# Virtual Reality Glove Project Write-Up

Zakhar Kanyuka

December 8, 2019

# Contents

# 1 Introduction

Some of the limitations of modern consumer-grade virtual reality (VR) systems include restricted spatial representation of the user's body and only rudimentary haptic feedback. Successful systems from Oculus and HTC offer infra-red camera-based spatial tracking of the user's hands and head. Additionally, they extrapolate the user's hand conformation from the signals from the triggers and touch-sensitive elements found on the controllers. As a result, the user's freedom of manual movement that is represented in the virtual environment is confined by the shape and features of the controller. Furthermore, both major companies' controllers offer no haptic feedback besides a vibration motor and the shape of the controller. While many developers have been successful in making engaging VR experiences that help the users forget about hardware's limitations, a more comprehensive manual interface than the existing controllers can be an upcoming step in refining VR products.

In this project, I set out to build a prototype of such a manual interface in the form of a glove. For each digit, I implemented respective orientation tracking, haptic feedback, and contractile force assessment mechanisms. To track the each digit's three-dimensional orientation relative to the palm of the hand, I used gyroscopes mounted on each digit and one on the back of the hand. The one-dimensional haptic feedback mechanism restricts digit contraction with the help of servomotors mounted on the user's lower arm. The contractile force exerted by the user against the haptic feedback mechanism is measured using ammeters which assess the current drawn by each servomotor to maintain position. The management of the transducers and the communication with the VR host device are mediated by an Arduino MEGA microcontroller board mounted on the user's upper arm. Full VR integration of the glove relies on spatial tracking facilitated by attachment of an existing VR system's controller. A companion script for Unity3D provides a framework for future integration in VR experiences. Other goals of the project included maximizing hardware reliability and inter-user fit.

## 2 Inspirations

A glove is hardly a new format for a controller in the world of VR and similar applications. Glove-based peripheral prototypes such as the Sayre glove developed at the University of Illinois have existed since the 1970s. The first VR-centric glove was released by VPL Research in 1987, a company at the centre of early VR development. More famously, in 1989, Nintendo released the Power Glove as a separately sold companion for their contemporary NES system. However, most of these projects saw little success due to prohibitive cost and limitations of the available technology. Furthermore, most focused solely on representation of the user's hand in virtual space with no haptic feedback system. More recently, small-scale companies such as CyberGlove, HaptX, and VRfree have put out impressive prototypes of gloves that offer feature sets akin to my project. However, these typically have heavy, unwieldy, or complex exoskeletons through which they apply haptic feedback. Most such projects are little-reviewed and have yet to reach the marketplace. For a more comprehensive review of the above and other glove-based peripherals, please see my past work, *The Ultimate Controller* [3].

Despite commercial failures of similar peripherals in the past, a haptic VR glove continues to be an appealing interface for enhancing entertainment, training, and telepresence applications of VR technology. This inspired me to build the first version of a VR glove prototype in early 2019. Most notably, the glove used a set of flex sensors to assess contractile position of each digit, a set of small hand-mounted servomotors to exert haptic anti-contractile feedback, and conductive stretch-sensing cord to sense the user's resistance to haptic feedback. While successful as a proof-of-concept, the glove suffered from poor structural integrity, only a single degree of freedom per digit in assessing contractile position, and excessive signal noise from stretch-sensing cord. The development of this project is described in detail in *The Ultimate Controller* [3]. The prototype described in this paper is meant to be a progression of the aforementioned project, addressing its flaws. Similar to the previous project, the present prototype differentiates itself from other offerings with a combination of the lack of an exoskeleton (an advantage as exoskeletons tend to make gloves heavy and unwieldy or unnecessarily complicated), 3 degree-of-freedom digit tracking, and a current-based assessment system for user-applied force.

# 3 Design and build process

## 3.1 Overall structure

The glove's construction consists of three main units: the microcontroller unit, which is mounted on the user's upper arm; the servomotor unit, which is mounted on the user's lower arm; and the hand unit, which is mounted on the user's hand. The microcontroller unit, held to the user's arm with an elastic armband, includes the Arduino MEGA microcontroller with a custom shield that facilitates a one-cable link to the electronic components of the other two units. The servomotor unit, held to the user's arm by a short sleeve includes the servomotor control board which powers and controls the servomotors and assesses the current drawn by each. Additionally, the servomotor unit has attachment points for each servomotor. Lastly, the hand unit, held to the user's hand by a tight-fitting work glove includes a gyroscope management board, a haptic feedback application bracket, and a haptic feedback harness. The following subsections will describe each of these units, the connectivity between them, and the servomotor subunits in more detail. A photo of the project in its present state can be found in Appendix A.

I built the electronic boards described on the base of consumable flexible perfboard material. I used standard male and female header pins to build cable connectors. All electronic connections other than the cable connectors were reinforced with solder. All structural connections were reinforced with hot glue.

## 3.2 The servomotor subunits

The haptic feedback mechanism employed by the prototype uses servomotors to pull the user's digits via metal cables opposite to the digits' contractile direction. The system is powered by a set of five TOWERPRO SG-5010 servomotors, one for each digit. To harness the power of the servomotors, a cable was attached to the far end of each servo's arm, and channelled through one of the built-in mounting holes. A 180° rotation of the servo arm results in a 4 cm range of motion for the cable. To allow flexibility in mounting the servos, the force is conveyed through a 30 cm derailleur cable. To summarize, a servo subunit consists of a servomotor, a derailleur cable with one end loose and the other end attached using hot glue to the mounting hole that serves as a cable channel, and a cable that is attached to the arm and passes through the derailleur cable. In the assembled prototype, the loose end of the derailleur cable is glued to the eyelets of the haptic feedback application bracket on the hand unit, and the servos are flexibly mounted at the servomotor unit using Velcro.

## 3.3 Inter-unit connectivity and power

The microcontroller unit serves as the bridge between the transducers, the power source, and the host. The Arduino MEGA is connected to the host computer using a USB type A/B cable, and it is connected to a 5 volt, 4 amp power adapter through the barrel-shaped power jack.

The electronic components of the three units are connected using two propitiatory ribbon cables. The servomotor subunit is connected to the microcontroller subunit using a 15-pin male-male ribbon cable, which I will refer to as cable A for the rest of the paper. Cable A has 5 PWM lanes (used for servo control), 5 analogue lanes (used for ammeter readings), SCL and SDA lanes for I2C communication (for gyros), a single free digital lane for future expansion, a Vin lane, and a ground lane. The connector for cable A is L-shaped to avoid the possibility of accidentally plugging it in the wrong orientation. The microcontroller subunit and servomotor subunits have corresponding female connectors for cable A. The custom shield for the Arduino MEGA serves as the microcontroller's adapter for cable A. The electronic components of the hand unit are connected to those on the servomotor unit (and by extension, through cable A to the microcontroller unit) via a 5-pin male-male ribbon cable which I will refer to as cable B. Cable B has SCL and SDA lanes for I2C communication, a single free digital lane for future expansion, a Vin lane, and a ground lane. Like cable A, it is L-shaped and has corresponding female connectors on the units that it links. The servomotor unit serves in part as a pass-through for the microcontroller's connection to the gyros. I formed the connectors using ribbon cable, male and female headers (where appropriate). Then I reinforced and insulated the male connectors using hot glue. It should also be noted that the Vin and ground connections make up system-wide power and ground rails.

## 3.4 Hand unit

The hand unit is held to the user's hand through an elastic work glove. I chose a tight-fitting glove for the project to ensure close adherence to a wide range of hand sizes. The tips of each fingers are reinforced with a thimble fashioned out of wire and hot glue. Cables attached to the thimbles are threaded through the glove along the outer edge of each digit so they could be attached to the servomotor subunits' cables at the haptic feedback bracket. This local cable system makes up the haptic feedback application harness. Velcro mounting points for gyroscopes are placed on middle phalanges of each finger, and the proximal phalanx of the thumb. Positioning the gyroscopes on these joints best allows for extrapolation of finger conformation with a single sensor per digit. The Velcro haptic feedback bracket attachment point spans the back of the glove. The extensive use of Velcro throughout the arm hand unit allows easy serviceability and inter-user adjustment.

The haptic feedback bracket is a metal structure with a flat base parallel to the user's metacarpals and eyelets that serve as attachment points for the loose ends servomotor subunits' derailleur cables. The end of the bracket closest to the phalanges has five functional eyelets through which the distal ends of the servo subunits' derailleur cables are threaded. Here, adjustable links between the servo-attached cables and harness-attached cables is made by twisting together the ends of each. Thus, the servos are mechanically linked to the tips of the user's fingers, completing the haptic feedback system.

Early design of the project relied on the possibility of 3D printing structural pieces for the hand unit. These include gyro mounts with lobes when one can drill pass-throughs for haptic feedback harness cable for each digit. The model for these can be seen in Appendix F. Additionally, I planned to 3D print a lightweight, plastic haptic feedback bracket with a shape moulded by the user's hand. When printing initial prototypes for the gyro mounts revealed the limited availability of 3D printers, the design was changed to what can be seen in the final product, and no further part prototypes were made.

I used hot glue to mount the gyroscope communication board atop the haptic feedback bracket. The schematic diagram for the board can be found in Appendix C. The board consists of five Adafruit L3GD20 3DOF gyroscope breakout boards permanently connected to the communication board through 4-line ribbon cables. Each such cable has SCL and SDA lanes for I2C communication, a Vin lane, and a ground lane. Additionally, an Adafruit 9DOF sensor breakout board which includes a L3GD20 gyroscope has been mounted directly on the gyroscope communication board. It can serve as a reference point for the other gyroscopes. Its additional accelerometers have not been implemented in this project, but remain an open for future development. The electronic components implemented on the gyroscope communication board use I2C for communicating with the microcontroller. Because all L3GD20H gyroscopes used share the same I2C address, they are managed with an Adafruit TCA9548A I2C multiplexer digital breakout board. Each sensor's I2C pins are linked to respective corresponding pins on the multiplexer. All of the breakout boards use the shared Vin and ground rails. As previously mentioned, this unit connects to the other units through cable B.

## 3.5 Servomotor unit

To facilitate mounting on the lower arm, a short sleeve fashioned out of another elastic work glove is used as the structural base for the servomotor unit. I mounted the servomotor control board on a piece of cushion foam with a base that is shaped to follow the curvature of the user's arm. This strategy enhances structural stability when mounting the large board to a user's body. Throughout the circumference of the sleeve, I placed Velcro attachment points for the servomotors. When the unit is assembled, the servomotors are connected to the appropriate headers on the servomotor control board.

The servomotor control board integrates two key functions of the prototype - management of haptic feedback and assessment of resistive contractile strength. The schematic diagram for the board can be found in Appendix D. The board has five three-pin male headers common among shields made for servomotors. Each header has ground pin, a Vin pin, and a PWM communication pin. The five INA169 current sensors serve as a proxy between the servo headers' Vin pins and the global Vin power rail. This allows them to monitor the current drawn by the servomotors. The current sensors have Vin, ground, and analog signal pins, as well as the pins between which the current is measured and travels to the servos headers. The servo header PWM pins and the current sensors' analogue pins make a direct and exclusive connection with the the corresponding pins on the microcontroller through the cable A link. The board additionally serves as a pass-through to the hand unit, to which it is connected via cable B.

### 3.6 Microcontroller unit

The Arduino MEGA microcontroller board is mounted to the user's upper arm with an adjustable elastic armband. The outward surface of the custom-made shield consists solely of the female connector for cable A. The schematic diagram for the shield can be found in Appendix E. On the flip-side, the female connector's pins are connected as follows: five analogue lines used for current sensors are connected to analogue pins 0-4; five PWM lines used for servo control and the expansion digital pin are connected to digital pins 2-6 and 7 respectively; and the Vin rail and ground rail lines are connected to Vin and ground pins respectively. As explained before, this unit is used for handling communication to the host via USB, managing the supply of power, and mediating the transducers. I have included further information on the firmware used by the microcontroller in the sections below.

## 4 Software overview

Please see Appendix B for the code relevant to the project.

### 4.1 The Arduino sketch

The Arduino MEGA microcontroller is programmed with two main functions in mind - conveying the states of the sensors to the USB connected host, and operating the haptic feedback system based on the host's instructions. Every loop, the microcontroller collects the singular signals from each current sensor, and three signals from each gyroscope. Then, it concatenates integer values that represent these signals, into a comma-separated string bounded by characters 's' in the beginning and 'e' in the end. The characters serve as a form of parity so the receiver can ensure that the string is intended to be an input. The string is then passed to the host over the serial port.The sketch is also designed to receive instructions in a similar format, but with five signals to dictate the positions of the servomotors and a remaining signal to control the unimplemented digital pin on the hand unit. The sketch parses the signals in the string to integers and applies them to the respective digital output pins.

The sketch switches between the I2C multiplexer's channels using a function provided by online by Adafruit [1]. I used an additional sketch adapted from Github user riyas-org's i2cscanner to reveal all of the I2C device addresses available on the I2C bus [2]. This was an asset when debugging I2C communication. Additionally, when, after implementing the I2C multiplexer, gyroscopes intermittently failed to initialize, stalling the sketch and causing serial communication with the host to time out. I found that inserting the core portion of the I2C scanner code as a for loop had an activating effect on the sketch. This solution remedied the stalling problems.

## 4.2 Unity3D interface script

To ease integration with VR systems, I wrote a script that enables Unity3D to communicate with the glove over the serial port, parse its signals to local public integers and concatenate output signals in the appropriate format. The script is designed to be placed under a hand model GameObject and serve as a starting point for future VR integration. In its present state, it takes but does references to but does not manipulate the Transform parameters of joint objects of an OpenVR hand model.

# 5 Manual

## 5.1 Assembly

Thread your right hand and arm through the microcontroller unit's armband, followed by the servomotor unit's sleeve, and the hand unit's glove. Adjust the armband and the sleeve for a snug, yet comfortable fit. Then, with the digits fully contracted and the haptic feedback cables in a fully extended position, adjust the position of the haptic feedback bracket until the cables are lightly tensioned. When disconnected from any power source, the servo arms can be manually adjusted to make this possible. If necessary, adjust the lengths of the cables at the link between the servo subunit's and haptic feedback harness cables. After this, reposition the gyroscopes so they are in the middle of the middle phalanges of each finger, and the proximal phalanx of the thumb. Finally, ensure that cables A and B are firmly connected and the microcontroller is connected to the host and the power source. Depending on the setup, USB and mains extension cords may be desirable. To implement the glove with a VR system, use a Velcro strap to tether a VR controller anywhere on the lower arm where it will be visible to the tracking systems. The project becomes active and starts communicating with the host once the serial port to the host is open.

## 5.2 Unity3D interface script

The Unity3D companion script offers a rudimentary graphic user interface, allowing the user to view the numeric signals the host receives from the glove and control the servomotors via sliders. Variables that handle each of these are publicly available to other scripts, and the script already takes references to joints of an OpenVR hand model, making it ripe for further development of a VR interface.

# 6 Reflection on build process, current state, and future directions

## 6.1 Reflection on build process

Much of the assembly went as expected. Soldering of the electronic components, for instance, although extensive and time-consuming, was highly successful and required minimal trouble-shooting. Mostly, each of the breakout boards and servomotors functioned as per their design. Furthermore, after the purchase of a hot glue gun, assembly of structural components became relatively quick.

Unfortunately, there were also numerous setbacks. The USB controller on first Arduino MEGA that I received for the project became corrupted, necessitating a re-flash at a later date. Diagnosing this problem and replacing the microcontroller board took up some valuable work time. At the same time I was also resolving some misconceptions surrounding I2C communication and the use of Adafruit's sensor libraries. Troubleshooting the two problems had compounding effects. Additionally, the original plans of the project relied heavily on 3D printing. However, working, available 3D printers and personnel who are qualified to use them proved unexpectedly difficult to find. By the time I had decided to seek alternative solutions, the delay had taken its toll on the project's schedule. At the final stages of the project, I also found out that I needed an I2C multiplexer to handle the six gyroscopes with identical I2C addresses; I ordered the part promptly, but because this occurred before a weekend, it took several days. When the part arrived and was implemented, inexplicable issues arose in initializing the connected gyroscopes. Lastly, throughout the development phase of the project, several situations arose in my other four classes that took up unexpectedly large amounts of time to resolve. Because of the additive effects of these setbacks, I was left with little time to maintain any milestones (besides the second for a somewhat functional Arduino sketch), troubleshoot the project, and implement a usable VR application.

## 6.2 Current state

Despite the delays, however, I was able to build a piece of functioning, interactive hardware which serves as a springboard for further development. Each of the implemented transducers have proven to work in the development environment. The glove successfully sends, receives, and parses signals over the serial port. Additionally, testing the prototype with a user who has smaller but more massive hands and arms than I do shows that the glove maintains inter-user fit. Thus, many of the project's core goals have been at least partially fulfilled.

On the flip-side, the glove's physical construction proved to be unreliable. Particularly, the links between the servo arms and the haptic feedback cables in the servomotor subunits tend to snap because they use metal solder as a structural component. I learned that this was a poor design choice soon after as I attended a soldering demonstration lecture where this technique was discussed in a negative light. Nonetheless, there was little time left to fix the mistake, and the solder joints proved a point of failure some during the final project demonstrations.

The Unity3D companion script is another aspect of the project that needs refinement. While some of its previous iterations worked smoothly without a VR setup, further refinements to implement the script with a VR system caused the serial communication with the glove to stall, unable to read the glove's output.

Lastly, because of the time constraints, and last minute setbacks on the board where it is to be connected, I did not implement a planned vibration motor. In the hardware, therefore, its infrastructure gives rise to a single digital PWM pin available for future expansion on the hand unit.

## 6.3 Future directions and conclusion

I would like to continue the development of this project in the future. The first steps will include development of a more reliable system for linking haptic feedback cables to servo arms, refining the firmware and companion Unity3D scripts, and implementing the vibration motor. Then, I would like to 3D print gyroscope attachment hardware and a lighter haptic feedback application bracket, facilitating a more natural and predictible fit for for the hand unit. Finally, I would like to develop a VR experience that takes advantage of the VR glove. After these key goals are complete, I would like to benchmark the glove against my previous prototype and any other gloves that I can get my hands in.

While I did not reach all of my desired goals in the project, given the time constrains the prototype is highly impressive. It offers three degree-of-freedom tracking, a robust anti-contractile haptic feedback mechanism, and a functioning means to measure contractile force for each digit. This feature-set alone makes the prototype noteworthy and worthy of refinement in the future.

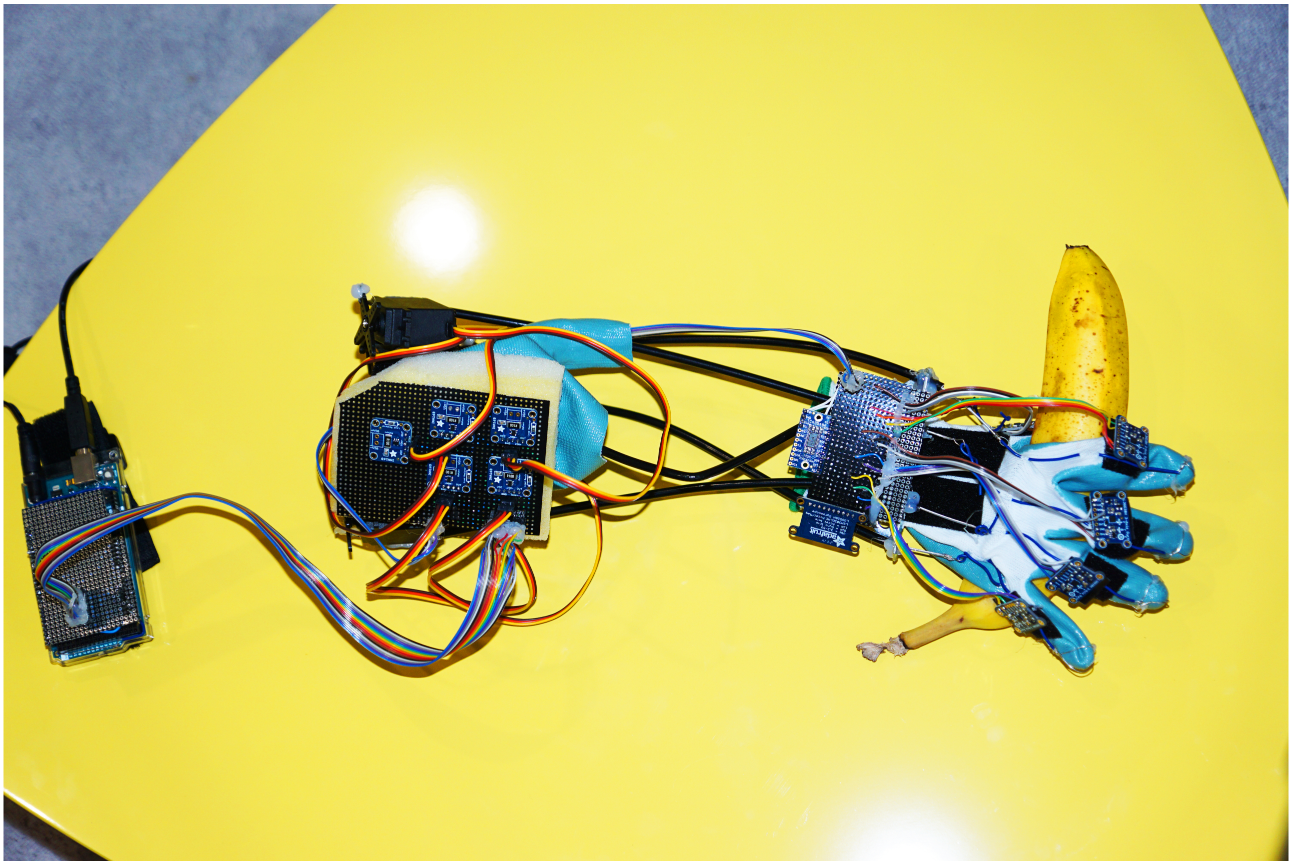
# References


(1) Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout Wiring and Text https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/wiring-and-test (accessed 12/02/2019).

(2) i2c scanner by riyas-org https://gist.github.com/riyas-org/ (accessed 12/02/2019).

(3) Kanyuka, Z. The Ultimate Controller, 2019.

# Appendix A

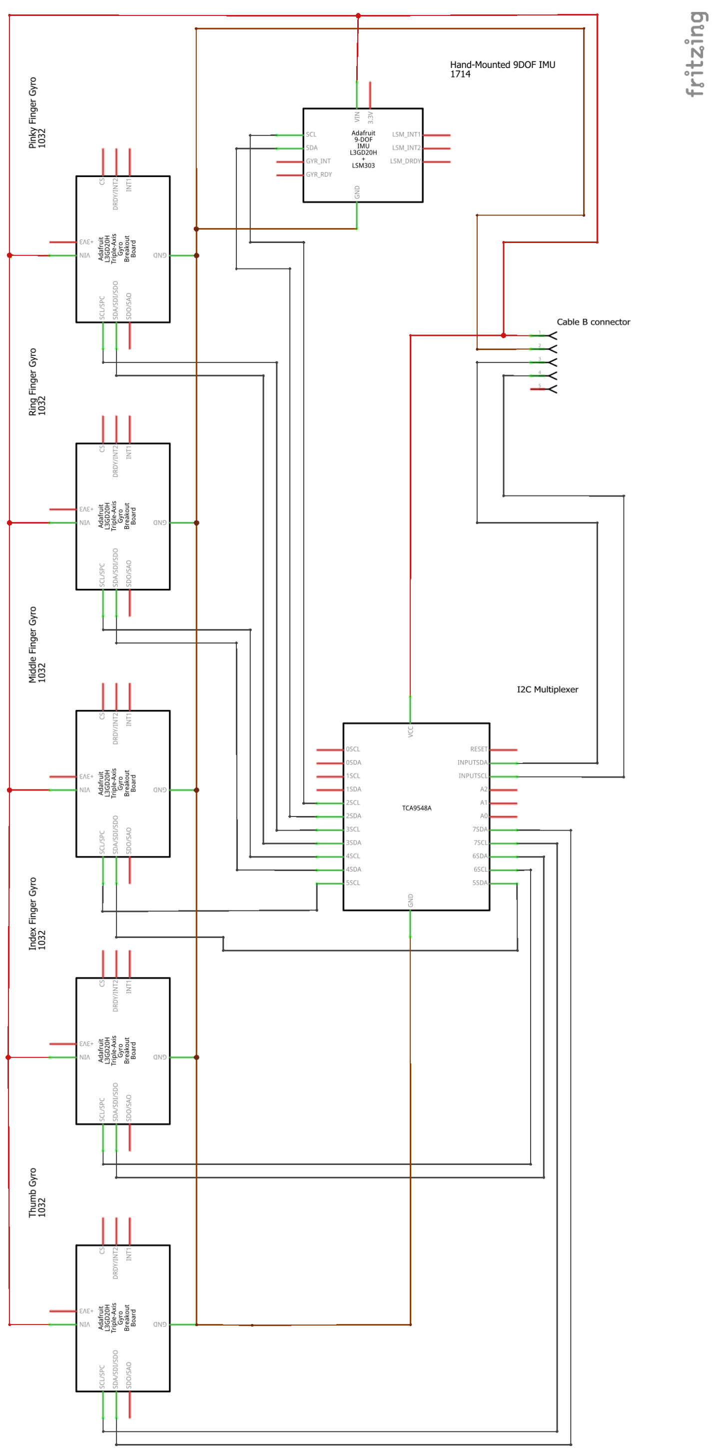


The prototype in its present state. A metric banana was placed in the photo for scale.

# Appendix B

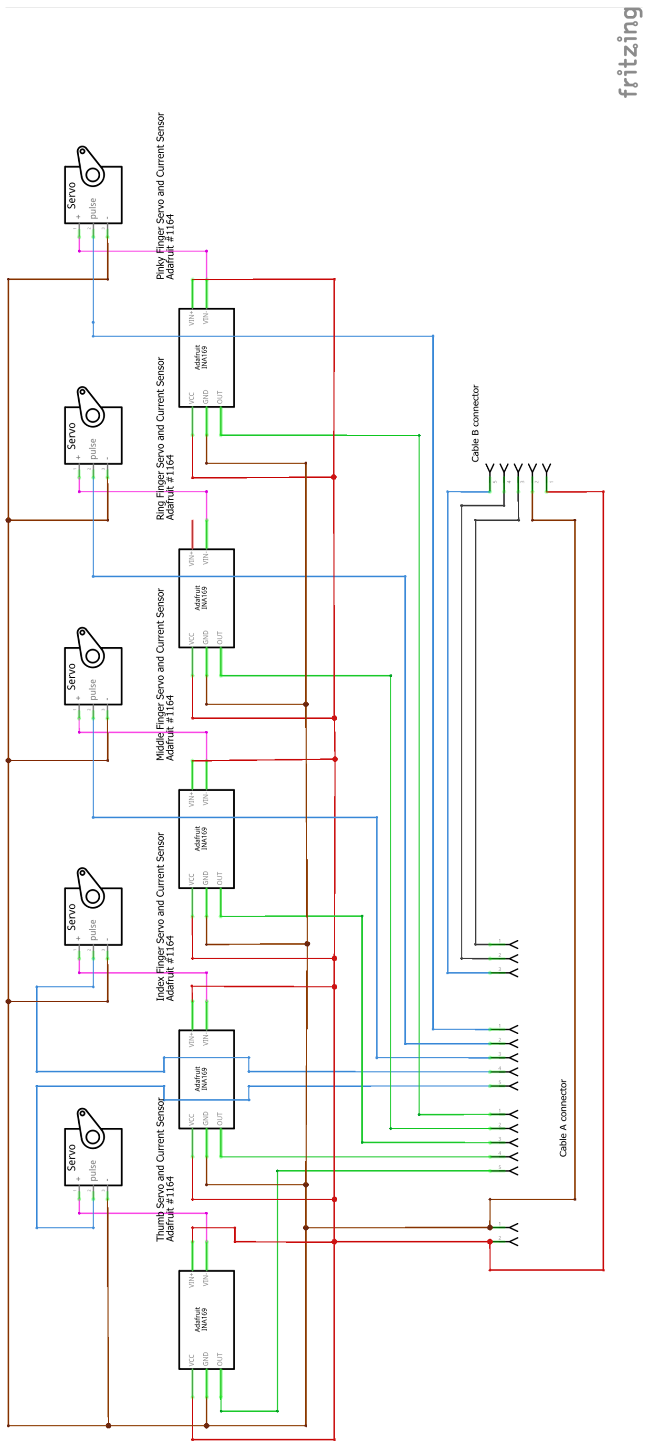
Any code relevant to the project can be found on the project's Github page at https://github.com/kanyukaz/ViRtuGlove/.

# Appendix C



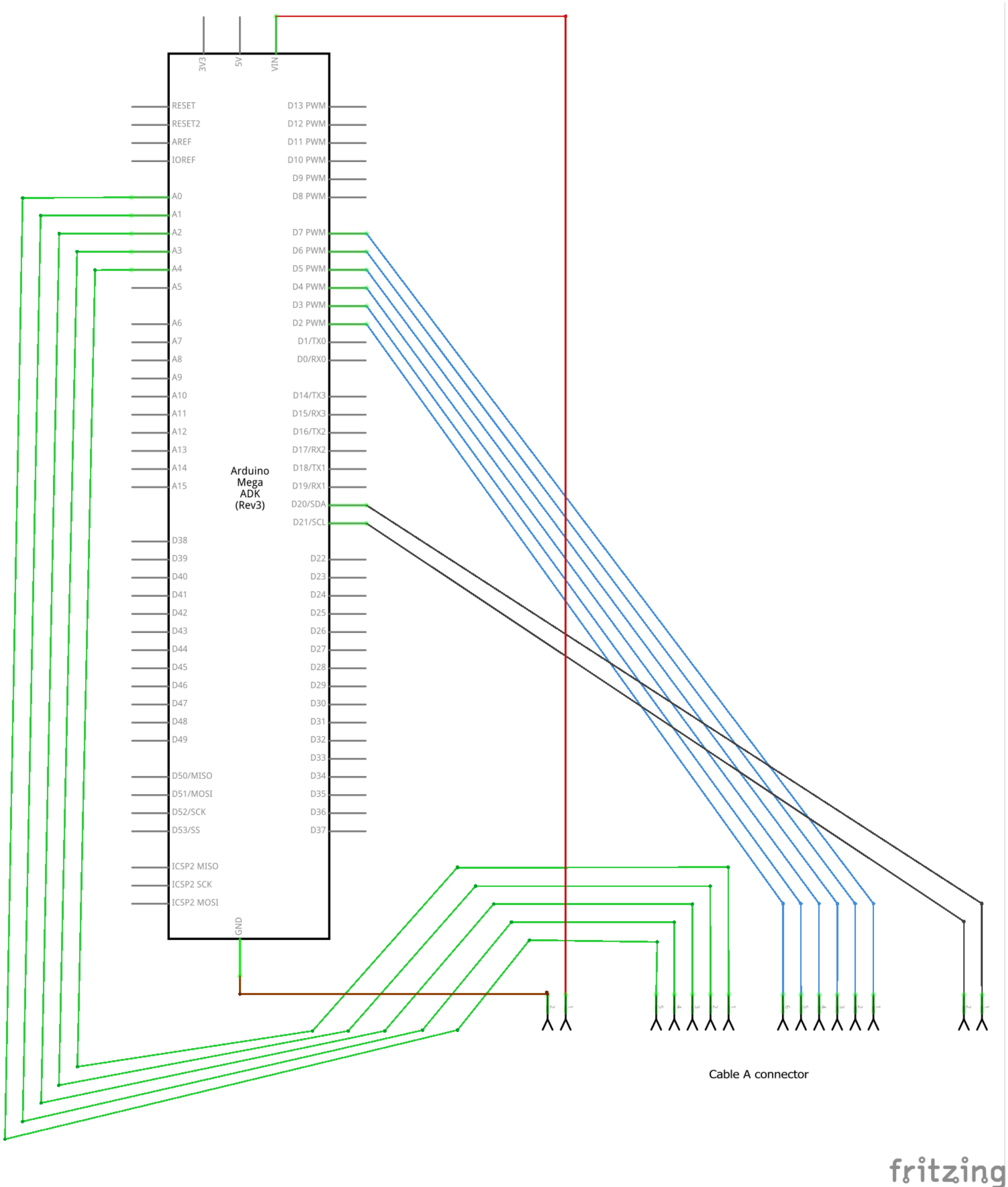
The design schematic for the gyroscope management board. The circuit was implemented as designed.

# Appendix D



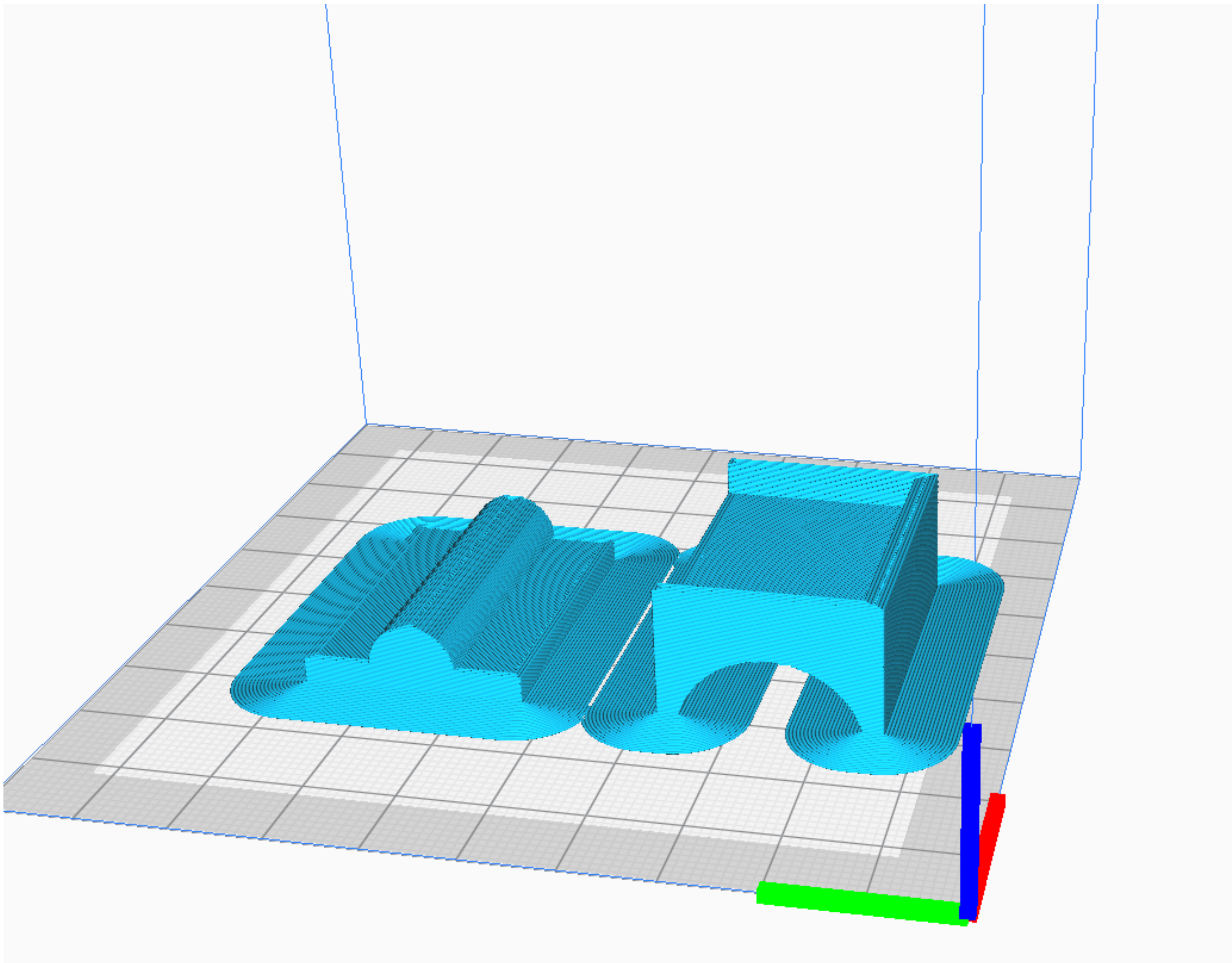

The design schematic for the servomotor control board. The circuit was implemented as designed.

# Appendix E




The design schematic for the shield that allows the Arduino MEGA microcontroller to communicate with the rest of the prototype via cable A. The circuit was implemented as designed.

# Appendix F



The model for the unused digital gyroscope mounting unit pre-sliced in Ultimaker Cura. While the part prototype was acceptable, no more than one copy could be printed before the project deadline, so the model was not used.