Project Periscope

Design Review

Team:
Aabhas Sharma (asharm36)
Michael Congdon (mcongdo2)
Siddharth Murali (smurali2)

TA:
Ankit Jain

ECE 445 – Senior Design

Fall 2015

University of Illinois at Urbana-Champaign

Table of Contents

1. Introduction……3
    a. Statement of Purpose 3
    b. Objectives 3
        i. Consumer Benefits 3
        ii. Product Features 4
2. Design……5
    a. Block Diagram 5
    b. Software Flowchart 6
    c. Electrical Schematics for Control Unit 7
    d. Block Description 10
        i. Camera Unit
        ii. Computer – Oculus SDK
        iii. Oculus Rift unit
        iv. Swivel unit
3. Requirements and Verification……13
    a. Modular Functionality 13
    b. Tolerance Analysis 16
4. Cost and Schedule……18
    a. Cost Analysis 18
        i. Labor
        ii. Parts
        iii. Grand Total
    b. Schedule 20
5. Ethical Guidelines and Safety Requirements……...24
    a. Ethical Guidelines 24
    b. Safety Requirements 26
6. Appendix……27

1. Introduction:

Title: Project Periscope

- Statement:

  Currently there's a lot we can do with VR, but can we actually be in an entirely different place in a matter of seconds? We're not quite there yet. The idea behind Project Periscope is to be able to create a more engaging virtual reality experience - one where we can actually have eyes in a very different part of the world.

  Our plan, in its element, is quite simple – connect a camera to an Oculus Rift. We want to view a camera's live feed on the Oculus' display while being able to physically navigate the camera based on the Oculus wearer's head movements – give the whole experience a more realistic feel. What this essentially means is that the camera will follow the wearer's movements, allowing a user to not just see a live feed of someplace else, but also be able to freely navigate through the view with a broader field of vision – all this in real time.

- Objective:

  Consumer Benefits

  - Be able to view a very different part of the world real time
  - Physically navigate through their field of vision - simulating human vision and free range of movement
  - Multiple applications:

    - Can be used as a remote surveillance system (with mobile capabilities as well)
    - Can be used for remote imaging - aerial, terrestrial, and marine
    - Exploring the world - see places where you can't otherwise go to. For example, visit the Eiffel Tower! A VR substitute for View Master
    - Can bridge telepresence with media events - such as football games and conventions or conferences like ComicCon - where the user can explore the event with their own freedom - look wherever they find most interesting
    - Open up future possibilities for nano-imaging, especially for the medical field, allowing doctors to see inside the human body like never before

- Product Features
    - Wireless Streaming
    - Wireless data-transfer between GoPro and the Oculus
    - First-person remote viewing of virtually any environment
    - Remotely controlled camera
    - Head motion tracking

2. Design:

Block Diagram

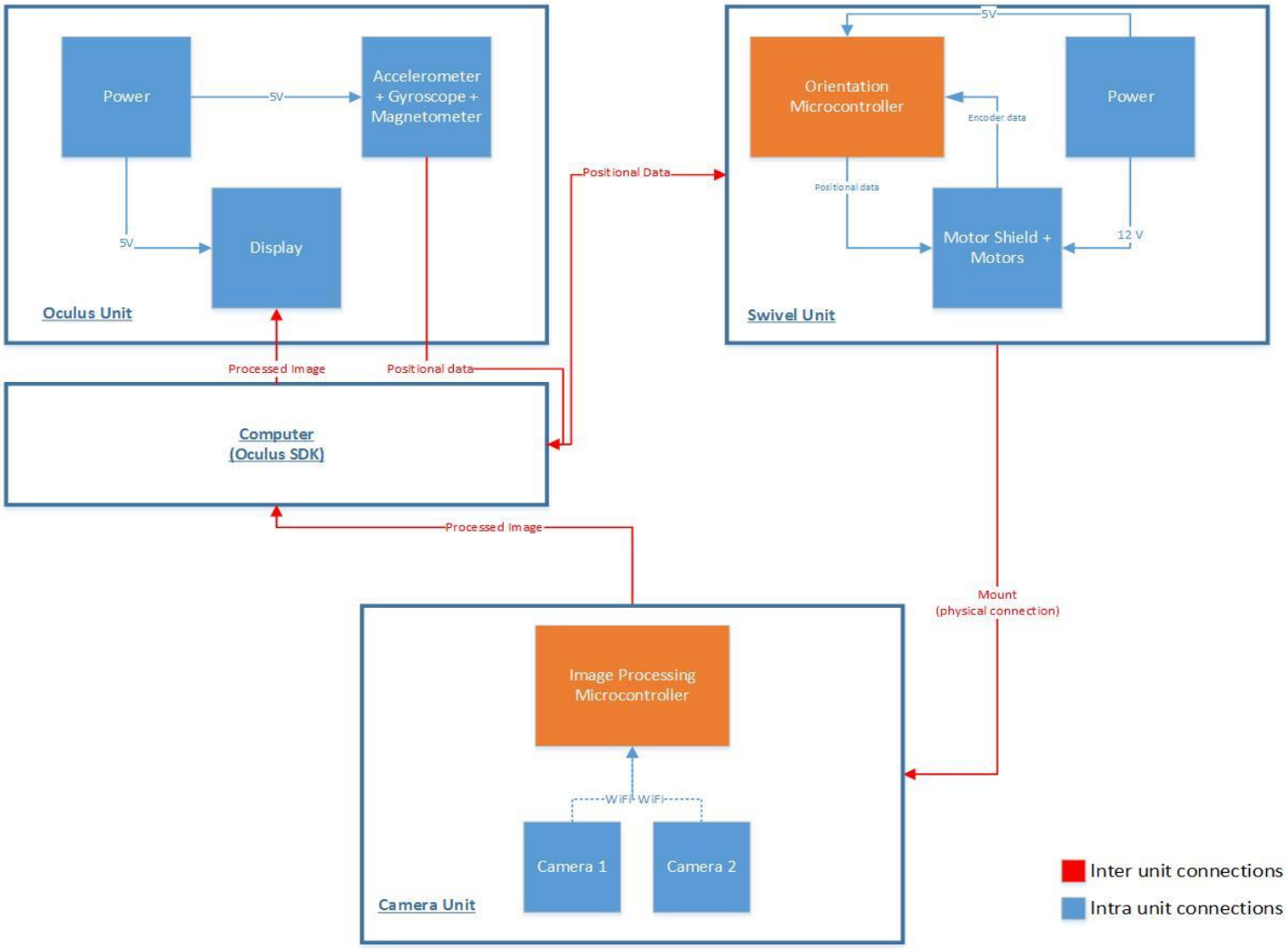


Figure 1: top level block diagram showing data and power connections between various modules.

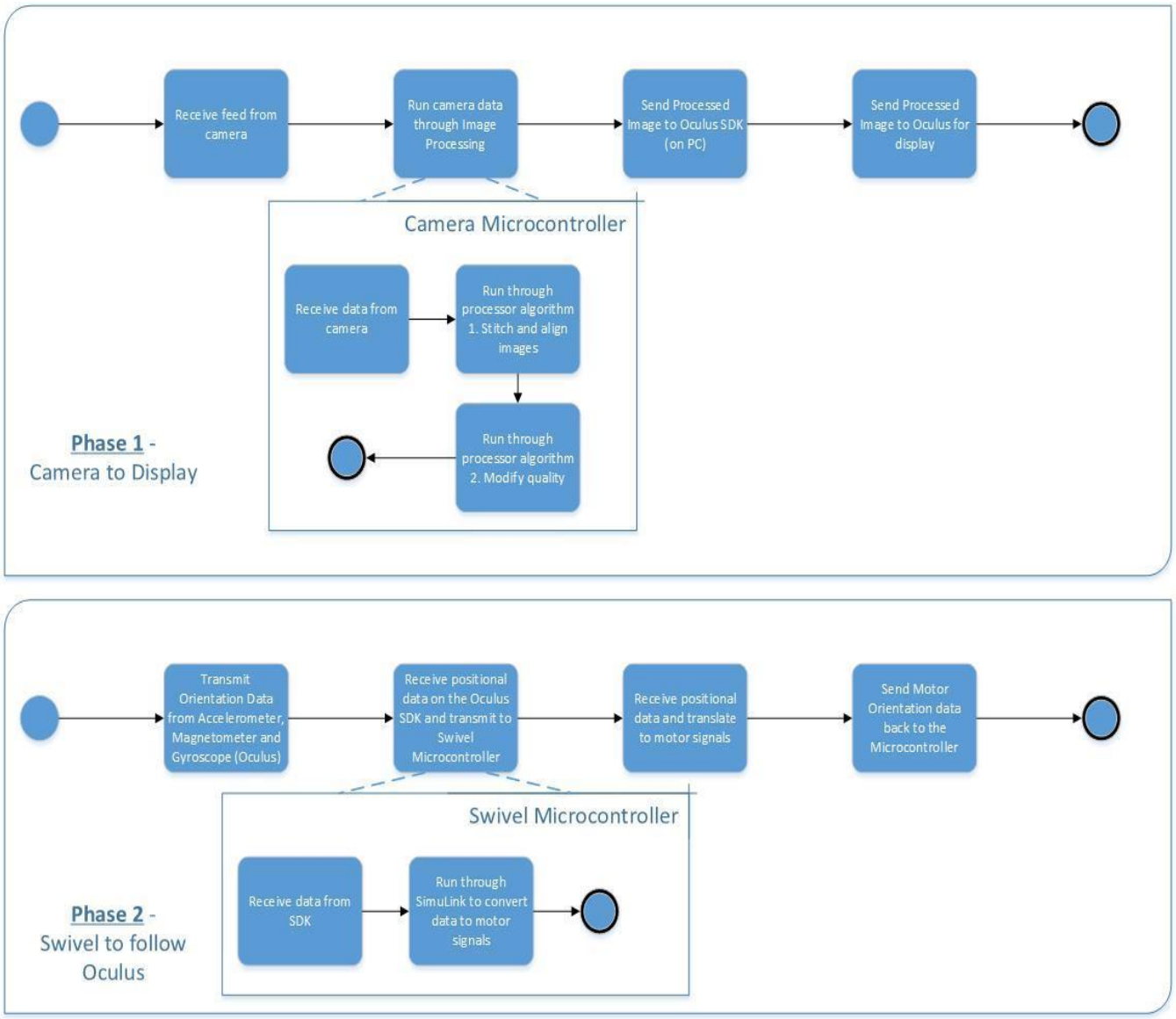


Figure 2: Software flow chart detailing process flow between Camera Unit, Oculus Unit and Swivel Unit

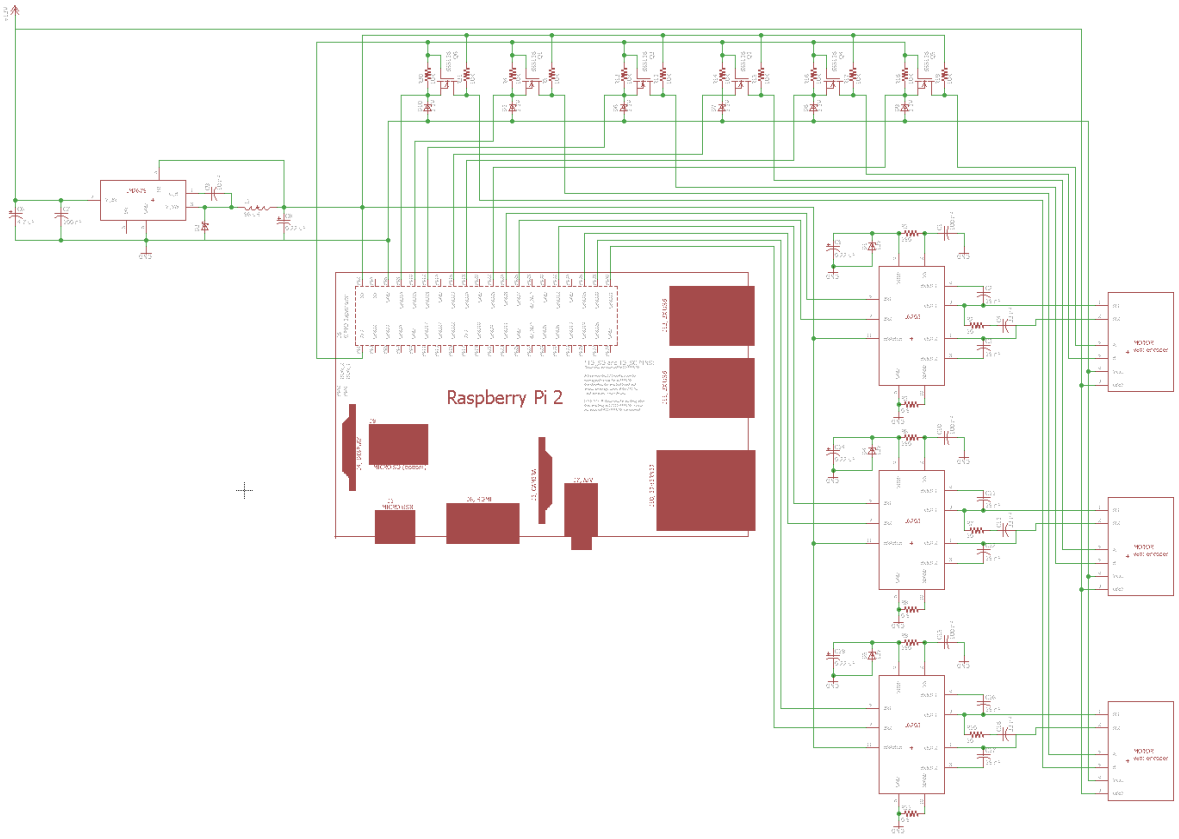


Figure 3: top level schematic showing basic interconnections of motor controller to drivers and motors

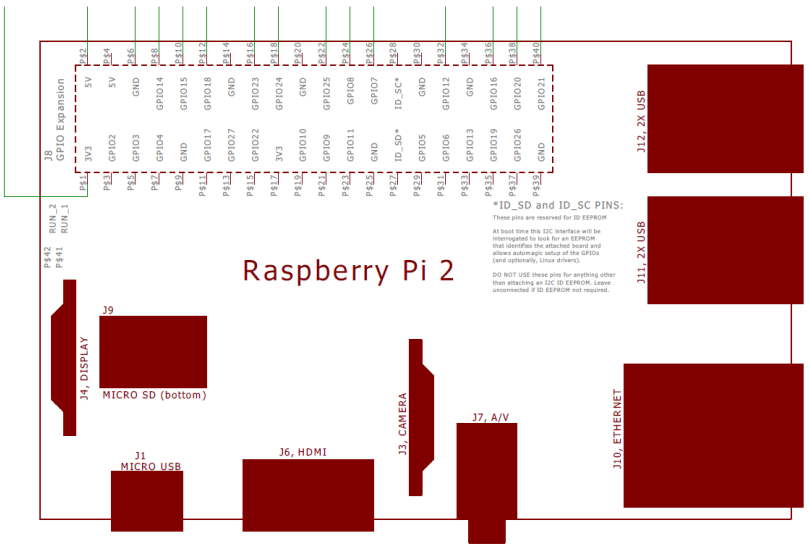


Figure 4: Detailed view of connections to Raspberry Pi

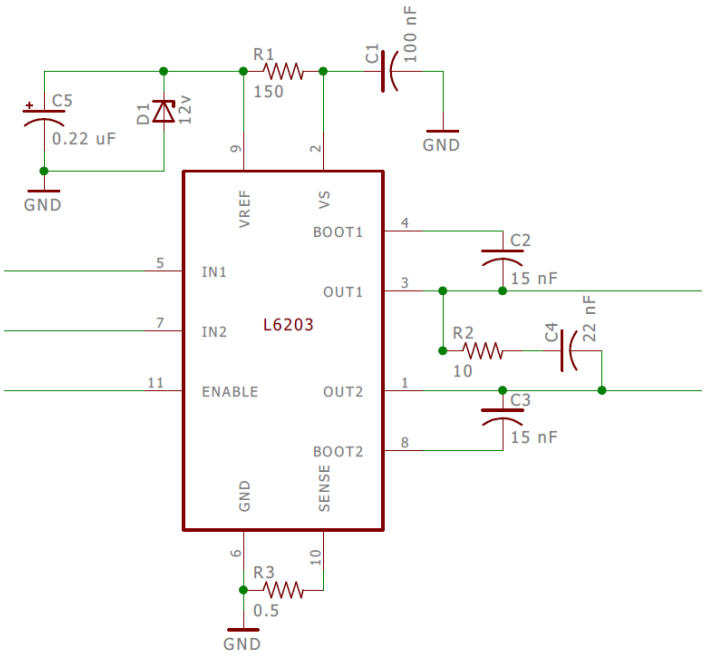


Figure 5: detailed view of motor driver circuit, including peripheral components necessary to regulate voltage and boost switching capacity

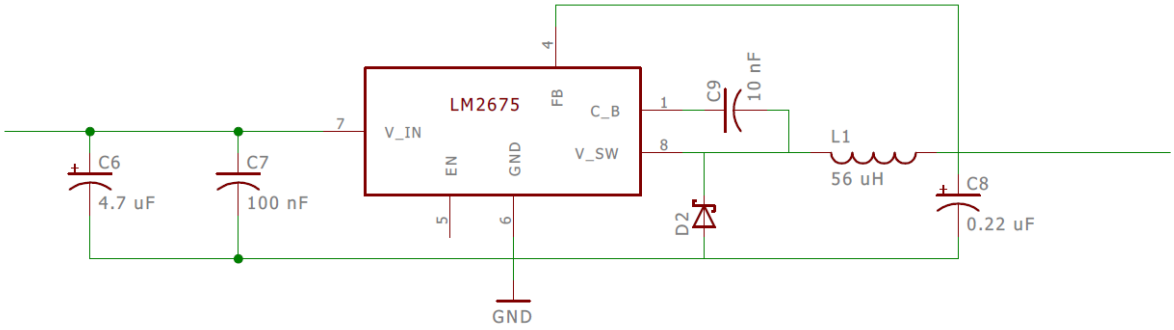


Figure 6: buck converter to provide 5 volts for the Raspberry Pi from 12 volt battery

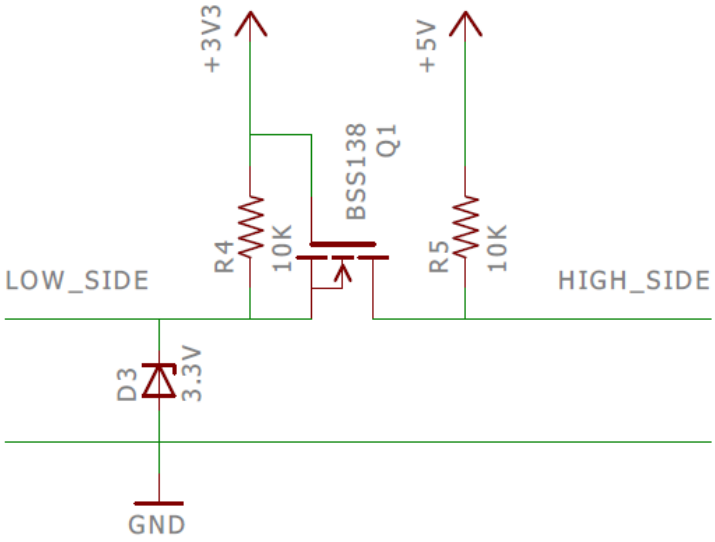


Figure 7: isolation circuit to protect Raspberry Pi from 5v logic level of motor encoders

Block Descriptions

- Camera
    - The camera is the “eyes” of our project. It has its own WiFi hotspot, which the laptop connects to in order to stream the video. The video feed is passed from the camera to an image processing program, which will stitch the two images together and then pass it to the Oculus SDK
    - Microcontroller 1
        - Will be using this to conduct the image processing required to modify and stitch together the camera feed to align with the Oculus display’s expected format.
- Computer - Oculus SDK
    - The Oculus SDK is what controls what you see as well as which way you turn. The video feed is transmitted from the image processing software into the SDK, and this transmits the feed into the Oculus display, showing you what the cameras see. The SDK also receives head movement data from the accelerometer/gyroscope and uses this to determine the direction in which the motor needs to move.
- Oculus Rift Unit
    - This consists of three parts – display, power supply, and motion tracking
    - The display gets the video from the SDK and shows it to the user as a 360 degree view.
    - The power supply powers the Oculus Rift Unit.
    - The motion tracking consists of an accelerometer, a gyroscope, and a magnetometer. These three combined relay the headset orientation data to the SDK.

- Swivel Unit
    - Swivel Unit Power Supply
        - Inputs: none
        - Outputs: 12v and 5v supply rails
        - Components:
            - lead acid battery
                - Selected for substantial capacity and low cost
            - Buck converter
                - Provides 5 volt supply for Raspberry Pi
                - Utilizes LM2675 chip to handle high side switching and feedback control to ensure steady output
    - Orientation Control Unit
        - Inputs: orientation data from Oculus SDK, encoder pulses from motor encoders
        - Outputs: PWM signals to motors
        - Components:
            - Raspberry Pi running Simulink modules
                - Wirelessly communicates with the computer in order to receive orientation data from the Oculus SDK.
    - Motor Shield and Motors
        - Inputs: PWM from control unit
        - Outputs: encoder pulses

- Components:
    - Isolation circuitry
        - Logic level converter will protect the 3.3 volt logic of the Raspberry Pi from the 5 volt logic of the motor encoders.
        - 3.3 volt Zener diodes will be used as a last line of defense if one of the LLC's should fail.
    - Motor drivers
        - L6203 DMOS full bridge drivers receive pwm pulses from Pi and drive the motors.
    - Brushed gear drive motors with quadrature encoders
        - The motors each have 2-pin hall sensor quadrature encoders with a resolution of 64 pulses per revolution of the motor shaft.
        - The gear box has a 70:1 gearing ratio, which gives a total of 4,480 encoder pulses per revolution of the output shaft

3. Requirements and Verifications

- Modular Functionality

Requirement	Verification	Points
Oculus Rift Unit		
● The Oculus position data must correspond 1:1 with User's movements with a tolerance of ~10 degrees	1. Compare vectors during application runtime 2. SDK 3D position vectors (Pitch, Yaw, and Roll) and User movements must align to the exact degree with a tolerance of ± 10 degrees	10
● Achieve a maximum latency of 18ms ± 2ms	1. Compare timestamps for corresponding User movement degree and 3D Vector degree 2. When they are exactly equal, timestamps should not differ by more than 18ms with tolerance of ± 2ms	10
Camera unit		
● GoPros need to be at a separation of 63.5mm with a tolerance of ~ 3.65 mm	1. Measure the distance using the Vernier caliper to obtain a measure of 63.5 ± 3.65mm	5

• Image processor barrel function must return modified image with ***exact*** concave depth as in this formula: $0.24r^4 + 0.22r^2 + 1$ where r is the distance between any pixel and the center of a rift lense[2] • Tolerance ~2px	1. Use test 2D images with available Oculus output images 2. Compare our processed output with sample Oculus image 3. Output should be the same as the sample image, with minor differences allowed	20
• To meet our latency standards of 18ms(± 2ms), the image processing must return output image within 15ms ± 5ms	1. Run program on Raspberry Pi. Make note of program start time 2. When program finishes, the timestamp at the end of the program must be within 15 ± 5ms of the program start time	15
Swivel Arm Unit		
• Buck converter maintains efficiency of 90+% while microcontroller is operating	1. Power Raspberry Pi with buck converter 2. Have Pi power 6 LED's through GPIO pins to simulate load of driving motor drivers	10

	3. Measure input and output voltage and current of buck converter, ensure adequate efficiency rating is met	
● Motors are able to produce 1.5 Nm of torque to within 10%	1. Attach simulated load calculated to require 1.5 Nm of torque to lift/rotate 2. Power motors with battery, make sure they are able to lift the load	10
● Motors should be able to actuate camera unit with less than 7% overshoot. (At 8+%, the differences in FOV's may be overcome and the video will oscillate)	1. Power on system, with either the real camera unit or simulated loads attached to each motor 2. Using step input (to simulate worst case scenario of infinitely fast head movement), have motors turn unit 180 degrees 3. Measure encoder outputs and analyze system response. Should not exceed 7% overshoot.	20
Total		100

- Tolerance Analysis

  A major challenge for us to be able to map a 2D camera image feed to the concave Oculus display. The Oculus offers the user a very immersive experience by presenting a virtual image that appears larger than it actually is. Although this is a key positive aspect that enables the complete VR experience, it could result in major health hazards if not done right. If the Oculus' display is screening images that are distorted or have unexpected skips in motion, the user could become vulnerable to seizures, epilepsy attacks, motion sickness etc.

  Given this situation, it is important for us to ensure we are significantly accurate in our image processing. We will be running our camera feed through barrel distortion algorithms[2] that will map our two dimensional feed to a concave screen using the following formula: $0.24r^4 + 0.22r^2 + 1$. Basically how this works is that for any pixel on the Oculus screen that is distance r away from the centre of the concave screen, the appropriate pixel to be copied over from the 2D camera feed is given by $0.24r^4 + 0.22r^2 + 1$. [2]

  Our tolerance to this has to be significantly small. As highlighted earlier, we can't have a poorly distorted image. Therefore, we are only accepting a tight tolerance of ± 2px offset in radius distance while mapping the 2D image to a barrel image.

  The way we will ensure this tolerance is using a fundamental method of verification that is used in most image comparison software - we will use sample Oculus images as the

input for our image processor. We will then compare our output image to the Oculus version of the sample image. While comparing, we will not just check each 2D pixel with its corresponding barrel pixel location, we will check all barrel pixels within a 2 pixel radius of the expected barrel location. Our verification program will run through the entire output image and compare each pixel with nearest neighbors (2px) on the sample output, if there is a match, we would consider this phase a success, else we will have to go back and debug.

4. Cost and Schedule:

Part	Value	Digikey #	Price	Qua.
Buck Converter (x1)				
Capacitors	4.7 uF	493-10470-1-ND	$ 0.24	1
	10 nF	490-1312-1-ND	$ 0.10	1
	100 nF	490-1318-1-ND	$ 0.10	1
	47 uF	P5539-ND	$ 0.23	1
inductors	56 mH	AIUR-06-560K-ND	$ 0.88	1
diode	Schottky	MSS2P3-M3/89AGICT-ND	$ 0.37	1
IC	LM2675	LM2675MX-5.0/NOPBCT-ND	$ 4.03	1
			$ 5.95	
Motor Driver (x3)				
resistors	150 ohm	311-150GRCT-ND	$ 0.10	1
	10 ohm	311-10ERCT-ND	$ 0.10	1
	0.5 ohm	311-0.5LWCT-ND	$ 0.47	1
capacitors	.22 uF	493-1096-ND	$ 0.23	1
	100 nF	490-1318-1-ND	$ 0.10	1
	15 nF	490-1643-1-ND	$ 0.34	2
	22 nF	490-3884-1-ND	$ 0.10	1
zener	12 v	568-6354-1-ND	$ 0.20	1
IC	L6203	497-1421-5-ND	$ 9.53	1
			$ 11.51	
			$ 34.53	
Logic Level Converter (x12)				
resistors	10 kohm	311-10.0KCRCT-ND	$ 0.10	2
IC	BSS138	BSS138CT-ND	$ 0.23	1
zener	3.3 v	MMSZ4684-TPMSCT-ND	$ 0.15	1
			$ 0.58	
			$ 6.96	
Microcontrollers, etc…				
controllers		Raspberry Pi	$ 50.00	2
vision		Oculus Rift	$ 350.00	1
cameras		GoPro Hero3+ Silver	$ 300.00	2
motors		70:1 Metal Gearmotor	$ 40.00	3
battery		sealed lead-acid	$ 20.00	1
battery charger		UPG D1724	$ 9.47	1
			$1,199.47	
		Total cost of parts	**$1,246.91**	

Name	Rate	Time	Cost
Siddharth Murali	$ 75.00	200	$ 37,500.00
Aabhas Sharma	$ 75.00	200	$ 37,500.00
Michael Congdon	$ 75.00	200	$ 37,500.00
		Total	**$ 112,500.00**
		Section	Total
		Parts	$ 1,246.91
		Labor	$ 112,500.00
		Total	**$ 113,746.91**

Schedule

Week	Task	Responsibility
30-Sep	Design circuitry for motor controller	Michael
	Obtain and set up Oculus Rift and SDK	Siddharth
	Get raw video feed from camera to computer	Aabhas
7-Oct	Design armature, meet with machine shop	Michael
	Design microcontroller algorithm for image processing	Aabhas
	Implement Oculus application for viewing raw video feed from a camera	Siddharth
14-Oct	Order motors, give them to machine shop. Finalize PCB design	Michael
	Program Oculus for transmitting head motion data. Cross reference parts list for ECE shop availability.	Siddharth
	Implement microcontroller algorithm for image processing	Aabhas

21-Oct	Characterize system dynamics, write control logic for Simulink	Michael
	Implement protocol to receive head motion data from Oculus and Simulink on a Pi	Siddharth
	Optimize image processing algorithm to return output within tolerance timeframe	Aabhas
28-Oct	Test/validate motor function and basic control	Michael
	Test/validate Oculus functions (display, head tracking)	Siddharth
	Design microcontroller algorithm for image processing Program Raspberry Pi for transmitting video to Oculus	Aabhas
4-Nov	Refine control logic	Michael
	Tolerance testing of system	Siddharth
	Test/validate video transmission and image processing. Prepare for Mock Demo	Aabhas

11-Nov	Final assembly of components	Michael
	Ensure mobility of system	Siddharth
	Ensure entire system functionality	Aabhas
18-Nov	Final testing and debugging – Hardware	Michael
	Final testing and debugging – Oculus + Motor Control	Siddharth
	Final testing and debugging – Software (Camera)	Aabhas
25-Nov	Make sure project is ready for demo	Michael
	Make sure final presentation meets format requirements	Siddharth
	Create presentation support content and talking points for team members	Aabhas
2-Dec	Create final paper content for Swivel unit and ensure uniform formatting	Michael
	Create final paper content for Oculus unit and ensure uniform formatting	Siddharth

	Create final paper content for Camera unit and ensure uniform formatting	Aabhas
9-Dec	Make sure all major parts are returned	Michael
	Make sure nothing is missing from lab kit	Siddharth
	Complete project closure related tasks such as IPR management and secure code sharing	Aabhas

5. Ethical Guidelines and Safety requirements

a. Ethical Guidelines

We will adhere to the following guidelines from the IEEE Code of Ethics as they are relevant to our project in the capacity listed below.

1. **To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;**

   We will be applying high safety standards to ensure that our product will be easy to use. We will have warnings that list the potential consequences of using our product for too long, and safety methods to be followed to ensure that the user has a enjoyable and safe experience with the product.

2. **To be honest and realistic in stating claims or estimates based on available data;**

   We will cite all sources used in formulating our claims, and we will mention the areas where we have chosen to make our own claims and give our reasoning behind them.

3. **To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;**

   We will have taken all the required courses in order to complete this project and we will do additional research in the areas that our expertise is lacking.

4. **To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;**

   We are working closely with the TAs and professors in the Design Review and peer evaluations of our project, and we will work to incorporate their criticism and suggestions into our final design.

5. **To treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;**

   We are a team and we will distribute work and give honest evaluations of team members without discriminating against anyone. If there is a problem we will meet with the TA/professor and work it out.

6. **To avoid injuring others, their property, reputation, or employment by false or malicious action;**

   We do not wish to cause harm to any person or their property in any way. Our product is intended to broaden the scope of a particular area and is not intended to harm anyone intentionally. We will mark all dangers of our product with safety warnings if required.

7. **To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.**

   We will do our best to support other teams and projects in this class by offering help that is within our area of expertise. We will participate in peer design reviews in order to aid this process.

Safety requirements

- There are a few common safety hazards connected with the usage of an Oculus Rift. Before using this system, go through the safety documentation created by Oculus Rift (Link).
- Pinching Hazard - Our product has motors that will rotate depending on user movement. If someone gets their finger caught it could cause injury. Individuals need to be careful while working with the motors and the camera mount as they may have pinch points. Also keep your hands away from the motor when the device is in use as that is the most likely to cause injury.
- Our product has many components that cannot be exposed to water. If they are, there is the chance of them shorting out and destroying the circuit or injuring the user.

Appendix:

1. Angular velocity and acceleration data acquired from this study:

   http://scholar.lib.vt.edu/theses/available/etd-08182005-222028/unrestricted/thesis.pdf

2. How barrel distortion works

   https://www.youtube.com/watch?v=B7qrgrrHry0&feature=youtu.be&t=11m26s

   Barrel Distortion

   http://stackoverflow.com/questions/28130618/what-ist-the-correct-oculus-rift-barrel-distortion-radius-function