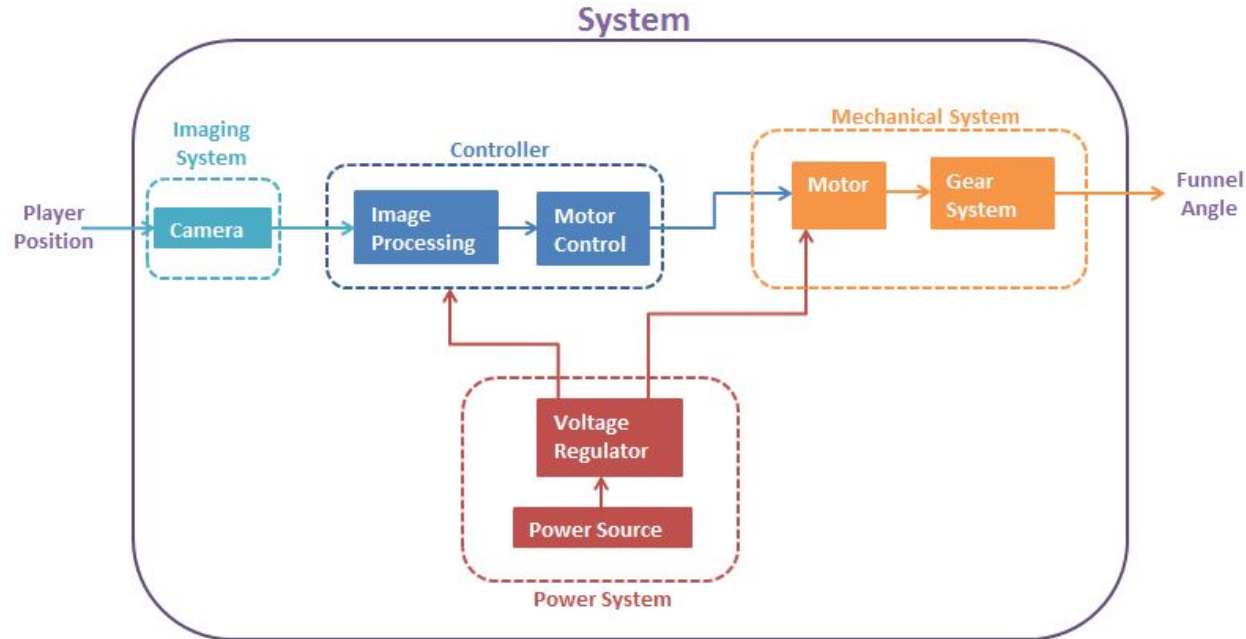


# General Requirements

- Track player at distance of 5-25 feet from rim
- Accurately track lateral movement of player in real-time
- Operational for  $\geq 1$  hour at a time
- System can withstand direct hit from basketball
- System weight does not pull rim downwards
- Easy setup/teardown of electronic part of system



# Block Diagram



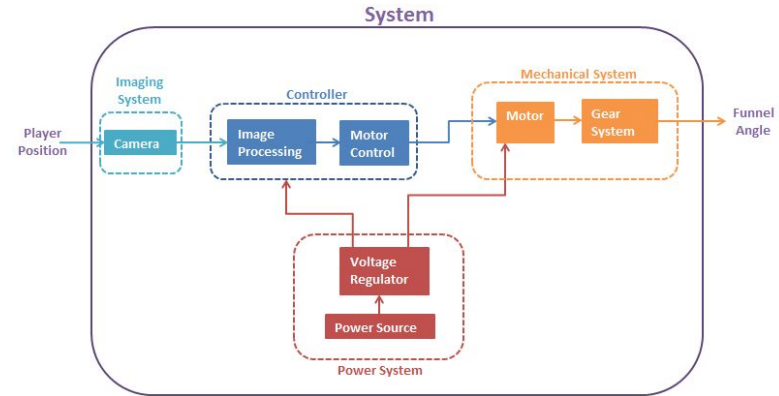
# FPR Deliverables

## ✓ Functional BRO System

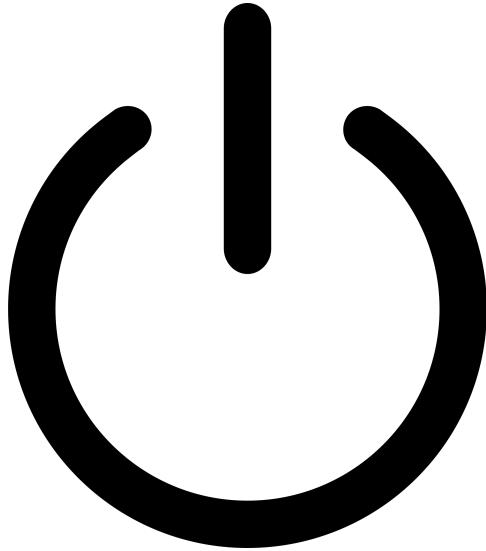
- Lead: All

## ✓ Key Tasks

- Functional PCB
  - Lead: Adam
- All Components Mounted to Funnel System
  - Lead: Devon



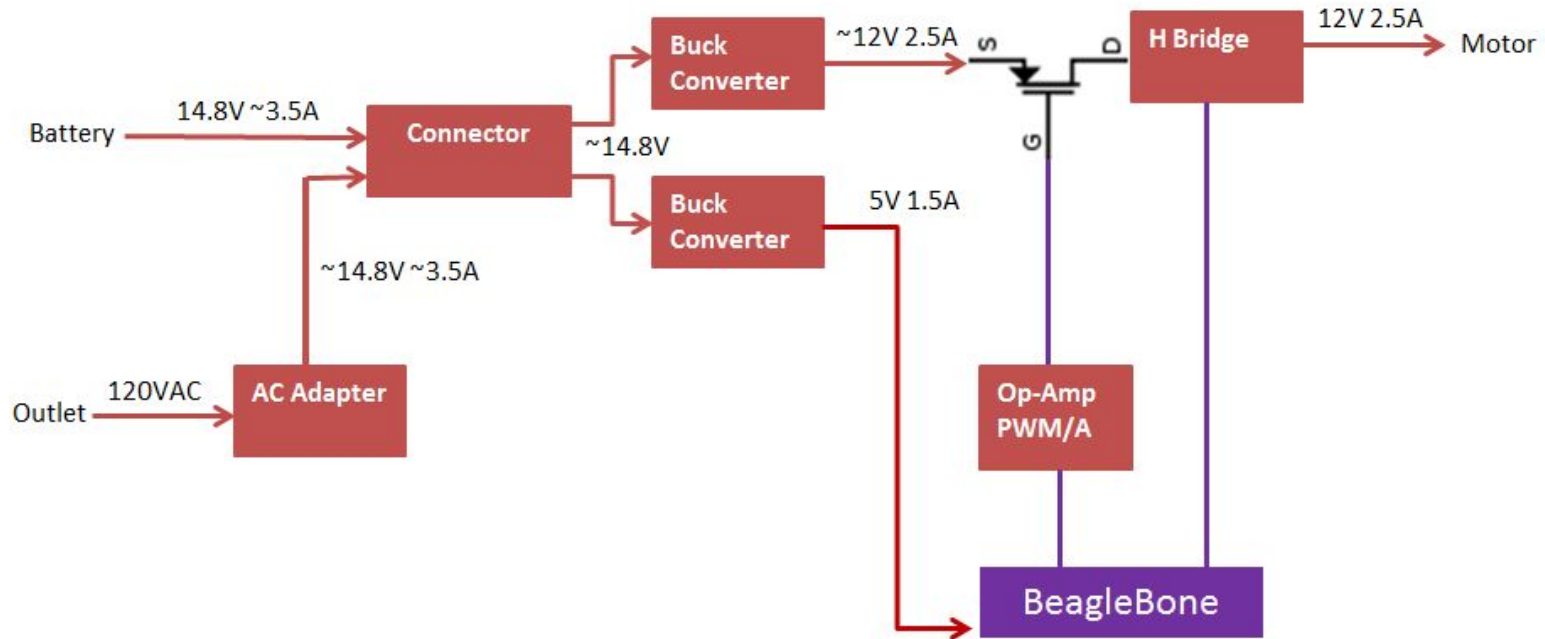
# Power System Requirements



- Supply battery power for  $\geq 1$  hour at a time
  - Limited by motor size (12V, max 2A)
- Lightweight (cannot weigh rim down)
- Power both controller (5V) and motor (12V)
- Switch between battery and outlet power



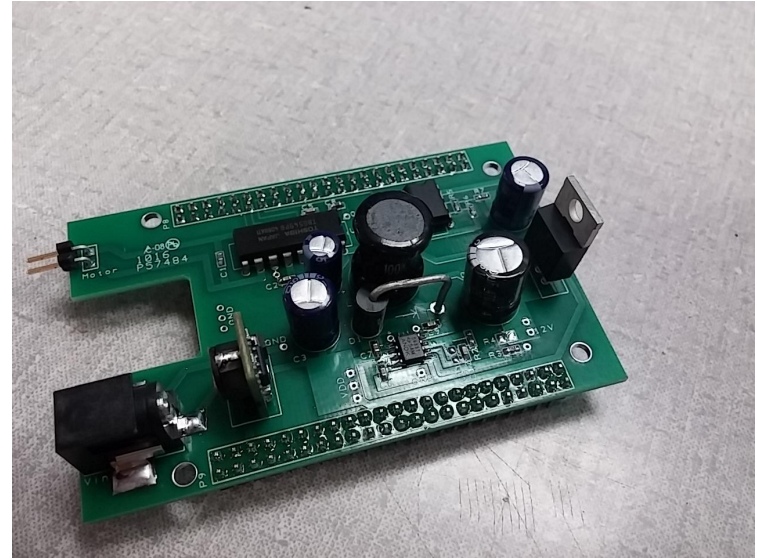
# Power System Implementation





# PCB Design

- Converted working breadboard design to PCB using PCB Artist
- PCB solved finicky breadboard issues



# Image Processing-Code

- Grabber thread continuously grabs frames to flush out camera buffer
- Color filtering process thread searches for Red-Blue-Green striped shirt pattern
- Calculates distance between center of player and center of camera frame
- Runs motor for an amount of time that scales with that distance to center funnel on player



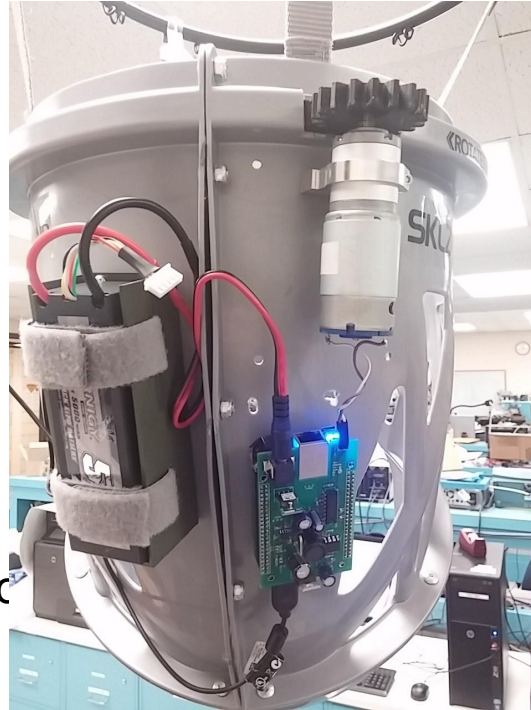
# Image Processing-Challenges

- OpenCV frame retrieval slower than Camera fps led to buffer build up and lag
  - Color Filtering=10-30ms, camera.retrieve()~200ms
- Beaglebone OS/processor restricted access to some OpenCV and V4L2 functions/solutions
  - Multithreading, reducing fps/resolution did not work
- Possible Solution: Write custom code to move and retrieve data instead of OpenCV function



# Mounting

- Designed and 3D-printed casings for the Webcam, the Motor, the PCB/BB, and the Battery/AC Charger.
- Motor, Battery, and BeagleBone are mounted in the back of the system to keep them protected from shots.
- Utilized foam under the mounted camera case to ensure a solid fit and to decrease stress of basketball hits to the case.



# Total Cost

Part	Development	Production (1000)
Processor	\$55.00	\$13.11
SKLZ Shoot Around	\$30.00	\$3.28
PCB	\$45.14	\$27.06
Webcam	\$28.00	\$28.00
Jersey	\$20.00	\$6.09
Battery	\$40.00	\$40.00
AC Adapter (optional)	\$13.00	\$13.00
Motor	\$40.00	\$40.00
3D Printing	\$29.74	\$3.40
Mounting Hardware	\$8.68	\$19.65
<b>Total</b>	<b>\$309.56</b>	<b>\$193.58</b>



# Questions?

