TrueTouch

Anthony Chan, EE, Alexander Dickopf, CE, Cameron Kluza, CE, and Jonathan Yip, CE

Abstract **- When thinking about virtual reality (VR), we often think of headsets such as the HTC Vive and Oculus Rift. These headsets are great at immersing the user because they have screens that occupy the user's entire peripheral vision. While great at immersing the user visually, these headsets lack immersion through other senses, specifically touch. Interactions with objects are done through a controller that often only provides minimal haptic feedback via vibrations. The user's hands are stuck conforming to the shape of the controller, and buttons are pressed to interact with objects. This restricts users to using a small subset of their hands' capabilities while interacting with objects in a virtual world. TrueTouch is a Bluetooth Low Energy (BLE) wearable glove that provides users a sense of grabbing and feeling the shape of an object. TrueTouch uses strings in combination with a ratchet and pawl system to restrict fingers from going through objects in the virtual space. TrueTouch also uses eccentric rotating mass (ERM) motors to simulate the feeling of impact or vibrations.**

I. INTRODUCTION

Headsets such as the HTC Vive and Oculus Rift have screens and headphones that provide visuals and sound to the user. While great at providing visuals and sound, the sense of touch in virtual reality setups is absent. When operating the headsets, the user uses controllers to interact with the virtual space. Controllers hinder the user's immersion because the user's hands are fixed in the shape of the controller. Additionally, the user presses buttons to interact with the objects, which is not realistic. Newer models, such as the Oculus Quest, have built- in hand tracking which allows the user to move their hands and fingers freely. This allows the user's hands to be in different positions; however when grabbing objects in the virtual world, the user's hands can go through objects, which reduces immersion. Moreover, plain hand tracking supported in newer headsets also loses almost all haptic feedback, including the vibrations that controllers could previously provide.

Integrated systems that provide enhanced interaction through multiple haptic feedback mechanisms exist on the market today but are often limited in terms of usability, affordability, and compatibility. Such solutions might be limited in their availability to specific industries or to fields of research. They can also include bulky exo-skeleton designs that add cost and inconvenience to an average user.

TrueTouch aims to solve these issues and provide a convenient solution to enhance immersion in virtual worlds through combined vibrotactile haptic feedback and physically limiting the motion of a user's fingers to simulate the ability to grip and hold virtual objects. A low-weight, wireless solution will provide these capabilities in a convenient and usable form factor.

A. Significance

Some applications of such a device are hands-on job training, rehabilitation, and games. On the topic of rehabilitation and medicine, according to Neuro Rehab VR, the use of VR-based rehab can improve neuroplasticity in patients (or "building pathways through practice and repetitions in the brain") in different ways such as through multisensory stimulation, task-specific practice, and use of affected limbs [1]. Using a device such as the TrueTouch in this type of application will allow patients with disabilities or going through recovery in their hand to practice different rehabilitation lessons through long-distance communication with their doctor, as well as acquiring specialized data through the VR program that the TrueTouch is linked to.

In the realm of gaming, realism and real-world accuracy for users has been a goal for many companies that specialize in VR-based products or games. This is especially true when it comes to interaction with virtual objects. The TrueTouch provides an initial baseline for implementing this goal by removing the standard controller used by current-generation VR headsets. These can then be replaced by the TrueTouch in order to give the user a sense of realistically interacting with and handling game objects that cannot realistically be interacted with, thus improving the immersion and enjoyment factors associated with VR gaming.

There are also certain jobs that require hands-on training for new employees in order to meet certain safety or experience requirements. This can include surgeons practicing certain types of operations on patients or car mechanics learning how to repair different parts of a car. Should the respective person not be able to come in their practice area in person (such as in the case of the early stages of the COVID-19 pandemic), the TrueTouch- when paired with a program developed for certain practices- can provide a realistic practice space for the user to train at home.

B. Context and Competing Solutions in Marketplace

There has been development for accessories that provide these missing senses. There are companies that are providing arm and leg tracking, treadmills, and haptic vests and gloves. One such accessory that provides a similar functionality of shape in virtual worlds is Wireality [2]. The Wireality project utilized a similar system of strings stopped by a ratchet-and-pawl mechanism in order to stop a hand from moving forward when contacting a virtual object or wall. However, the Wireality device is mounted to the shoulders and can only stop motion away from the point of mounting (i.e. moving your hand/fingers away from your body). This limitation doesn't allow proper interaction with the fingers curling around and gripping objects. Our project targets this different application of grip by altering how the wiring is connected to the glove and where the ratchet-and-pawl mechanism is mounted. Moreover our project is wireless, while the Wireality project is wired, which further limits its general-purpose use cases.

Another similar system in the marketplace is the VRgluv [3]. The VRgluv provides very similar capabilities to our project, including targeted limiting of individual fingers' range of motion to simulate grabbing and holding objects, vibrotactile haptic feedback, and (optionally) wireless gloves. The gloves implement these capabilities with an exoskeleton on the glove that can limit the fingers' range of motion. However, the VRgluv is not available to consumers and is instead targeted towards industry and research. Our project will be something much more readily available and usable, and will provide these capabilities without a bulky exoskeleton attached to the back of a user's hand.

C. Societal Impacts

Providing a realistic sense of touch enables VR to deliver more compelling experiences to users. Now users can pick up and interact with objects more naturally. Being able to interact with virtual objects in a more realistic manner could benefit several potential markets and fields. General consumers of virtual reality products that use their virtual reality devices for gaming can experience enhanced immersion in virtual environments. Several of our design choices were made to make the experience ideal for these users, including minimizing the weight of our device, making it wireless for easy and convenient use, and eliminating any bulky exoskeleton that could feel unnatural wrapped around a user's hand.

These benefits apply equally to several other use cases in other fields. Training for things like surgery, war, operating heavy machinery, and much more can be improved with TrueTouch. Virtual training environments are already seeing attention [4], [5], [6], and adding convenient and convincing interactions with objects to these use cases can improve their effectiveness.

D. System Requirements and Specifications

We have designed our requirements so the TrueTouch glove will be comfortable and usable.

Requirement	**Specification**	**Value**
Usable	Wireless	Yes
	Weight	<2 lbs
	Battery Life	>1 hr battery life
	Robust	Withstand 20N of finger force
Responsive	Latency	<100 ms

Table 1: Requirements and Specifications

The usability requirements are aimed at delivering a good user experience. Being connected by wires would limit the use space and tether down the user. A heavy, bulky system would decrease comfort. For battery life, we've noted that Oculus recommends a break every 30 minutes until the user gets accustomed to the headset [7]. So, an hour of battery life would be enough to accommodate new and experienced users. Latency can reduce user experience and can cause safety issues in more critical applications. So, we aim for at most 100ms of latency to ensure that the user will not perceive a delay in seeing the virtual object being touched and feeling the touch. Lastly, having a system that can withstand 20N of pulling force will ensure that the user will not break the system, even when squeezing an object.

II. DESIGN

A. Overview

For our solution we have two distinct subsystems: one that provides the feeling of shape and the other that provides the feeling of impact. These distinct systems can be seen in the hardware block diagram (Fig. 1) and the software block diagram (Fig. 2).

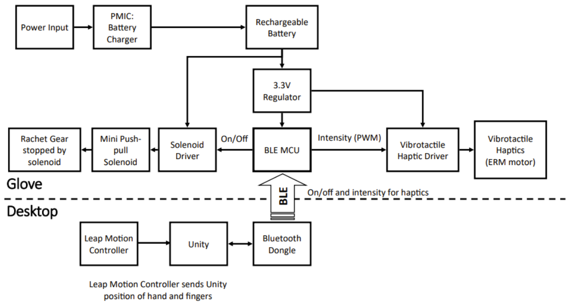

Figure 1: Hardware block diagram

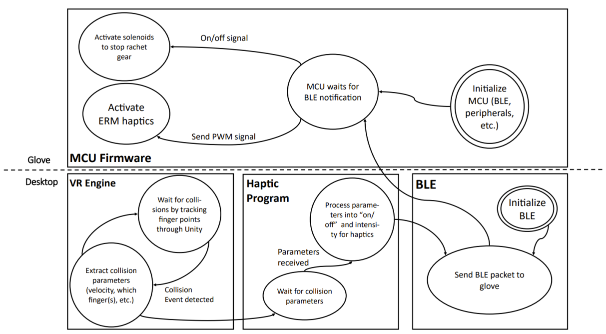

Figure 2: Software block diagram

The subsystem that provides the feeling of shape is located to the left of the "BLE MCU" block in the hardware block diagram (figure 1). This subsystem consists of solenoids that push into a gear. This subsystem, the microcontroller, PCB, and battery are housed in an enclosure that is strapped to the user's forearm. On the right side of the BLE MCU in Fig. 1 is the subsystem responsible for providing the feeling of impact. In our solution, we use vibrotactile feedback to simulate this sensation. More specifically, eccentric rotating mass (ERM) motors are used to provide vibrations. These ERM motors are located on a glove that the user wears, one on the tip of each finger and one on the palm. Since our solution is wireless, all of these components are powered by a rechargeable battery, seen at the top of Fig. 1. At the bottom is our virtual environment which is run on a desktop. The desktop has a Bluetooth dongle that communicates with the microcontroller. The virtual environment is created in Unity. Since our device does not provide hand tracking, a Leap Motion Controller is used to track the hand and fingers.

The software block diagram is shown in Fig. 2. This explains the communications between each part of our system. First, the device and computer are connected via Bluetooth Low Energy (BLE). Once connected, the TrueTouch device waits to receive messages from the desktop. Starting from the bottom left, Our VR game engine, Unity, waits for the user's hand or fingers to collide with an in-game virtual object. Once one of the fingers comes into contact with the object, Unity extracts the necessary parameters and sends a message though BLE to the microcontroller. When the microcontroller receives the message, it reads the message and activates the corresponding ERM motors and solenoids to provide haptic feedback.

B. Power Source

Our design utilizes solenoids that require 5 volts and 1 amp to operate, and ERM motors that vibrate on 2-3.6 volts with low current. The rest of our system just needs 3.3 volts to power on. Our initial thought was that we needed a relatively large power supply just to be able to keep the solenoids powered. Since we planned to use 5 solenoids in our final design, there will be a scenario where all 5 will need to be powered at the same time. This requires a steady current of 5 amps alone for the solenoids, which is not a practical method for a final product until a method is found to reduce that power requirement down to 5 volts and 1 amp. With this method, we only need a pulse from the power supply and not all 5 solenoids will be pulsed at the same time.

C. Bluetooth Module

To control the glove and communicate wirelessly with a computer, a BLE module is utilized. This project uses an nRF52832-based BLE module from Laird: the BL652-SC [8]. The nRF52832 is able to run both the Nordic BLE stack and a user's application on the same CPU core, eliminating the need for a separate MCU and BLE module. The BL652-SC module contains RF layout and circuitry needed for the nRF52832 all on a small and easy to use form factor.

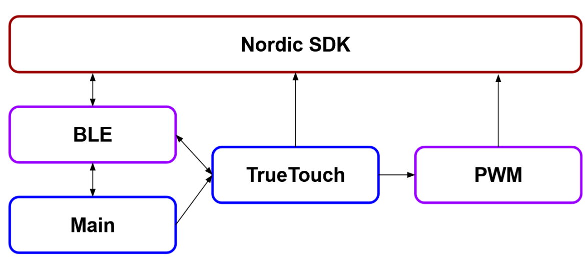

Figure 3: Firmware block diagram

This module is responsible for controlling all of the logic and hardware (solenoids and ERM motors) on the glove. The glove acts as a BLE GAP peripheral and starts advertising itself upon startup. It advertises its name “TrueTouch” and the Nordic UART Service (which it uses to emulate a wireless serial connection). The computer (acting as a BLE GAP central), scans until it finds the glove’s module and then initiates a connection. Once the connection is formed, the glove module acts as a BLE GATT server, hosting the Nordic UART service, allowing the computer to send commands to the glove.

The TrueTouch firmware is organized as described in figure 3. The TrueTouch module is written on top of the BLE UART, receiving commands and handling them. Nordic’s BLE and PWM are abstracted in the purple blocks to provide simpler interfaces for the application. The TrueTouch receives data asynchronously and processes it when serviced by the main loop. Currently it only handles two commands.

The first command is to set ERM motor’s intensities by setting PWM duty cycle. This command receives a set of fingers to modify and an 8-bit intensity (where 255 is max duty cycle and 0 is off). The PWM software module uses Nordic’s hardware PWMs to send distinct duty cycles out across 6 channels.

The other command is for “pulsing” the solenoids. It also receives a set of fingers to modify, and a 32-bit millisecond duration. The solenoids will, in turn, be set high for the duration then turned off again. The pulse sequence is started by setting a pin high and starting a timer. Once the timer expires, the active pin is turned off and if any more pins are waiting to be pulsed, the next pin is set high and the timer restarted. Pulsing only one pin at a time reduces the maximum instantaneous power draw. The pulse duration is sufficiently short that all solenoids appear to pulse at the same time.

Testing this module involved iterating through verification of both the software and hardware. Initial development was done on a development board with a source-compatible BLE module, which eliminated worry of hardware issues. The firmware was developed and tested for functionality on this board. After the firmware was initially verified, the module on our custom PCBA was tested by uploading the firmware and verifying its behavior. A combination of supporting hardware and software were designed for the testing process. This included a rig of LEDs to see signals activating, and GUIs that allowed fine-grained control of the commands sent over BLE. Detailed examples of this are shown in Appendix C.

D. Ratchet and Pawl

The feeling of shape is provided by strings in combination with a ratchet and pawl. The thought process behind this is that the length along the back of the finger is longer during finger flexion compared to finger extension. By routing a string along the finger, attaching it at the tip of the finger, and varying the length of the string, it is possible to restrict the amount of flexion the finger can perform. Going off of this, our system has 5 strings (one for each finger) coming out of a housing located on the forearm. It is important that the strings cannot slide off the back of the fingers, so there are guards to keep the strings in place. Inside the housing, there are 5 spools that hold the strings. As a finger flexes, the spool rotates and unravels, lengthening the string. When the finger transitions from flexion to extension, there is slack in the string. This is handled by torsional springs. Each spool has a torsional spring. The spring is responsible for retracting the slack on the string. Through testing, this torsional spring is strong enough to retract the string while also being weak to be unnoticable to the user. On top of the spool is a gear. Across the gear is a pawl connected to a solenoid.

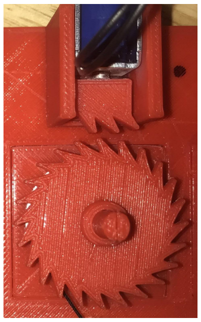

Figure 4: Ratchet and pawl.

When the solenoid extends, the pawl gets caught in the gear, preventing the spool from spinning, and string from lengthening. Since the string cannot lengthen anymore, the finger is prohibited from flexing any further.

A concern with the solenoids is that they may use too much power if they are extended for prolonged periods of time. To combat this, the teeth on the gear and pawl lock into each other and are held in the locked position by the user pulling on the string. Once the user stops pulling on the string (letting go of the virtual object), the solenoid will retract the pawl and the finger can move freely again. This way, the solenoid only needs to extend for a fraction of a second- enough for the pawl to catch on the gear.

E. Eccentric Rotating Mass Motors

Eccentric Rotating Mass (ERM) motors are used to provide the feeling of impact to the user. There are ERM motors embedded into the TrueTouch glove, one at the tip of each finger and one at the palm.

ERM motors were selected because they can vibrate intensely and are small in size. These ERM motors are connected to pulse width modulation pins on the PCB. This allows for varying vibration intensities.

F. Unity and Leap Motion Controller

To demonstrate our device, we decided to create a virtual world with objects in it that the user could pick up. We arbitrarily chose Unity, but there are other game engines that can be used to design VR demos. Originally, we were planning on using the HTC Vive headset; however, due to complications with the set up, we had to change that. Instead of having the visuals on a headset, we displayed the visuals on a monitor. Instead of using the HTC hand tracking SDK, we decided to use a Leap Motion Controller. The Leap Motion Controller is a USB device that uses IR light to track the hand. This combination is essentially the same as using a VR headset. Unity handles all of the collisions that happen between the virtual hand and object. In Unity, a script was put into place to detect when a finger comes into contact with an object. When this event occurs, Unity sends a message containing information about which finger and vibration intensity to the microcontroller via BLE.

III. THE REFINED PROTOTYPE

A. Prototype Overview

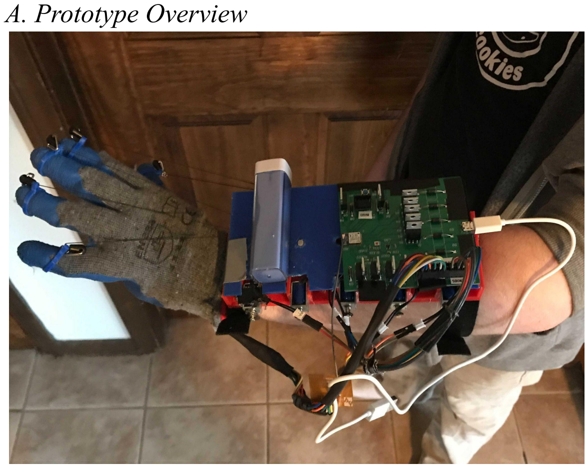

Figure 5: Entire TrueTouch system

Our end product is depicted in Fig. 5. It consists of a 3D printed housing strapped to the forearm. As previously mentioned, the enclosure contains five solenoids responsible for stopping a spool of string from spinning. Coming out of this enclosure are five strings; one string routed up the arm and back of the glove, and along the back of each finger of the glove. On the palm side of the glove are the six ERM motors, one at the tip of each finger and one at the palm. Also on the forearm enclosure is the PCB. The Bluetooth Module on the PCB is connected to the computer. The computer is running Unity. The microcontroller waits to receive messages from Unity.

B. Electronic Hardware Component

The main electronic hardware components for this project consists of a single printed circuit board, mini push-pull solenoids, and ERM motors. During our initial hardware design, we had to think of a method to stop the string from further extending. The idea of using push-pull solenoids was brought up so when the solenoids push out, the string will be stopped. However, the solenoids need a constant 5 volt and 1 amp supply to stay in the extended position. This power concern was fixed with the ratchet and pawl system which only needs a short pulse of 5 volts and 1 amp. We picked these specific solenoids because they were relatively small in size and require the least amount of power. As for our ERM motors, they were about half the size of a penny which allowed us to attach them to various points of the glove. The variable vibration strength was sufficient enough to simulate the feeling of impact as well. These ERM have an operating range of 2-3.6 volts and this was well within our power capabilities.

Our custom PCB came out to be approximately 3.5 x 3.6 inches. The main components of the PCB include the driving circuitry for the solenoids and ERM motors, linear LDO regulator, 24-pin connector, 5-pin connector, and Micro-USB connector. We picked a linear LDO regulator instead of a switching regulator because we only needed to step down 5 volts to 3.3 volts, and this was a simple and convenient solution that did not require extra circuitry. The 24-pin connector was used as a way to connect the solenoids and ERM motors to the PCB, and the 5-pin connector was needed to program the board. These 5-pins on the PCB were connected to SWDIO, SWDCLK, NRESET, 3.3 volts, and ground of the BL652-SC. We decided to not have the power source on board because the size of the 5 volt, 1 amp rechargeable batteries appear to be too obstructing to the user's view, as well as affecting the overall aesthetic. Instead, we are using an external rechargeable battery that is lower profile and easier-to-hide. It connects to the PCB with the onboard Micro-USB connector.

After completing the schematic for the PCB, we had to consider where to place certain components when designing the layout. Since we are using BLE to send/receive signals, the placement of the BL652-SC plays a factor in how well the antenna radiates. It is suggested by the datasheet that we put the module as close to the edge as possible to avoid noise. However, in our specific case we are using an external antenna. This means that we can be slightly more lenient with that placement suggestion. The pin connectors are both located near the edges as well so they can be accessed easier. Our design has a few redundant parts due to the nature of how many solenoids and ERM motors we are using. There are 11 driving circuits from the combined 5 solenoids and 6 ERM motors. The circuitry for those are all grouped together in an organized fashion so that it may facilitate assembly and make things a lot more identifiable after manufacturing. Fig. 5 shows the layout of our PCB before adding ground pours.

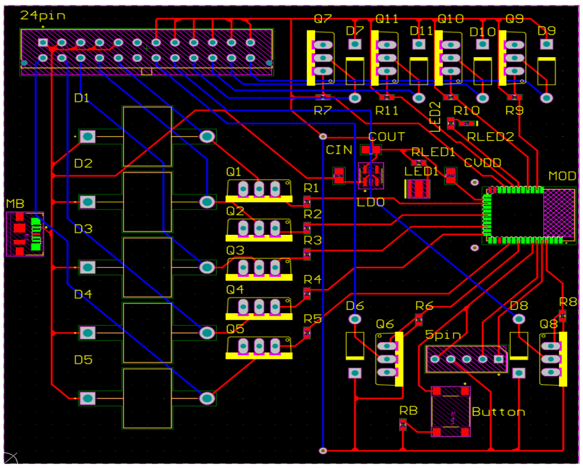

Figure 6: PCB Layout

C. Product Functionality

Our prototype was mostly functional by CDR. We had most of the blocks satisfied in our diagrams. The user's hand is able to be detected by the Leap Motion Controller and Unity processed the information when an object was touched. The BLE module then receives this information wirelessly and tells which solenoid/s and ERM motor/s to power on. We had the concept and functionality working, but the hardware and design still needs to be migrated over onto the glove and a housing compartment design needs to be finalized and polished for our solenoids and PCBA.

D. Product Performance

TrueTouch was able to simulate the feeling of shape and impact of an object. This is hard to quantify but through our testing, we were able to close our eyes and it felt like there was an object in our eyes.

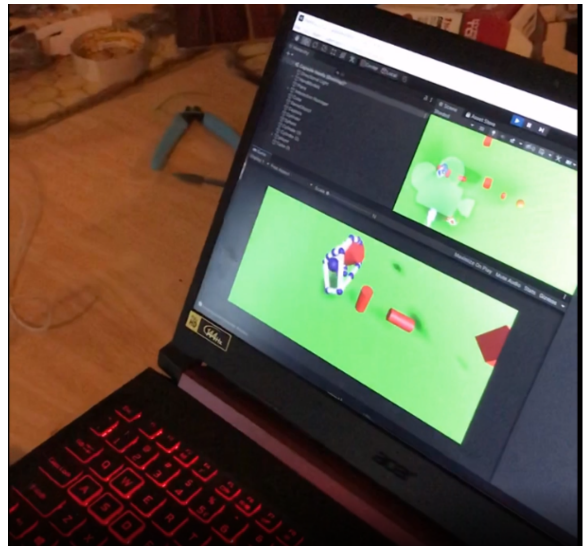

Figure 7: Wrapping fingers around a cylinder shape in Unity

Figure 8: Corresponding hand position of figure 7 in the real world.

An example of this can be seen in Fig. 7 and Fig. 8. In Fig. 7, the virtual hand has the fingers wrapped around a cylinder. At the same time in Fig. 8, it can be seen that the real hand is in the shape of a cylinder. There were multiple objects in our unity scene and we were able to grab objects with different fingers. When grasping objects where some of our fingers are occluded by others, the virtual hand would sometimes not align with the object. However this is a limitation on how well the leap motion controller was able to track our hands rather than a limitation on our TrueTouch system.

Our wireless specification is met due to our system design. The weight was met by using appropriately sized components and 3D prints. To measure this, we utilized a scale and found the system to only weigh 0.82 lbs altogether. So, we've amply met the requirement of being less than 2 lbs. To test the 20N requirement, we were able to attach a

5 pound weight to the end of one of the strings while it was locked. 5 lbs should exert approximately 22.2N, and since the system withstood this weight, this requirement is met.

Testing the battery life is difficult because the power consumption depends on the exact use of the device. In order to test this we've reviewed the product in a worst-case scenario where the motors are constantly active and the solenoids are constantly being pulsed. If the device were run for an hour, approximately 16.8% of the battery life would be used by the electronics on the PCB and the ERM motors. With the remaining 83.2% of the battery life we would be able to actuate the solenoids approximately 411,910 times. However, since the solenoids are designed to pulse only one solenoid every 16 ms, the most actuations you could achieve in 1 hour under constant pulses would be 225,000. So, we consider the hour long battery life requirement met.

Finally to test the latency, we had to calculate the latency path and measure each component in this path. The path of latency we measured includes the BLE communication in one direction, the MCU response time, and the solenoid actuation time. Specifics of how these were tested are given in Appendix C. Overall, we are approximately 85% confident that our device will fall within the 100ms latency spec. Unfortunately, BLE communication has a very high variance, and despite theoretical throughput of more than 1Mbps, we see very high BLE round-trip times (RTT) of approximately 155ms (corresponding to a 77.5ms communication in one direction). Since the communication itself shouldn't take this long, there are likely delays in the operating system and Unity game engine that cause this high RTT. Windows does not provide a good BLE interface, so we were unable to fine tune the BLE connection parameters in order to try to get improved BLE times. Due to this, this specification is considered partially met.

IV. CONCLUSION

We met our goals for MDR, CDR, and FPR. For MDR, we focused on creating the mechanics of our glove like the ratchet and pawl and housing. We created a simple python interface to manually pulse the solenoids. For CDR, we focused on getting a fully integrated system. We set up a unity scene with a cube and placed proximity detectors on the fingers. Once Bluetooth was set up, we were able to send the desired bits corresponding to the collision to our microcontroller on the glove. While this was being tested we also designed a custom PCB, which we transitioned to for FPR. During FPR we demonstrated a fully functioning system which achieved all of the system specifications that we originally set for ourselves.

ACKNOWLEDGEMENTS

We would like to acknowledge and thank our advisor, Professor Anwar. She has been tremendously helpful in keeping us on track and providing aid whenever we need it. We would also like to acknowledge and thank our evaluators, Professor Ciesielski and Professor McLaughlin for their feedback. Lastly we would like to thank the coordinators, Shira Epstein, Baird Soules, and C.V Hollot for organizing and leading Senior Design Project 2021.

REFERENCES

[1] A. Yates, “Efficacy of VR in Rehab: Principles of Neuroplasticity,” *Neuro Rehab VR*, 22-Sep-2020. [Online]. Available: https://www.neurorehabvr.com/blog/efficacy-of-vr-in-rehab-neuroplasticity. [Accessed: 09-April-2021].

[2] C. Fang, Y. Zhang, M. Dworman, C. Harrison, “Wireality: Enabling Complex Tangible Geometries in Virtual Reality with Worn Multi-String Haptics,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1-10, doi: https://doi.org/10.1145/3313831.3376470.

[3] “VRgluv | Force Feedback Haptic Gloves for VR Training.” VRgluv. https://www.vrgluv.com/. (accessed Apr. 8, 2021).

[4] D. Chang, J. Hopfenblatt, P. Edara and B. Balakrishnan, "Immersive Virtual Reality Training for Inspecting Flagger Work zones," 2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), Utrecht, Netherlands, 2020, pp. 327-330, doi: 10.1109/AIVR50618.2020.00066.

[5] Q. Chen et al., "Immersive Virtual Reality Training of Bioreactor Operations," 2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Takamatsu, Japan, 2020, pp. 873-878, doi: 10.1109/TALE48869.2020.9368468.

[6] J. Ulmer, S. Braun, C. -T. Cheng, S. Dowey and J. Wollert, "Gamified Virtual Reality Training Environment for the Manufacturing Industry," 2020 19th International Conference on Mechatronics - Mechatronika (ME), Prague, Czech Republic, 2020, pp. 1-6, doi: 10.1109/ME49197.2020.9286661.

[7] Oculus, “health and safety warnings.” [Online]. Available: https://www.oculus.com/legal/health-and-safety-warnings/. [Accessed: 09-Apr-2021].

[8] BL652-SC nRF52832 BLE Module, Data Sheet, *Laird Connectivity*, http://assets.lairdtech.com/home/brandworld/files/Datasheet%20-%20BL652.pdf.

[9] “Haptic gloves for virtual reality and robotics,” 15-Mar-2021. [Online]. Available: https://haptx.com/. [Accessed: 09-Apr-2021].

[10] nRF52840 Feather Express, Product Page, *Adafruit*, https://www.adafruit.com/product/4062.

APPENDIX

A. Design Alternatives

When originally coming up with this wearable device, we considered many methods of restricting the finger’s movements. Different methods affect the product's complexity, size, and price.

When we decided on this problem statement, we already had an idea of what was on the market. We knew of the products that were exoskeletons that the user wears, such as the HaptX [9]. This concept of an exoskeleton however, was not something we felt comfortable in implementing. If we were to design an exoskeleton, we would need a lot of mechanical knowledge. We decided to steer away from this idea, since we wanted to focus more on computer and electrical engineering concepts. From online demonstrations and images, these exoskeletons are also very bulky. Compared to the HaptX, the TrueTouch is lighter, smaller, and solves a similar problem. Limitations of Truetouch compared to exoskeletons are that it cannot vary resistance and cannot track the hand. With an exoskeleton, it is possible to vary the amount of resistance against the fingers bending. TrueTouch has two states: free movement and stop finger flexion.

For TrueTouch, we decided to use the Leap Motion Controller to track the hand and fingers. If the hand is occluded or out of view of the Leap Motion Controller, hand tracking for TrueTouch would be unavailable. Exoskeletons have hardware on the glove that does hand tracking. This tracking is more reliable as it does not rely on having the hands in the sight of any sensor. We could have implemented hand and finger tracking through potentiometers on the spools of the string, however, we decided it would be extraneous. Newer models of VR headsets are starting to implement native hand tracking therefore there is no need for the device to have handtracking too.

Truetouch takes inspiration from Wireality the most [2]. This paper is where we got the idea of using strings to restrict movement. By using strings we were able to keep the cost of our product low. While it may not seem like it from our expenditures, our end product is fairly cheap to reproduce. Many of the parts can be 3D printed or bought. Most of our expenditures are for shipping and PCB, which could be reduced significantly. Truetouch and Wireality are very similar in implementation but differ in their goal. Wireality focuses more on the movement of the arm. The ratchet and pawl mechanism is on the shoulder which controls the movement of the arm in relation of the shoulder. When touching a virtual wall, the hand would be stopped from going through it. Conversely, TrueTouch has the ratchet and pawl on the forearm, which does not prevent the arm movement. TrueTouch is more focused on the movement of fingers, specifically holding objects.

B. Technical Standards

The Laird BL652-SC module that we use is in compliance with several technical standards. The module is pre-certified as compliant with various regulations. These include FCC certification and CE certification. These certifications represent that the module’s RF output is within acceptable limits and passes various safety and environmental criteria. Having these certifications means that our hardware operates within acceptable parameters mandated by these regulatory bodies and can even be sold in consumer markets.

This module also employs Bluetooth Low Energy, a standard for wireless communication that is developed by the Bluetooth Special Interest Group

(SIG). Bluetooth was previously standardized as IEEE 802.15.1, but is no longer maintained by IEEE. Bluetooth Low Energy was introduced in the Bluetooth 4.0 specification by the Bluetooth SIG. BLE support is incredibly common in devices nowadays, especially for the short range, low bandwidth, wireless communication needed for our project. Using this standard for our project will allow much easier adoption of the glove into existing systems since many of these systems will have built in support for BLE.

C. Testing Methods

Testing the glove was done in an iterative process. First, a known-good piece of hardware with a source-compatible BLE module was used as the base for software development. This development hardware was the Adafruit Feather nRF52840 Express [10], which had an nRF52840-based BLE module on it. Once the firmware was confirmed to be working correctly on the development hardware, it was flashed to the BLE module on our custom PCBA and its functionality was verified with the same set of tools developed for testing the software on the development hardware.

The glove firmware mainly consisted of three separate components that were developed and tested in isolation:

- BLE
- PWM
- GPIO

All of these were developed by starting from known-good examples provided with the Nordic SDK and adapting them to our project then cleaning up any unnecessary pieces of code left over in them.

For the BLE communication, the Nordic UART service example was adapted to provide a simple interface for the firmware's main application to use. It was tested using Noridc's nRF Connect mobile application, which is specifically designed for testing and verifying BLE designs. With the mobile app we could scan, connect, discover services, and communicate with the remote device. Once these 4 steps were able to be successfully performed, the BLE component was considered complete.

The PWM also derived from an example provided with the Nordic SDK. The GPIO functionality was trivial enough that no example was needed and the Nordic SDK's GPIO functions could be called directly instead of requiring an intermediary layer of software to simplify them.

Initial verification of the PWM required an oscilloscope to assert that the frequency and duty cycle was what it was expected to be. Then, to test both the PWM and GPIO, an LED test rig (shown in figure 8) was created that simply featured 11 LEDs (5 representing the 5 solenoids and 6 representing the 6 ERM motors). These were connected to free pins on the development hardware and allowed quick and simple visual verification that things were working.

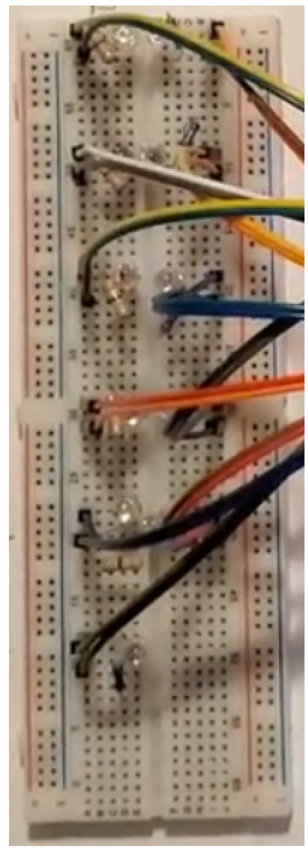

Figure 9: LED test rig with LEDs for the ERM motors (left half of breadboard) and solenoids (right half of breadboard)

To assist with testing of all components integrated together (the BLE, PWM, and GPIO), several GUIs were created that allowed full control of the device by allowing users to select and set all parameters for messages to send. The final GUI is created in Unity and integrates the Unity BLE communication layer used to communicate with the TrueTouch device in our project.

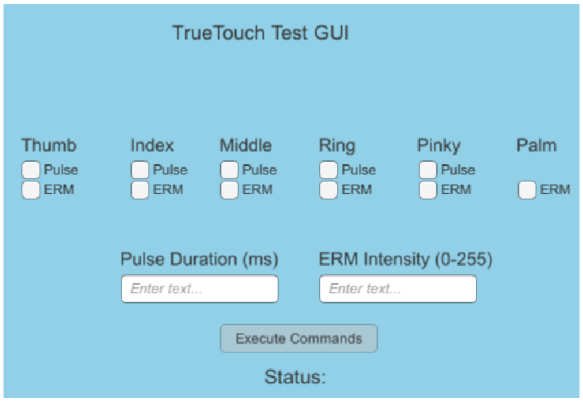

Figure 10: Unity TrueTouch test GUI

For testing the latency, several calculations were needed. We calculated the RTT for BLE communication (and used half of this as the transmission time in one direction), the time the MCU spends processing, and the time for the solenoid to actuate.

To calculate the RTT we added temporary debug statements in both the Unity code and the MCU firmware. The MCU firmware was updated to echo every message it receives back to the sender (acting as an ACK). The Unity script was updated to record timestamps when messages are sent and received, then calculate RTT as the difference in these timestamps. We had the following statistics for RTT:

Mean	155.42
Standard Deviation	18.80
Min	101.50
Max	185.40

Table 2: BLE round trip time statistics

Measuring the time for the solenoid actuation was difficult, so instead we calculated the time for it to take using information from the datasheet. Given a mass of less than 12.6g (we care about the plunger mass, but don't have it so use the entire mass of the solenoid to get an overestimate of actuation time), an actuation force of 80g (0.78N) and a distance to travel of 3mm:

$$a = \frac{F}{m} = \frac{0.78}{(12.6 \div 1000)} = 62.26\ m/s^2$$

$$\Delta d = 3 \times 10^{-3} m = v_i t + 0.5 a t^2$$

$$\Rightarrow \frac{3}{1000} = 0t + 0.5(62.26)t^2$$

$$\Rightarrow \mathbf{t = 9.81\ ms}$$

Altogether then, the average latency is 87.65 ms. In the worst case, we've seen 102.85 ms. Given the average and standard deviation, we expect the system to be within spec approximately 85% of the time.

D. Project Expenditures

For the TrueTouch glove, our team was given a budget of $500 for both the prototype and final product design stages. During the prototype design phase of the project, our budget was allocated towards buying the necessary hardware for the glove's construction. This includes 7 Adafruit push-pull solenoids, 12 ERM motors, a pair of Ringer R507-09 Duty Gloves, and 3 Adafruit Nordic nRF52840 Express Feather MCUs. Additional parts were bought for the listed hardware for all team members to have due to our team being separated during the course of the construction. Overall, this first phase of the project had a total expenditure of $176.34, including shipping fees.

The final product design phase had our budget focused on PCB design and implementation. Specifically, the budget was allocated for PCB printing and component acquisition (including all solderable components and external hardware, such as a 5 volt, 1 amp external battery for wireless capabilities). Including cost of shipping, the total expenditure for both phases of the project came to $338.66. This leaves a remaining budget of $161.34 for the TrueTouch design.

E. Project Management

Due to COVID-19, our team was unable to be together to work on the Senior Design Project. To combat this, we divided responsibilities amongst team members based on expertise to complete our project. Alex Dickopf was responsible for the construction and testing of the entire system because he is close to campus. Anthony Chan was in charge of hardware design since he is the only electrical engineer in the group and is most adept with circuit construction. Cameron Kluza was responsible for the firmware because he has had the most experience with microcontrollers. Jonathan Yip was responsible for the integration with Unity and the team website. Although these were the assigned responsibilities, each member assisted other team members when needed.

Overall, the team worked well together. We met every deliverable we set upon ourselves for MDR and CDR. Our project is not easily demonstrated in a remote situation, however our group was able to present our project sufficiently during MDR and CDR. Coordination was done through Slack and Zoom. Weekly meetings with our advisor, Professor Anwar, were scheduled to keep us on track.

Alex has a good understanding of the overall system and 3D print design. Anthony understands anything related with the PCB design and power systems. Cameron wrote firmware for the microcontroller and established the communication between the desktop and microcontroller. Jonathan used Unity to create a virtual world and to detect hand-object collisions. Additionally Jonathan created the website for the team and kept it up to date.

F. Beyond the Classroom

A. Anthony Chan

My main responsibilities for this project was the initial circuit design of the prototype for MDR as well as the design of the PCBA. The solenoids and ERM motors needed to be driven by some circuitry with components that required specific specifications. A reference from the solenoid product page was used in helping our design. I was familiar with how this reference driving circuit worked as this was something I saw frequently in class.

Soon after MDR, the task of designing and ordering the PCBA was at hand. I had no prior experience with Altium or anything related to designing PCBAs for that matter, but was confident because there were a lot of resources available for me to learn. The learning process for me was watching the M5 tutorial videos on YouTube. Most of the time spent designing the PCBA was not actually in Altium, but more so on reading datasheets and deciding on which components to use. In the datasheets there is important information that needs to be noted such as the rated voltage/current, size, and if the component needs additional circuitry. I learned a lot about what certain components do and a lot other terminology regarding electronic parts. Since there are other parts that are external to the board, I had to consider how it connects to the PCBA and a convenient way of providing enough power. When doing the layout, I was tasked with how well I was able to organize everything and follow best practices. After spending a bit of time completing the layout, I was unsatisfied with how it looked in terms of component placements. I redesigned the entire layout until everything was neatly grouped and the size was as small as I could get it with the components we had. The PCBA was then ordered and Cameron was able to assemble and test it.

The current PCBA could definitely be further improved with cheaper and smaller components which would result in a smaller sized PCBA. However, it is fully functional and already fulfills our needs in this project.

From the teamwork to designing in Altium, I believe this valuable learning process during the difficult pandemic will be extremely useful in my professional life. Learning and using Altium actually sparked my interest in pursuing more work in designing PCBAs and could be a potential future career for me.

B. Alexander Dickopf

My main responsibilities on the team included the integration of hardware and software for the TrueTouch during each design phase, as well as being in charge of our budget for the project. As a computer engineering student, I have been interested in embedded system design, as well as the realm of VR in engineering. However, despite being a computer engineering student, (excluding my projects I have done for my courses) I have not done many independent or home projects. As a result, this kind of undertaking, as well as its size, was daunting at first glance.

At the start of the project, I had to refamiliarize myself with the Arduino IDE when programming test code for our MCU, as I have not used the Arduino IDE that much before. Thankfully, Arduino provides great documentation (as well as being based on C) on their website. With the help of my teammates, the basic circuit and MCU was both created and programmed as intended, leading to our first prototype design.

I also did not have much 3-D printing experience leading into this project. While I had basic experience with 3-D printing in the past, it was all from analysis and looking, rather than actually designing and printing. As Jonathan was in charge of the digital design for the early stages of the project, my collaboration with him in both printing on my end and going back and forth on design changes has helped both my sense of robustness in design and actual application in CAD software.

In the later stages of the project, I am thankful to have learned more about game design (specifically basic C# scripting and integration of different applications, such as the Leap to the Unity game engine). In helping Cameron with the integration of the system, I gained a lot of valuable from him as well as in my own experience that can be carried over to different areas or projects.

Lastly, my role as the budget lead for the team and being in charge of the full integration of each subsystem has given me valuable management and leadership skills. Budget management required careful planning in balancing both component usefulness (as per our requirements) and cost effectiveness, as well as figuring out how many of each item to buy. The subsystem integration required a similar train of thought, with a specific focus on planning out due dates and reporting data and results after integration. All in all, this project has given me both technical and soft skills that I can take forward in my future ventures that I am very proud to have acquired.

C. *Cameron Kluza*

My main responsibilities were the firmware for the microcontroller and the BLE communication between the microcontroller and the desktop. In creating tools for the desktop to communicate with the microcontroller, I was forced into a very unfamiliar environment, working with Windows APIs and Unity development. I had to take time to learn a bit about the Windows development environment and the Unity editor at various layers in order to get things to start working properly. Official documentation available from Windows and Unity were very helpful throughout this process. Moreover, various example projects and libraries provided by Windows, Unity, and miscellaneous helpful users on Github, were very helpful to get started on practical implementations of the code needed for our own project.

And of course working with the microcontroller gave me the opportunity to keep learning more about so many aspects of embedded development. I learned various pieces of info about the Segger and Nordic development tooling (such as how to properly set up a git project with the Noridc SDK), more specifics about the nRF52 series (such as how their PWM works), and more specifics about debugging CPUs running multiple threads of execution. The last point introduced me to “monitor mode debugging” where essential functionality isn’t stopped when other components are being debugged. This was necessary on the nRF52 chips so the BLE stack could keep running even if I used breakpoints to debug my application code.

The embedded knowhow I learned will directly apply in my professional life as an embedded engineer. The pieces of info I learned about BLE communications on Windows and Windows in general will almost certainly come in handy at some point in my career. Whether or not what I’ve learned about Unity will help me is unclear, but it was still a worthwhile and interesting tool to learn!

D. *Jonathan Yip*

My main responsibilities were designing a virtual world in Unity and creating the website. I was forced into unfamiliar territory. I have always wondered what goes into game design. Although our demonstration world is very simple, I was able to get a preview of how to use Unity to design a game. I referenced many tutorials online as well as the Unity documentation to get it running.

When creating the website, I had no idea where to start. I was originally going to go to a website builder to create the website, however I decided to challenge myself. I decided to learn HTML and CSS through various sources. In the end I was able to create and design a website that contains words, pictures, and embedded videos.

Working with a team was also something I learned. Since everything was remote, our team had to figure out how we would approach our project. Everyone on the team had different schedules and priorities which we had to account for. Sometimes our individual parts did not work when put together. Debugging these issues made me realize that working with a team is much different than working with myself. When I do things myself, I know all the intricacies of the product. Now that everything is split amongst group members I had to learn how to communicate ideas I had with the group.

There is no guarantee that I will use these specific technical skills in the future. However, the skills of struggling with something new and teamwork will stick with me forever.

E. Additional Hardware Documentation

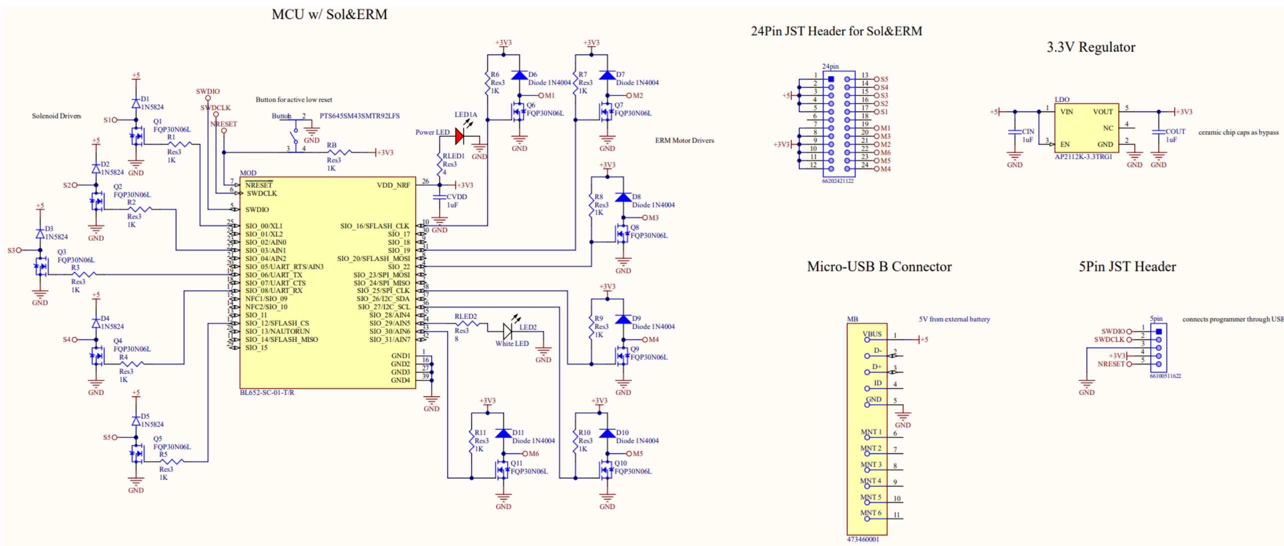

Figure 11: PCB Schematic

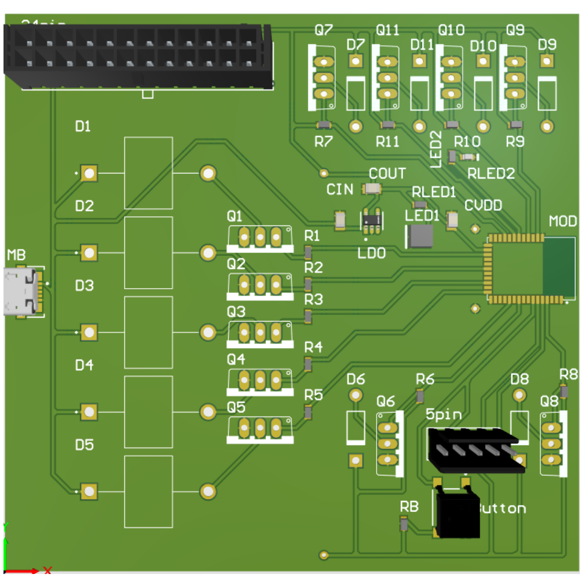

Figure 12: PCB Top Layer

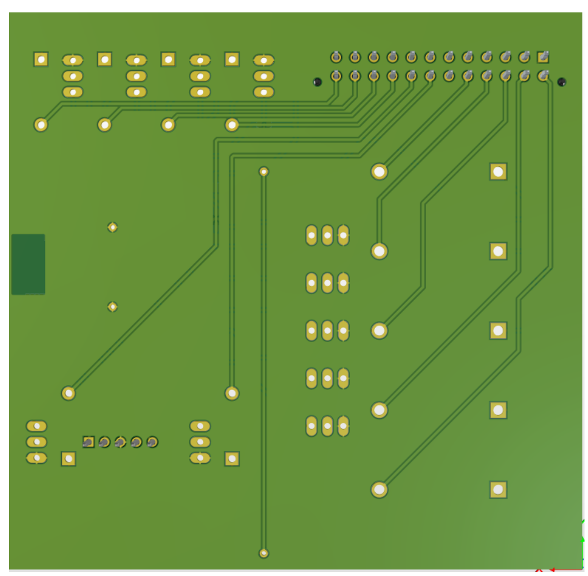

Figure 13: PCB Bottom Layer