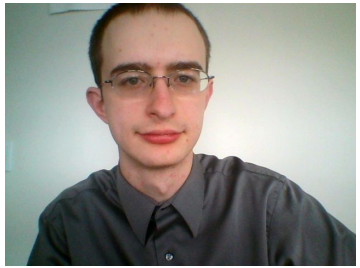
B.R.O.

Basketball Return Optimizer
CDR

Team Members



Derek Foster (EE)

Devon O'Rourke (CSE)

Brian Acker (CSE)

Adam Paranay (EE)

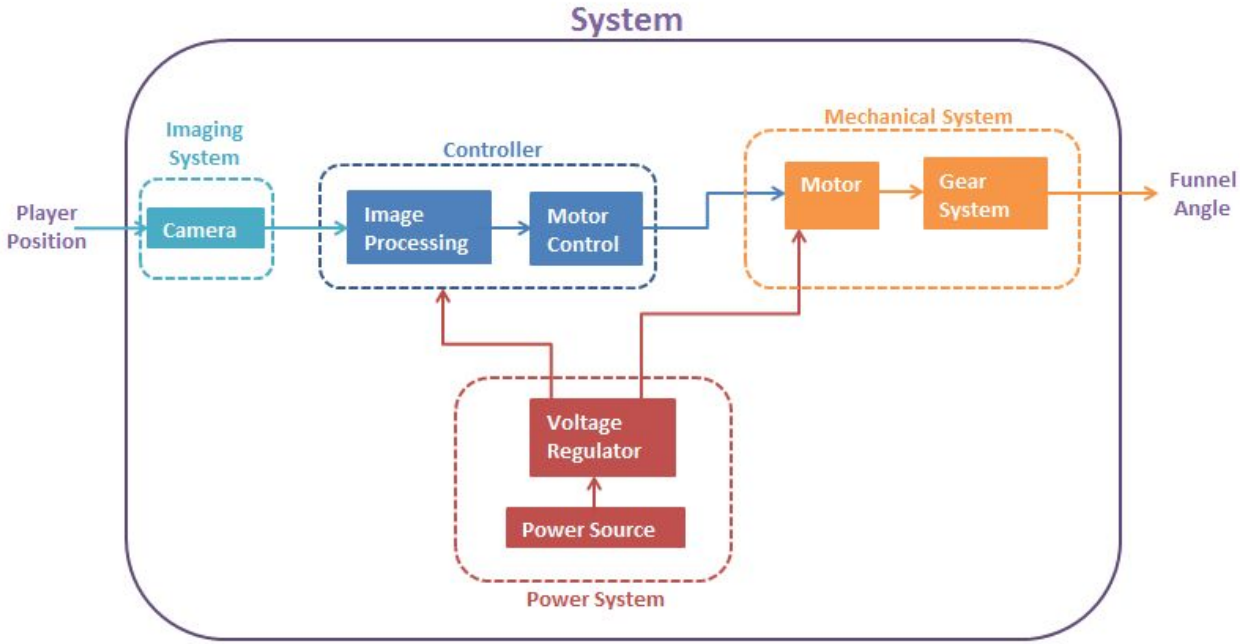
Project Overview

- 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

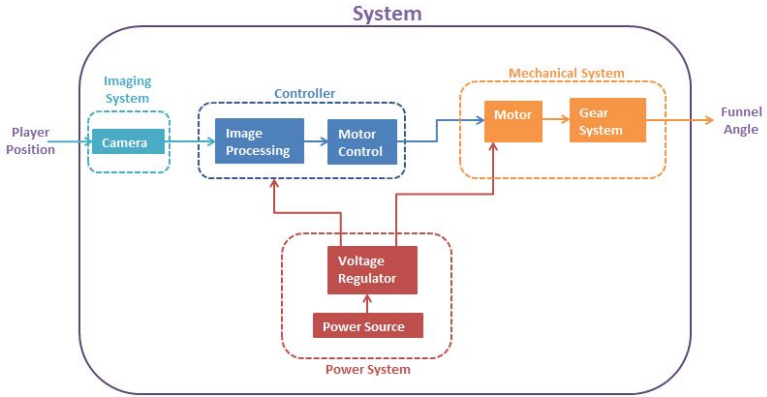
- Track player at distance of 5-25 feet from rim
- Accurately track lateral movement of player in real-time
- Operational for >= 1 hour at a time
- System can withstand direct hit from basketball
- System weight does not pull rim downwards
- Easy setup/teardown of electronic part of system

Block Diagram

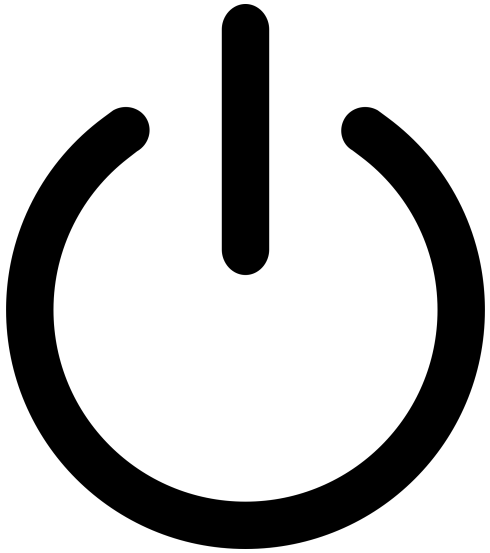


CDR Deliverables

- [x] Camera/BBB/Motor Integration
  - Lead: Brian & Devon
- [x] Completed power system breadboard design
  - Lead: Derek & Adam
- [x] Mounting considerations for hardware
  - Lead: ~~Adam &~~ Devon

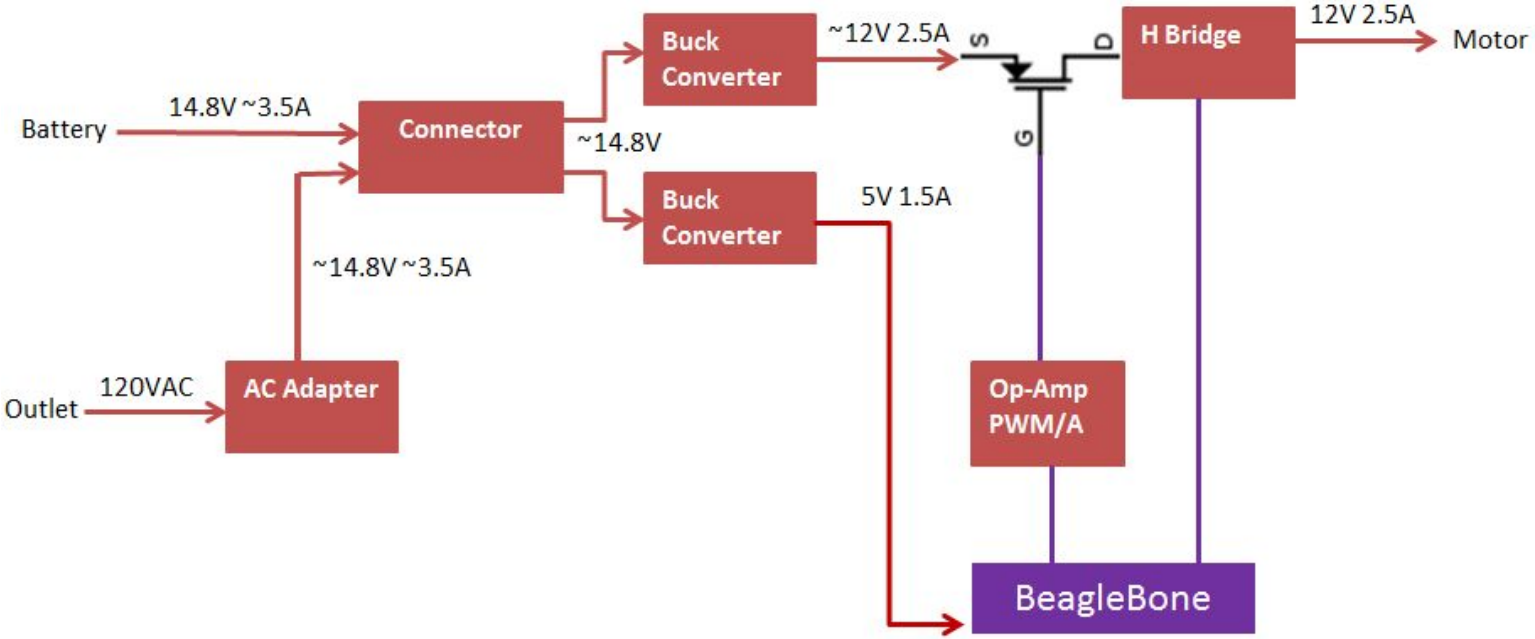


Power System Requirements



- Supply battery power for >= 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



Power System Challenges

- Motor Path
  - Problem: Motor loaded down buck converter output
  - Cause: High motor inrush current
  - Solution: FET between buck and motor to limit current
- BBB Path
  - Problem: Loading of buck converter
  - Cause: Unknown
  - Solution: $4.30 off-the-shelf 5V 1.5A switching regulator

Image Processing-Setup

- Logitech c270 720p HD Webcam
- Beaglebone Black Microcontroller with Debian Linux Operating System
- C++ Image Processing Code written in Cloud9 IDE.
- Code run on Beaglebone boot from an auto run startup script.

Image Processing-Code

- Uses OpenCV and V4L2 libraries to access camera and process images
- Uses 24-bit BGR pixel format in OpenCV Mat
- Compares pixel values to hard-coded target color to find target pixel coordinates
- Determines whether target is left, right, or centered and sets output signals accordingly.

Communicating with the motor

- Uses 3 GPIO pins as output pins-One for ‘turn left’, one for ‘turn right’, and one for enable.
- Uses one PWM pin to supply a voltage to the gate of the PMOS.
- Code controls pins by writing to corresponding GPIO files. Bonescript sets PWM at beginning and shuts it down at the end.

Image Processing-Moving Forward

- Timing
  - current processing takes 10-25 ms per frame, takes new frame as soon as last one finishes processing
  - Must make sure to capture quickly enough to take >=5 frames per second but not move too quickly to confuse motor.
- Output voltages
  - Beaglebone outputs were noisy causing some unexpected motor responses. Fixed by minimizing signal overwrites.

Mounting

3D print the casing or alter existing plastic casing

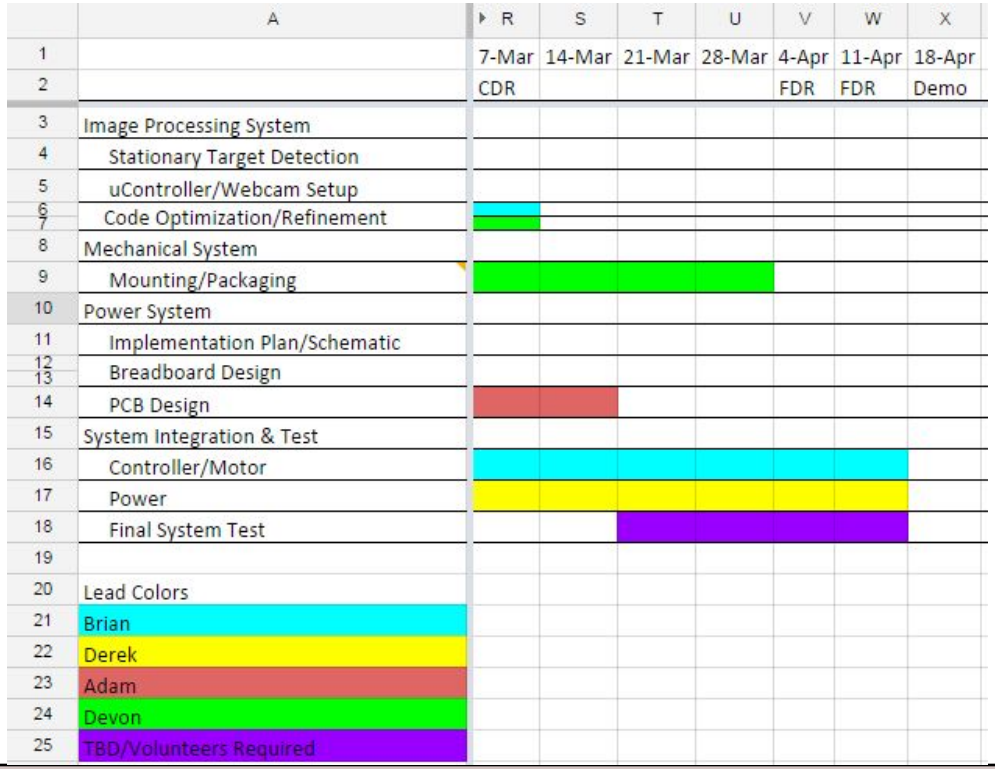
PCB and Beaglebone will stack

Weight will be distributed evenly

Module that can hold either the battery or AC unit

Friction not an issue at this time.

Gantt Chart



	A	R	S	T	U	V	W	X
1		7-Mar	14-Mar	21-Mar	28-Mar	4-Apr	11-Apr	18-Apr
2		CDR				FDR	FDR	Demo
3	Image Processing System							
4	Stationary Target Detection							
5	uController/Webcam Setup							
6	Code Optimization/Refinement							
7								
8	Mechanical System							
9	Mounting/Packaging							
10	Power System							
11	Implementation Plan/Schematic							
12								
13	Breadboard Design							
14	PCB Design							
15	System Integration & Test							
16	Controller/Motor							
17	Power							
18	Final System Test							
19								
20	Lead Colors							
21	Brian							
22	Derek							
23	Adam							
24	Devon							
25	TBD/Volunteers Required							

FPR Deliverables

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

Cost Estimate

- Budget Spent: $316.03
- Budget Remaining: $183.97
- Purchases Remaining
  - Mounting HW - $20-$40
  - PCB & Parts - $40-50
  - AC Adapter - $13
  - Jersey $20-30

- **Worst-Case Estimated Total: $473.78**
- **Worst-Case Final Product Cost: $370**

Questions?

Cost Estimate

- SKLZ Shoot Around - $30
- Beaglebone Black - $55
- Webcam - $26
- Motor - $40
- Gear Printing - $11
- Mounting HW - $20-$40
- Battery/Charger - ~~$70-$100~~ $75
- Power supply - ~~$30~~ $23-$28
- PCB - $35
  - Prototyping & Shipping - $103.78
- Jersey - $20-$30