Virtual Reality Controllers for Medical Training

By
Corey Zeinstra
Vignesh Gopal

1. Introduction **3**
1.1 Block Diagram 4
1.2 High-Level Performance Requirements 5
2. Design **6**
2.1 Design Procedure 6
2.1.1 Battery 6
2.1.2 Voltage Regulator 6
2.1.3 Inertial Measurement Unit 6
2.1.4 WiFi Controller 6
2.1.5 Time of Flight Sensor 7
2.1.6 Pressure Sensor 7
2.1.7 LEAP Motion 7
2.1.8 Unity Application 8
2.2 Design Figures 8
3.Verification **11**
3.1 Syringe/Ambubag 11
3.1.1 Distance Sensor 11
3.1.2 Universal Controller 11
3.1.3 IMU 12
3.1.4 WiFi Chip 12
3.1.5 Voltage Regulator 13
3.1.6 Pressure Sensor 13
3.2 Simulation 14
3.2.1 Unity Application 14
3.3 Quantitative Results 14
3.3.1 Sensor Accuracy and UDP Consistency 14
3.3.2 Latency 15
4. Costs **16**
4.1 Parts 16
4.2 Labor 16
5. Conclusion **18**
5.1 Accomplishments 18
5.2 Uncertainties 18
5.3 Ethics and Safety 18
5.4 Future Work 19
References **20**

1. Introduction

The healthcare industry is one of the largest and fastest growing industries in the world. It is projected to rise from 7.077 trillion dollars to 8.734 trillion dollars by 2020 [1]. Currently, a major issue that medical professionals encounter while training is the lack of resources available to actually simulate complex and delicate procedures. A synthetic human cadaver costs about 40 thousand dollars each, and fewer than 20,000 real cadavers are donated to science each year [2-3]. This pretty much matches the number of medical students in all of the US. With the growing demand for improved health care, doctors must be adequately trained to perform procedures. Clearly, there is a mismatch between the number of doctors and training resources available for training physicians. This leads to a rise in malpractice and botched medical procedures.

The need for improved medical tools and devices is a growing industry. Deloitte projected that the medical technology sector would experience a compound annual growth rate (CAGR) of about 15.9% between 2016 and 2021 [1]. While a majority of this industry focuses on devices for real procedures, because of the rising costs allocated towards training physicians, there is an increasing demand for simulated training for medical students. Recently, virtual reality training has been tested for testing the competency of EMTs. Mcgrath [5] concluded that simulation softwares are indeed effective for training and testing. However, he noted that the rapidly changing Virtual Reality (VR) industry has made it hard for healthcare professionals to handle the various platforms available. Also, many proposed solutions are relatively high cost, so it is important to ensure that sensors and controllers are used efficiently. We plan to design a cost-efficient solution that would be easy for any general medical professional to use and train on. In the end we created a board with an attached suite of sensors that could read information from several medical tools and relay that information back to a VR application. A block diagram for this solution is given in figure 1.

1.1 Block Diagram

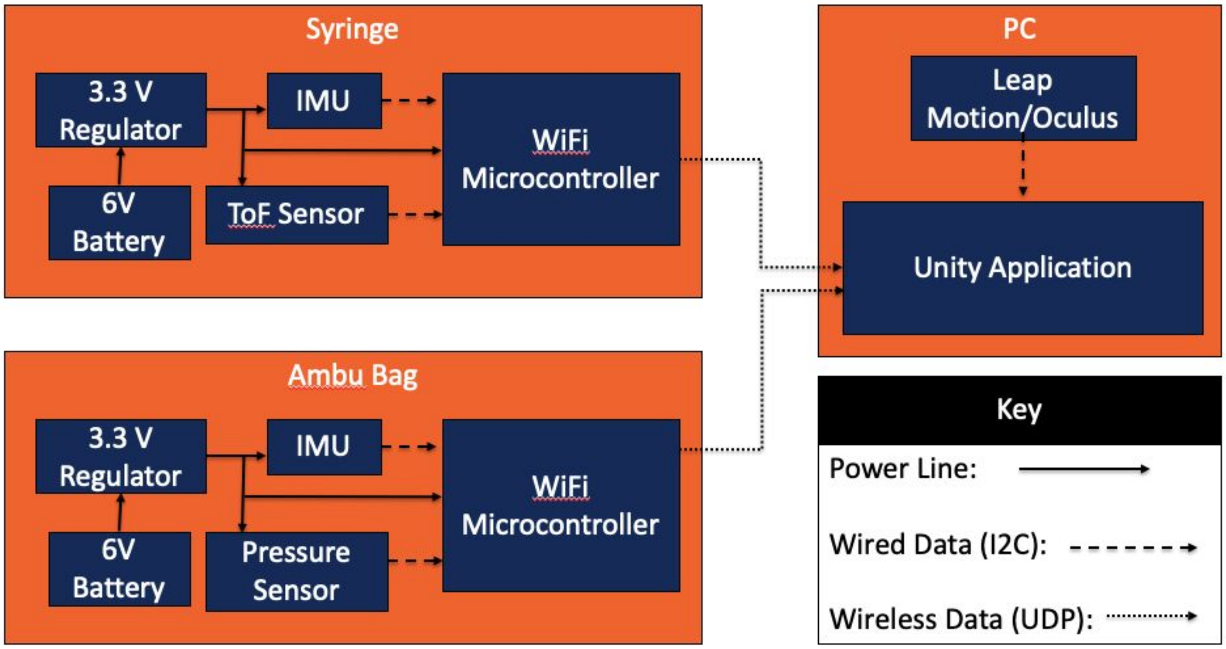

Figure 1. Block Diagram

At its core our project consists of 3 distinct elements. The first being the Personal Computer (PC) and its peripherals which include the Oculus Rift and the LEAP motion tracker. Then the other two elements are the controllers that we created, they are very similar by design and were created to have swappable sensors to allow for more devices and readings later on. They both consist of a battery and a linear voltage regulator to step the voltage down to the required level for our chipsets to function properly. Then both of them have an 9-axis Inertial Measurement Unit (IMU) that measures the orientation and acceleration of the chip and sends that information onto the Unity Application on the PC. Then each device has a different sensor to read a specific parameter about the tool that it was attached to. For the syringe a Time of Flight (ToF) sensor was used to measure the distance from the end of the barrel to the plunger of the syringe. This was then used to calculate how much fluid was to be held in the virtual syringe. For the Ambu Bag a Pressure sensor was used to determine the amount of pressure being pushed into the patient's lungs. It was designed to mimic the manometer that can be found on several off the shelf Ambu Bags or other Bag Valve Masks. Finally there is the WiFi Microcontroller that is designed to take information from all of the sensors using an I2C connection and then relay that information over a UDP signal in order to the Unity Application. This block diagram has several changes from the original project proposal, but at its core is basically the same as the one created for the Design Document.

1.2 High-Level Performance Requirements

1) Pressure and Distance sensor should be calibrated to within 10 kPa and 3 mm respectively.
2) Universal Controller should be able to relay information to Unity application with low latency (<100 ms)
3) Power source (batteries) must be able to last the length of at least 2 simulations (~30 minutes total).

2. Design

2.1 Design Procedure

2.1.1 Battery

In the end there were a lot of problems with the batteries and powering the devices. Our original plan was to purchase a rechargeable Lithium battery to power our devices. However before we purchased it we decided to test with some different single use batteries, both CR2032 and AA type batteries were tested and we were unable to get them to power the board properly. After those failures and with time running low we decided that our best option would be to simply attach a USB power bank to the device and wire that into the USB breakout board. This was not a desirable approach to powering the device and in future iterations and with the PCB actually working we think it would be best to switch back to a small rechargeable lithium battery with voltage between 3.3V and 6V depending on which voltage regulator and sensors would be chosen.

2.1.2 Voltage Regulator

Again the voltage regulator was originally designed and built into our PCB however with that not working we ended up purchasing breakout boards that had their own built in voltage regulators. For cost and complexity this is not the ideal situation and the next iteration of the product should remedy this.

2.1.3 Inertial Measurement Unit

We decided to use a 9 Degree of Freedom IMU that could measure orientation, acceleration and magnetic fields. An IMU is almost necessary to make a VR controller we might have been able to replace it with a suite of Infrared sensors and emitters but that seemed bulky, complex, and unnecessary. The specific BNO055 was chosen over some cheaper chipsets because it had a good suite of available resources and code repositories that made our project more manageable. Also the orientation, acceleration, and magnetometer were necessary to work in conjunction with the LEAP Motion to both initialize the controller position and make sure that it lines up with certain position and movement values throughout the simulation.

2.1.4 WiFi Controller

For this device, we decided to use an ESP8266 WiFi controller that was responsible for reading sensor data through its I2C bus then sending this information over UDP packets to our Unity

Application. We decided to use I2C connections as it allowed us to connect multiple sensors to the same WiFi controller. Also, we chose to use UDP over TCP because we needed multiple sensors to send data with low latency. If we had chosen TCP, each reading sent would require an acknowledgement requiring the order to matter. Using UDP, we were able to send packets of information quickly without having to worrying about the order they were sent in. The ESP8266 was capable of meeting our requirements.

2.1.5 Time of Flight Sensor

We decided to try using an Adafruit VL6180X Micro-Lidar distance sensor. This part was used to measure the distance the plunger was away from the bottom of the syringe. This was then analytically converted to the volume of liquid in the syringe. The sensors range was between 0 and about 150mm, which was a sufficient range for the syringe we used. However, we ran into some issues with the accuracy of the ToF sensor. When the plunger was pressed against the sensor, we were still receiving a non-zero value. In the future, we plan to use a better distance sensor to get a more accurate reading.

2.1.6 Pressure Sensor

We chose an Adafruit MPRLS sensor in the case of the pressure sensor. We decided to use an air pressure sensor because it could be attached and read values similar to that of the Manometers found on off the shelf Ambu Bags. The MPRLS also allows us to attach tubing to the end and then read the pressure of the air inside that tube unlike many other pressure sensors this method was said to be good for "blow and suck" applications. Finally the MPRLS sensor was made by adafruit and we were ordering several other parts from them so to keep things simple we decided to just order the pressure sensor from them as well.

2.1.7 LEAP Motion

The LEAP Motion was used at the request of HCESC. They have already done some tests with the LEAP Motion tracker and wanted it to be used in the final product. It was also chosen because it was able to reduce the weight and complexity of the actual controller. It was attached to the headset and added some position tacking ability without having to make any changes to the physical controller itself. In future iterations it would be good to look into Infrared sensors and emitters like those that are used in off the shelf VR controllers like the Oculus Rift and HTC Vive. We were worried however that this would add a lot to our development time and complexity and were worried that the added weight would make smaller controllers like the syringe to become clunky and hard to use.

2.1.8 Unity Application

We decided to use the Unity Game Engine to power the software side of our project as it is a popular VR game development engine that we are well versed in using and it could handle all of the requirements it needed to.

2.2 Design Details

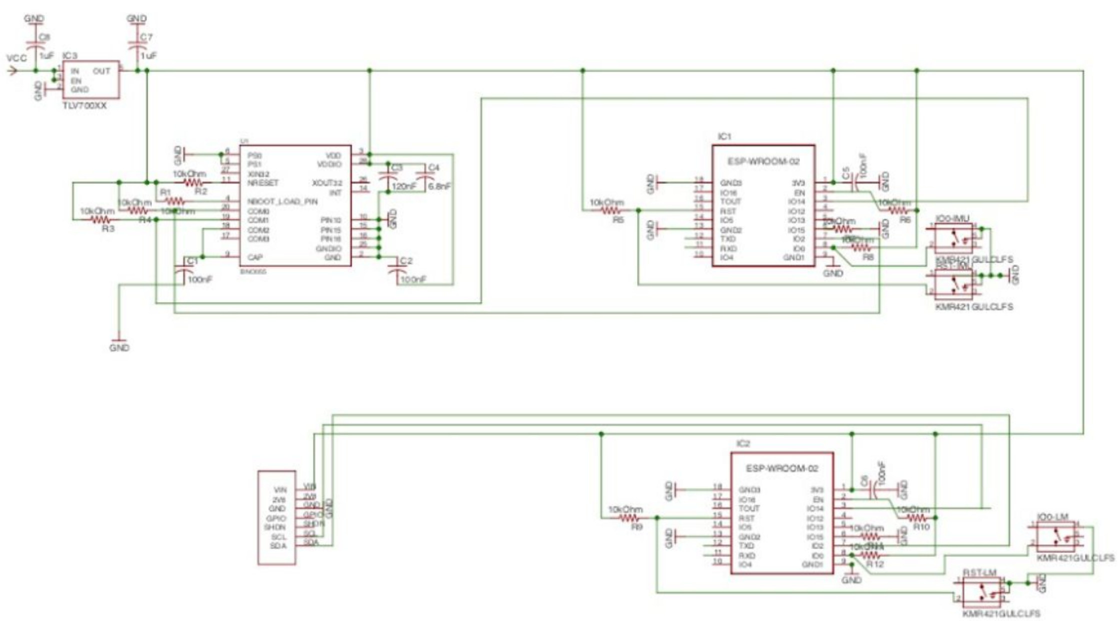

Figure 2. Full schematic for project

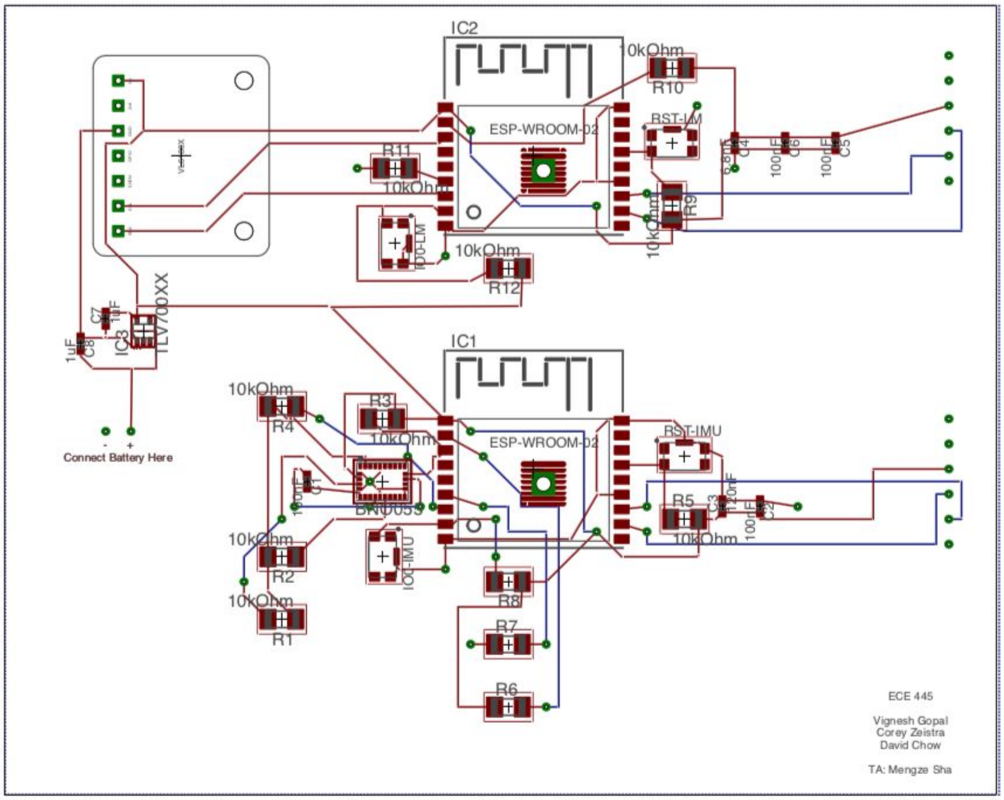

Figure 3. Full PCB layout for project developed on Eagle.

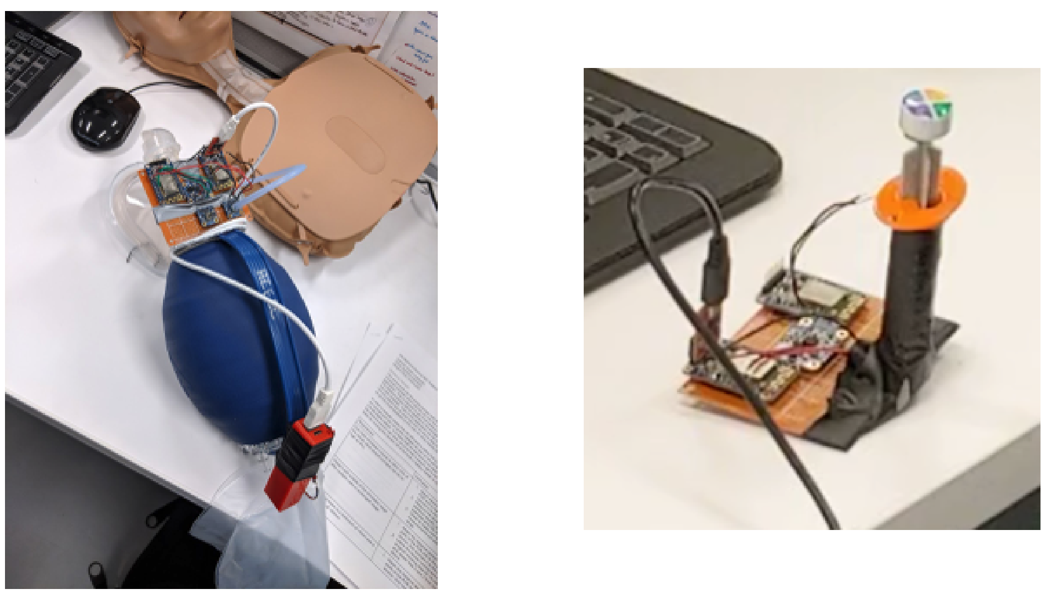

Figure 4. (Left) Ambu Bag with pressure sensor; (right) Syringe with distance sensor

3.Verification

3.1 Syringe/Ambubag

3.1.1 Distance Sensor

Requirement	Verification
Can track simulated volume to within 10% of the actual value (actual syringes can be as high as 5%).	1. Attach distance sensor to I/O ports of PCB and to the power source. And connect it to the syringe. 2. Measure the sensor at five different positions equally spaced between fully extended and fully closed 3. Check to make sure all of the measurements are within the allowed range.
Doesn't drift outside of the simulated range after multiple uses or prolonged usage.	1. Attach the distance sensor to the I/O ports of the PCB and to the power source. And connect it to the syringe. 2. Leave the device running and the program tracking data for 10 minutes and repeat step two and three from row above.
Can read down to a minimum of 0mm and a maximum of 150mm	1. Attach the distance sensor to the I/O ports of the PCB and to the power source. And connect it to the syringe. 2. Depress the plunger until it is 0mm away from the sensor and record the output. 3. Extend the plunger until it is 150mm away from the sensor and record the output.

3.1.2 Universal Controller

Requirement	Verification
Must be able to read data from at least one external sensors. (Pressure sensor, ToF, etc)	1. Plug sensor into the I/O ports of the board and have the board send their data to the computer, check on the computer if both devices data are being tracked.
Must have a low consistent latency. An end to end latency of <50ms is ideal <100ms is acceptable	1. Measure latency from three main sources: a. Sensors to microcontroller (sampling rate) b. Wifi to PC c. Unity application (frame rate)

3.1.3 IMU

Requirement	Verification
Should be able to work within the bounds of that the Voltage regulator is constrained to 3.6V to 3.0V	1. Connect the IMU to the power supply and slowly move it from 3.0V to 3.6V make sure that the IMU is reading correct values the entire time.
Should be able to send quaternion data through I2C bus to WiFi module. Accuracy is not vital here as the angle is not essential to the simulation. The quaternion data should be good enough to look and feel real to the user	1. Hook the IMU up to the Microcontroller. 2. Have the Microcontroller output the raw data from the IMU while rotating the device 3. Visually see with processing test of rotating bunny if IMU is responding accurately

3.1.4 WiFi Chip

Requirement	Verification

Must be able to send at least 95% of collected data wirelessly through UDP messages	1. Repeatedly print to serial and send UDP packets in same loop for 30 seconds 2. Compare number of serial lines and number of UDP packets received.
Must be able to operate within the bounds of 3.0V to 3.6V.	1. Connect the microcontroller to a power supply and vary the voltage between 3.0V and 3.6V. 2. Have the microcontroller constantly running some function to make sure it keeps working consistently.

3.1.5 Voltage Regulator

Requirement	Verification
Must be able to maintain a voltage of between 3.0V and 3.6V throughout the entire lifecycle of the battery.	1. Connect the battery to the Voltage Regulator and add in a circuit that will drain the battery. 2. Constantly measure the voltage coming out of the regulator to make sure it is constantly within the required bounds. 3. Recharge the battery and repeat step 2 twice more to ensure it works consistently.

3.1.6 Pressure Sensor

Requirement	Verification
Must measure a change in pressure of roughly 10-20 kPa when used on artificial lung	1. Attach an actual pressure sensor to the Ambu bag as a control. 2. Wire up our pressure sensor. 3. Use the Ambu bag and see if their outputs are within range.
Must be able to accurately signal when the user squeezes the bag so that interval can be tracked.	1. Squeeze the Ambu bag at a regular 5 second interval for 2 minutes and record the results. 2. Make sure these results correspond to the actions you performed.

3.2 Simulation

3.2.1 Unity Application

Requirement	Verification
Must correctly react to the data coming from the controllers and external sensors. All 3D models must move accordingly.	1. Connect the controllers to the unity worldspace and measure how they map in the virtual world to how they are set up in the real world.
Must perform operations quickly to keep the end to end latency low.	1. Move the controllers in the real world and measure how long it takes for it to react in the virtual world. 2. Repeat step 2 10 times and the leave the controller stationary for 2 minutes. 3. Repeat step 2 another 10 times.

3.3 Quantitative Results

3.3.1 Sensor Accuracy and UDP Consistency

To test the sensors, we tested each device several times to see how the simulation was reading compared to what the syringe/ambu-bag was at in the real world. For the syringe, we used the built-in stopping points to test the distance reading at several different distances. For the pressure sensor, we used a dummy lung to see if the average pressure reading was between 10-20 kPa which should be the typical amount. To see how consistently these readings were being sent throughout the simulation, we ran the simulation several times and tested how many packets were received vs. how many were sent.

UDP Consistency	99.93%
Linear Motion Accuracy	93.42%
Pressure Sensor Difference	17.75 kPa

3.3.2 Latency

To test latency, we had to measure the latency at several different points in the device. Below is a table of all the areas that experienced non-negligible latencies.

Latency Tests		
Connection	**Latency**	**Description of Measurement**
Sensors to Wifi	<50 ms	The sampling rate is 1/50ms
WiFi Unit to PC	~22 ms	Send message to PC and record response time over 2 (See right)
Unity Processing	<10 ms	Frame rate
Total	**~ 80 ms**	Sum of latencies

4. Costs

4.1 Parts

Below is a table of the parts that were required to make our prototype.

Part	Cost (prototype)	Cost (bulk)
Syringe	$15.00 or Donated	$2.50
Distance Sensor	$7.50	$2.00
Universal Controller (PCB)	$3.10	$0.10
IMU	$13.00	$2.10
Ambu Bag	$20.00 or Donated	$4.00
Wifi Chip	Average $7	$1.20
Leap Motion	$0 (Donated)	$76.75
Resistors and Capacitors	$10 (single unit purchases)	$1.03
Total	**$47.60 - 82.60**	**$89.68**

4.2 Labor

Based on the information we collected, our development costs are about $45/ hour, and about 10-15 hours/week for two people. And we consider to about 10 weeks this semester.
So our labor cost is about:

Max: 2 * 45 *15*10 = $13,500
Min : 2 * 45 *10*10 = $9,000

Week	Duties	Corey	David	Vignesh
26-Feb	Buy parts and controllers	x		x

4-Mar	Configure Sensors	x		x
11-Mar	Design and Assemble PCB			x
25-Mar	Program Wi-Fi controller with sensors and VR headset	x		x
1-Apr	Build Unity Application	x	x (environment setup only)	
8-Apr	Begin testing device	x		x
April 15 to end	Refine prototype	x		x

5. Conclusion

5.1 Accomplishments

Overall, we were able to accomplish most of the goals that we had for this project. Our sensors were able to reliably read rotation, distance, and pressure with accuracy that met our design requirements. Our unity application was able to render the simulation with low enough latency to avoid motion sickness, and the simulation itself was able to go through the entire procedure needed for intubation and injection with feedback given to the user while the actual operation was happening. One accomplishment we were not able to achieve was to use the exact PCB we designed. There were 2 wiring issues that we could not resolve on the PCB itself. This required us to perfboard the schematic we designed with some small modifications.

5.2 Uncertainties

The main source of uncertainties in our simulation were from the IMU and distance sensor. We did not have a precise way of checking whether the IMU rotation measurement was accurate. However, since the angle was more for the user to find the simulation more lifelike, precision was not essential for the IMU. The distance sensor was not very reliable either. There was some oscillation that we corrected in software. We theorize that this may be caused by ambient light that interferes with the distance sensors reading. Other than that, future iterations will have to work more closely with medical professionals to make the simulation more useful for actual training procedures.

5.3 Ethics and Safety

We must ensure that the real environment that the trainee is in completely safe as the user would not be able to see what is actually around them. For this, we think that the best course of action is to not use blunted tips on the laryngoscope device (this is fine because the actual device does not need to cut into anything). Also, we plan on having the actual devices tethered to the work station that we develop. This way, the user would not be able to move far away from the dummy while they are essentially blinded to their actual environment. Overall, the design does not have too much room for safety hazards, but we find it important to address the ones that do exist.

Ethically, VR in medicine has had a history of contentious points. Historically, many medical professionals have raised concerns regarding the accuracy and verisimilitude of simulations such as ours [6]. Also, many professionals have raised concerns regarding biases in VR research applications stating that researchers would be motivated to make advancements in their simulations at the expense of patients that would be later impacted by this technology. We believe that our project will not violate any of these ethical concerns as our device will be used as ancillary training tool for students who will have to perform the procedure on cadavers regardless. Essentially, this tool would not be a replacement for practicing on an actual body, but would rather be a training tool used before actually carrying out the procedure. Also, we plan on using well-accepted 3D models of a human larynx/throat in our simulation to ensure adequate verisimilitude.

The IEEE ethics code rule number 6 states that we must be careful when discussing the limitations of our work as one could be encouraged to give more credit to their project than actually deserved [6]. I think that it is important for us to make sure that our device is clearly only intended for supplemental training purposes. Ethically, our product must explicitly state the differences between the simulation and how the procedure would be in real life.

5.4 Future Work

This project was definitely more of a proof of concept than an actual finished product. As it sits we would be comfortable bringing the Ambu Bag to a convention or demoing it to individuals in the industry but much future work is still necessary in order to polish the product up. Firstly the during the design process of this product we prioritized modularity and ease of creation over cost and functionality. This was the right choice for this class being that we had a limited time frame and getting to product working was top priority. In the future if this project is continues we would like to first off revamp our microcontroller WiFi system either by purchasing a more powerful chip ESP32 or by adding some circuitry to allow the current chips to handle more sensors on the I2C bus. This would also allow us to make a smaller PCB (and one that actually works) which would make our design more compact and useable. Secondly the VL6180X was not up to our standards the data sheet said it would easily match our requirements but in practice it was far too inconsistent to be used. In the future another system for measuring distance would be preferable. Finally we would like to get the actual battery system working, we were forced to hack an improper solution together at the end and it would be much better if we could use some cheaper off the shelf solution for the battery. Also it would be good to look into some position systems past the LEAP motion, It worked for our purposes fine, but was expensive and buggy and possibly a homemade solution could be preferable where cost and accuracy are concerned.

References

[1] S. Burrill, “2019 US and Global Health Care Industry Outlook,” *Deloitte United States*, 25-Jan-2019. [Online]. Available: https://www2.deloitte.com/us/en/pages/life-sciences-and-health-care/articles/us-and-global-health-care-industry-trends-outlook.html. [Accessed: 07-Feb-2019].

[2] S. Bachai, “Synthetic Human Cadaver To Be Used By Medical Students Instead Of Actual Humans,” 13-Nov-2013. [Online]. Available: https://www.medicaldaily.com/synthetic-human-cadaver-be-used-medical-students-university-arizona-instead-actual-humans-262768. [Accessed: 07-Feb-2019].

[3] M. Maccall, “The Secret Lives of Cadavers,” *National Geographic*, 29-Jul-2016. [Online]. Available: https://news.nationalgeographic.com/2016/07/body-donation-cadavers-anatomy-medical-education/. [Accessed: 07-Feb-2019].

[4] “Laryngoscopy: Purpose, Procedure, Types, And Complications.” [Online]. Available: https://www.webmd.com/oral-health/what-is-laryngoscopy. [Accessed: 07-Feb-2019].

[5] J. L. Mcgrath, J. M. Taekman, P. Dev, D. R. Danforth, D. Mohan, N. Kman, A. Crichlow, and W. F. Bond, “Using Virtual Reality Simulation Environments to Assess Competence for Emergency Medicine Learners,” *Academic Emergency Medicine*, vol. 25, no. 2, pp. 186–195, 2017.

[6] Ieee.org. (2019). *IEEE Code of Ethics*. [online] Available at: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 22 Feb. 2019].

[7] Niehorster, D., Li, L. and Lappe, M. (2017). The Accuracy and Precision of Position and Orientation Tracking in the HTC Vive Virtual Reality System for Scientific Research. *i-Perception*, 8(3), p.204166951770820.

[8] Orland, K. (2019). How fast does “virtual reality” have to be to look like “actual reality”?. [online] Ars Technica. Available at: https://arstechnica.com/gaming/2013/01/how-fast-does-virtual-reality-have-to-be-to-look-like-actual-reality/ [Accessed 22 Feb. 2019].

[9] Adafruit.com. (2019). *Fully Integrated Proximity and Ambient Light Sensor with Infrared Emitter and I2C Interface*. [online] Available at: https://cdn-shop.adafruit.com/datasheets/vcnl4000.pdf [Accessed 26 Feb. 2019].

[10] Sparkfun.com. (2019). *MPU-9250 Product Specification Revision 1.0*. [online] Available at: https://cdn.sparkfun.com/assets/learn_tutorials/5/5/0/MPU9250REV1.0.pdf [Accessed 26 Feb. 2019].

[11] Ambu.com. (2019). [online] Available at: https://www.ambu.com/products/anaesthesia/resuscitators/product/ambu-disposable-pressure-manomenter [Accessed 8 Mar. 2019].