CS 362 Semester Project

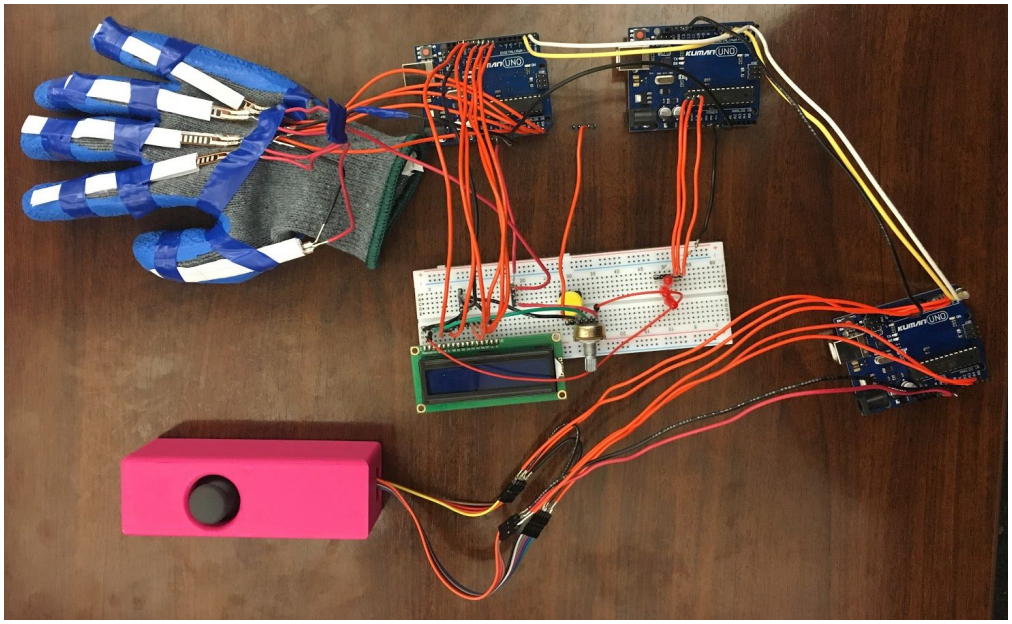
Dylan Vo | Mayank Mohan | Brandon Frale

Project Summary

Our project was designing and building a glove that can be used as a video game controller by capturing the user's hand gestures as input. Besides the glove part of the controller, which fits over the user's right hand, there is also a secondary controller with a joystick and buttons for the user's left hand, similar to the nunchuck accessory for the Nintendo Wii. This device can be used to play any PC game that is compatible with a standard game controller, such as an xBox controller.

The glove is equipped with five flex sensors, one to measure the flex of each finger on the right hand, and an LCD screen to display flex calibration and device status information. The left-hand joystick controller has a joystick and two buttons. Both the glove and joystick controllers connect to a base that has status LEDs and a calibration button.

For this device, there is one arduino to gather the inputs from the glove and another arduino to gather inputs from the joystick controller. Both of these arduinos forward the inputs gathered to a third master arduino that communicates through serial usb to a driver program on the user's PC.



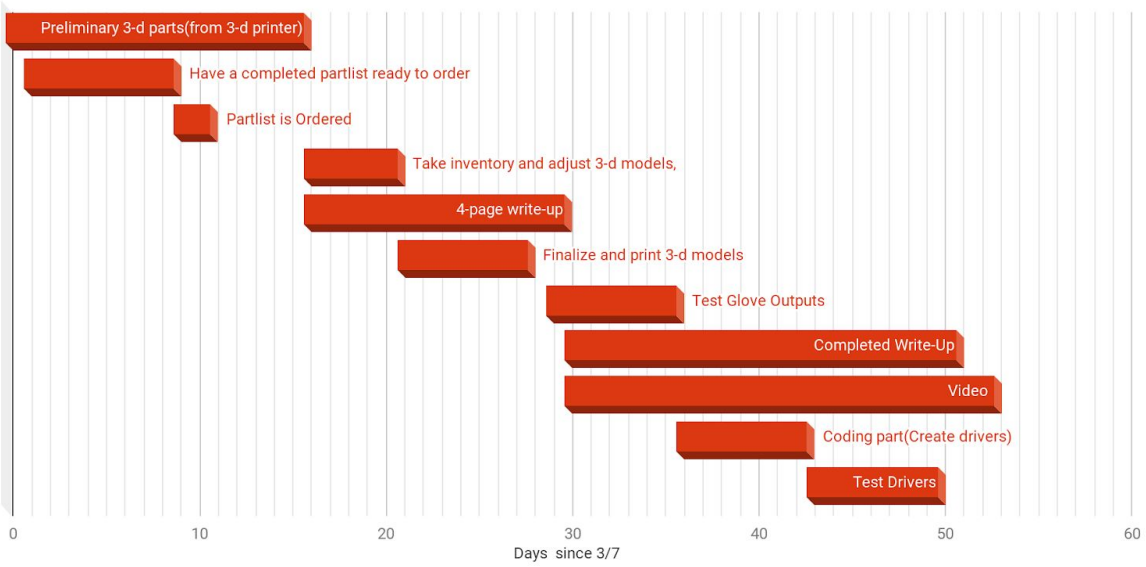
The final product.

Requirements and Planning

Project Timeline

This is our original timeline for when we expected to get each part of the project done. This was mostly based on what we felt like were reasonable estimates. Overall, the project stayed pretty consistently on schedule until April.

Start Date	***End Date***	***Goals***	***Status***
3/7	3/23	Preliminary 3-d parts(from 3-d printer)	ê
3/8	3/16	Have a completed partlist ready to order	ê
3/16	3/18	Partlist is Ordered	ê
3/23	3/28	Take inventory and adjust 3-d models,	ê
3/23	4/6	4-page write-up	ê
3/28	4/4	Finalize and print 3-d models	ê
4/5	4/12	Test Glove Outputs	ê
4/6	4/27	Completed Write-Up	ê
4/6	4/29	Video	ê
4/12	4/19	Coding part(Create drivers)	ê
4/19	4/26	Test Drivers	ê



Requirements

Since our group has three members, these are the requirements that we needed to satisfy:

1. Use three arduino devices.
2. Use at least six different kinds of external devices.
3. Utilize some sort of communication mechanism between arduinos.

For the first requirement, we decided early on that we would have two arduinos to collect input from the user and a third master arduino to combine and format the data from the first two before sending it off to the PC. To accomplish this, we decided to use serial communication between the input devices and the master device, which satisfies the third requirement as well.

For the second requirement, our original plan involved the use of slide potentiometers, an accelerometer, and a gyroscope to capture a large range of user inputs on the gloved hand, buttons and a joystick on the controller hand, and a few LEDs on the master device as status lights. As certain technical and logistical limitations became apparent, we dropped the slide potentiometers in favor of flex sensors, and the accelerometer and gyroscope were scrapped and replaced with an LCD screen that is used during calibration of the glove.

Research

The goal of this project was to have our controller work with potentially any PC game that has controller support. Building a device driver that can do that from scratch is a daunting task outside the scope of this project. After searching for libraries or other tools that would facilitate the connection between our device and these games, we found vJoy. vJoy is a device driver that acts as a virtual joystick. There are two parts to vJoy. The actual device driver, and an SDK that includes tools to allow a program to write to the virtual device. This was the perfect solution for our project, as it allowed us to simply take in data from an Arduino, convert it into the proper format, and let Vjoy to handle all of the third party software interaction.

Since we decided to use vJoy, some of our initial ideas for input devices had to be changed. vJoy allows up to 128 buttons and eight analog axes per controller. We were only planning on using three buttons, but input from a gyroscope, accelerometer, joystick, and five finger axes would use a total of twelve axes, so we had to get rid of the gyroscope and accelerometer to get down to seven axes.

Additionally, while our original plan was to use slide potentiometers and springs to measure finger movement, we found that flex sensors would be significantly easier to use.

Now that we knew how our device was going to interact with games, we needed to figure out how our Arduinos were going to interact with each other. Normally, an Arduino board can only

have one serial connection active at a time on digital pins 0 and 1. Luckily, one of the built in libraries in the Arduino IDE is the Software Serial library. The Software Serial library allows the user to set up multiple serial connections on different pins, as long as only one is being used at a time. For our purposes, this was more than sufficient.

Design and Fabrication

Individual Arduino Functions

A requirement from having three people in our group was to use three Arduino devices in our final project. Due to the number of inputs that our project uses, we make full use of each device, and each one has its own clearly defined role.

Master

The master Arduino has a simple loop that does three things. First, it starts listening on the controller's serial connection and sends the controller an interrupt signal to tell it to send its current data. Once it gets the data from the controller it puts it into a data structure and does the same thing with the glove arduino. Once both devices have been polled for their data, the combined data packet is then forwarded to the VJoy program on the user's PC through the USB connection. The master also has three status LEDs that let the user know of connection problems between any of the devices.

Controller

The controller constantly loops and updates its button states and joystick values, waiting for an interrupt signal from the master. When the interrupt signal is received, the current input values are immediately sent over the serial connection to the master.

Glove

The glove loop is basically the same as the controller loop. The flex sensor values are constantly being updated while the device waits for an interrupt signal to send them. Additionally, besides the master, there is another interrupt attached to a button. This button causes the glove to enter calibration mode, and an attached LCD screen prompts the user to fully flex their fingers so that new minimum and maximum values can be calculated for each sensor. When not calibrating, the LCD shows the current flex value of each finger, with a refresh rate of about 4 times per second.

Controller Driver Program

In order to bridge communication between the Arduinos and vJoy, a simple program was written in C++. The program prompts the user for a port to connect to through a serial connection. It then establishes itself as the sole writer to a particular vJoy device as well as establishing a

serial connection on the port specified. It then continually reads data from the serial port, formats it, and passes it on to the vJoy virtual device.

Code repo (Arduino code included):
https://github.com/MayankMohan/ArduinoController

Wiring and Components

Given the described functionalities of each device above, we had to decide how we wanted to everything to connect. The tables below describe which Arduino connects to which part and what pins those connections are made on. See the appendix for the full wiring schematics.

Master

External Device	Connection	Notes
Left Hand Controller	Serial in (pin 2) / Digital out (pin 4)	Using Software Serial Library to do multiple serial connections.
Right Hand Glove	Serial in (pin 3) / Digital out (pin 5)	Using Software Serial Library to do multiple serial connections.
PC	Serial in / out (USB / pins 0, 1)	Hardware Serial, using USB and makes pins 0 and 1 unusable
3 x LEDs	Digital out (pins 11, 12, 13)	

Controller

External Device	Connection	Notes
Master Device	Digital interrupt (Pin 2), Serial out (pin 1)	Hardware Serial
Joystick	Analog in (pins A0, A1) / Digital in (pin 6)	Uses 3 power and 3 ground wires.
2 x Buttons	Digital in (pins 4, 5)	

Glove

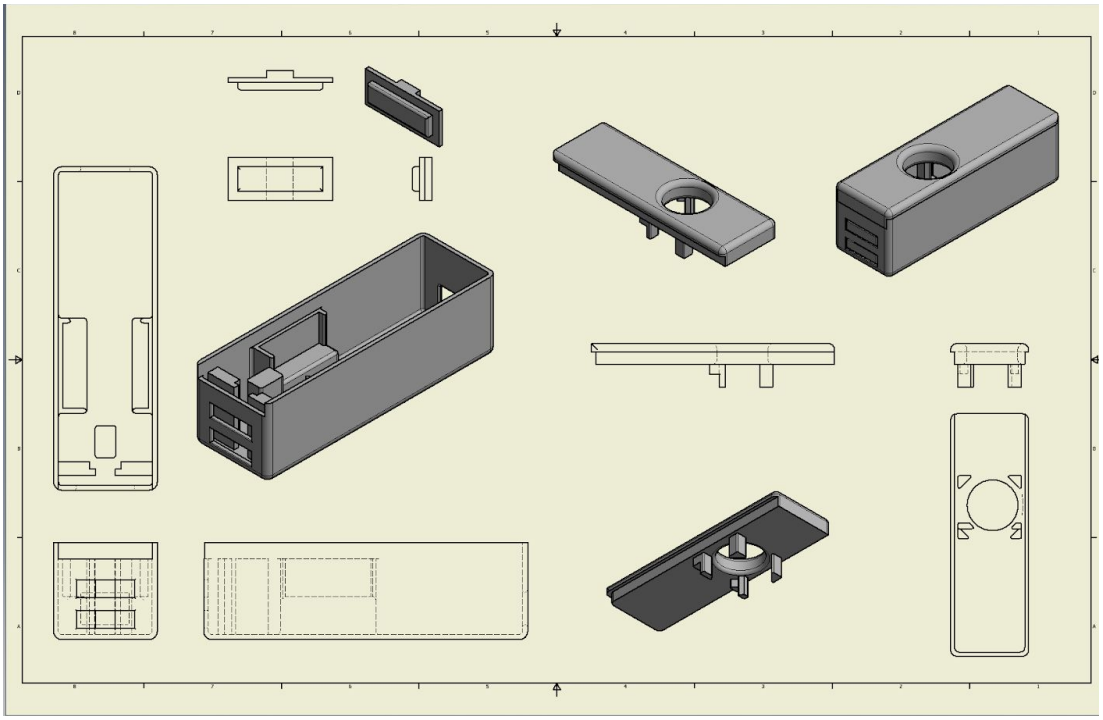
External Device	Connection	Notes
Master Device	Digital interrupt (pin 2), Serial out (pin1)	Hardware Serial
5 x Flex Sensor	Analog in (pins A0 - A4)	
LCD Screen	Digital in / out (pins 8, 9, 10, 11, 12, 13)	
Button	Digital in (pin 4)	Calibration button

Physically, most of the wiring was done by soldering wires to pin headers. Our final product has over a dozen pin header strips and about three dozen individual wires. Although this creates a

big web of unwieldy, unconfigurable wiring, it was worth it to be able to plug each strip into a row of pins without having to spend too much time figuring out what goes where.

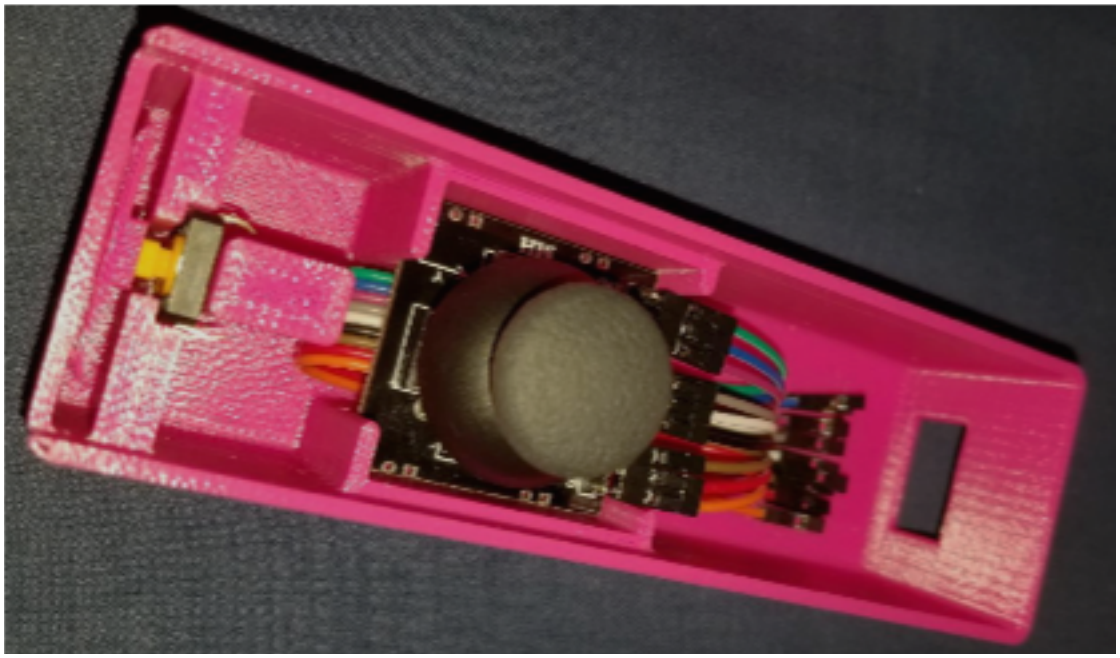
Inventor and Controller Printing

For our controller design, we decided to do a design similar to a Nintendo Wii nunchuck. We chose this design because it allowed us to use the joystick and the two buttons all with one hand, since the other hand would be wearing the glove controlling the flex sensors. We initially drew a sketch of the design we wanted on paper. After measuring our components, we used Inventor to make a 3D model of the casing that houses the joystick, buttons, and their wiring. We decided to choose Inventor over SolidWorks because it was readily available to us as students to download and install on our personal computers.



Inventor schematic of controller casing.

The joystick fits snugly in a little square space near the front of the controller. The two buttons sit on top of each other in front of the joystick, facing forward. There is a bar separating the joystick and the buttons so that the bases of the buttons are held in place when being pressed. The joystick rests elevated in the controller so that there is room underneath it to run wires to the buttons. Wires run from the components in the front of the controller out a slot in the back. The caps for the buttons were also 3D printed and are held securely in place by the button bases. To access these components the top of the controller slides off fairy easily, yet not so easily that it would happen by accident.



View of controller components in casing.

Glove Construction

In order to get the flex sensors to stay on the user's fingers, we decided to fasten them to the outside of a glove. The glove itself had to be made from an antistatic material, since any sort of static charge might interfere with our electronics. We ended up with an antistatic glove from Home Depot.

To fasten the flex sensors to the glove, we first wrapped each sensor in paper, that anything touching to back of the glove wouldn't interfere with the sensors. Now that the sensors were wrapped in paper, we simply taped one to the back side of each finger. In lieu of any real sewing skills, this simple solution was surprisingly effective for our purposes.

Post Mortem

When doing this project, we started with a rough timeline of what we wanted done by when in sort of a ground up order of building things. Since this assignment was open ended, what we originally had planned ended up being not completely, but at least considerably different than our final product. Given our goals and resources, this project was a success, even though certain parts don't work exactly how they should. Of course, knowing what we know now that we didn't know in the beginning, there are things that given another chance, we would definitely do differently.

Process

One thing that would have saved a lot of headache is if we would have built the project more piece by piece, instead of layer by layer. We had the separation of responsibilities for each Arduino established early on, but instead of building each Arduino one at a time, we built them slowly together. Toward the end what we had were three connected Arduino devices, but none of them was known to work by themselves, let alone together.

Design

One aspect of our design that we would change almost completely is how our device is worn. Due to problems with our wiring, we had to have the glove and controller right next to each other instead of being a more natural distance apart. Currently, the way we have our device setup is that the Arduinos and a breadboard just hang from the glove and controller. Ideally, there would some sort of base station or the whole thing would be wearable, but compared to getting the thing to work, this was not a priority.

Printed Controller

For most of the group, this was our first time working with CAD software, and using Inventor was not as simple as we first anticipated. The learning curve was frustrating at times, especially because there were simple cuts we wanted to make that ended up taking a long time to execute, despite knowing exactly what shape we wanted. Once we got past our first model, adjusting and remaking it afterward was a lot easier.

The first model that we had designed did not turn out how the way we wanted it to. We wanted the case to have a sliding mechanism on the lid so that wiring could be sealed and hidden inside, but the printer uses a temporary scaffolding material to elevate certain parts and in our case we couldn't remove enough of the temporary material from our print to get the sliding parts to move the way we wanted them to.

Also, while our dimensions were correct on paper, we neglected to take tolerances into account, causing the pieces of our first printed controller to be too tight. The buttons were too tight to press easily, and it was difficult to fit things into place without feeling like we were about to break the whole thing. A second controller was printed with these changes in mind. This one was a lot better, but the lid didn't have a lip where it should have and the range of motion for our joystick wasn't enough. Instead of reprinting the whole thing again, which is a six hour process, we simply made a new lid with better dimensions.

Construction Defects

There are certainly things that we could change about our design, but given the chance to simply rebuild our project, there are a number of things we could have done better as well.

Wires

The wires we used to connect everything together were thick and not very flexible, making our setup brittle and awkward. It turns out that there was some really great wire available to us in the makerspace, but by the time we found it, we had already spent hours soldering together our current setup.

During the final testing of our project, we were receiving inconsistent and confusing data when communicating across devices. Each device, when plugged into a testing configuration with dummy data being sent back and forth, everything was working fine. It turns out that in order to hold the controller in one hand and wear the glove on the other, we had used a wire that was so long that a proper signal couldn't travel from end to end without getting distorted.

Finally, although we color coordinated all of our wiring, some clearer labelling would have been really helpful.

Glove

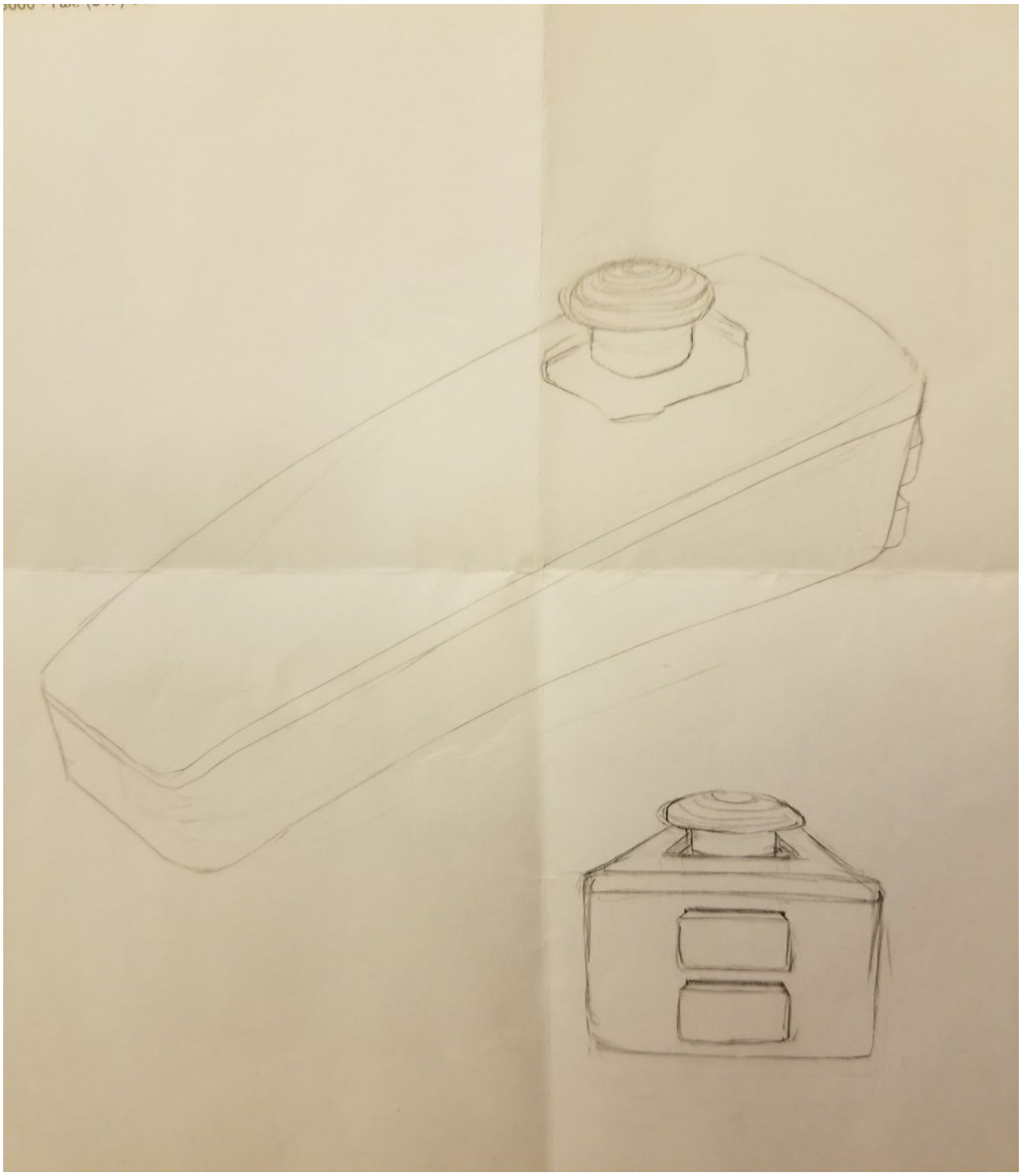
Unfortunately, the glove we bought to attach the flex sensors to was too small for everyone in the group to fit their hands in. It would have been better if we had gotten a larger size. Also, when attaching the flex sensors to the glove, we may have wrapped the tape around a little too tightly. Combined with replacing our extra long wire that was mentioned before, having a small, tight glove makes our device awkward to use.

Besides flex sensors, the glove has an LCD screen for calibration. For some reason, we could not get the LCD screen to work, and we didn't have time to disassemble the giant web of already soldered wires to fix it.

Controller

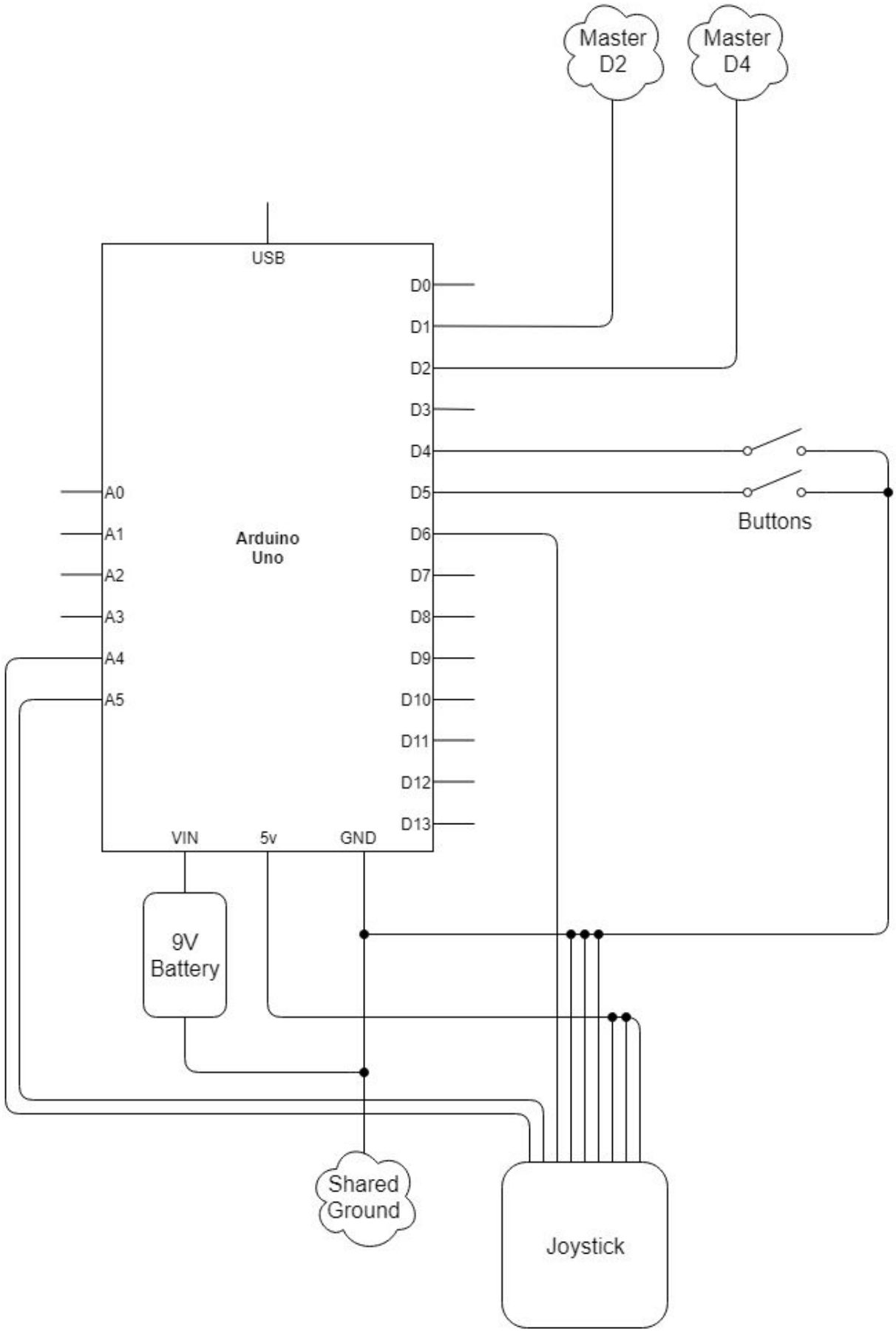
When assembling the controller, we had to super glue our buttons together so that they wouldn't move inside of the case. A wire was not in contact with a button when we did this, and the glue created a layer of insulation on the lead, requiring replacement of the button.

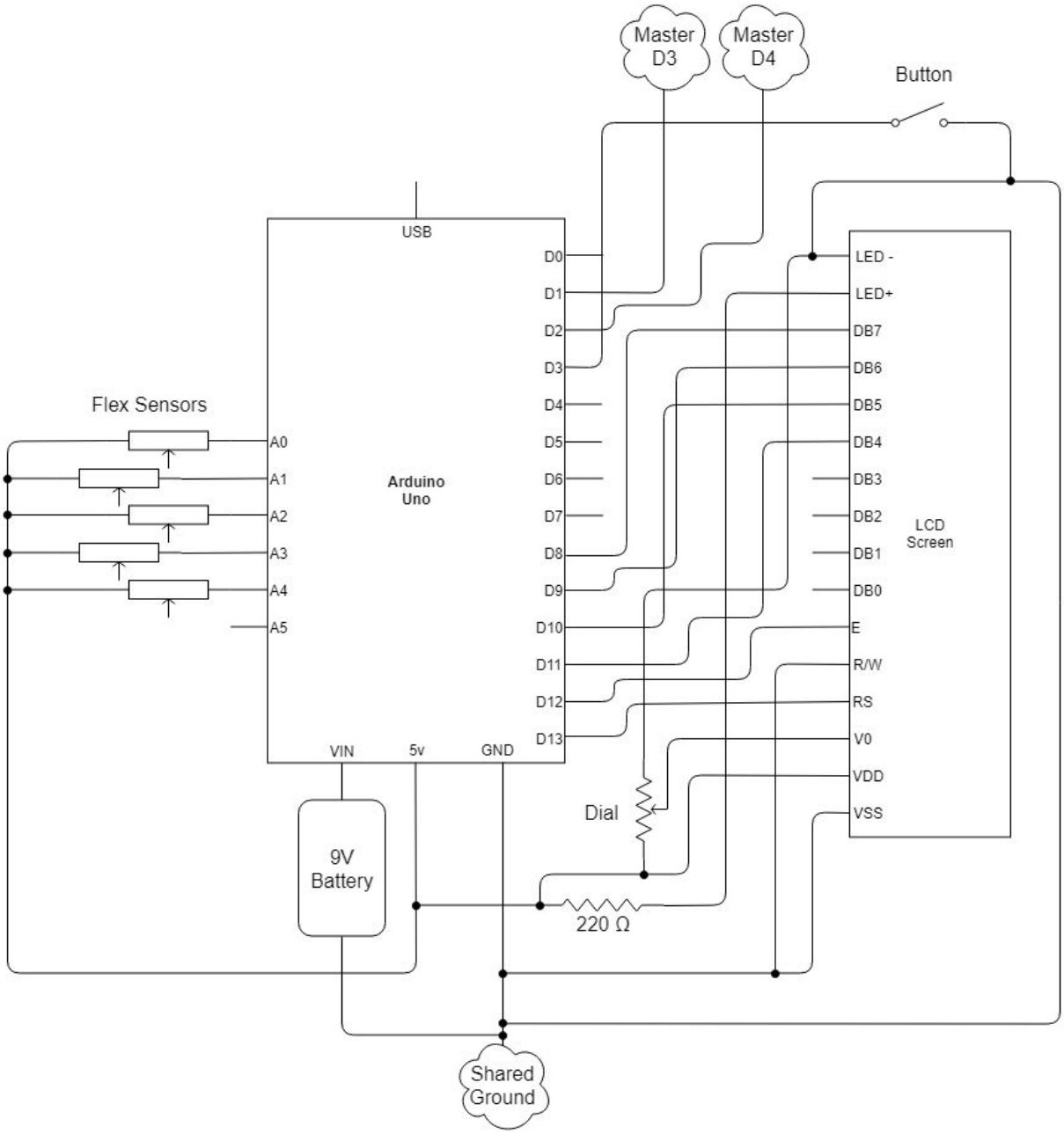
Appendix

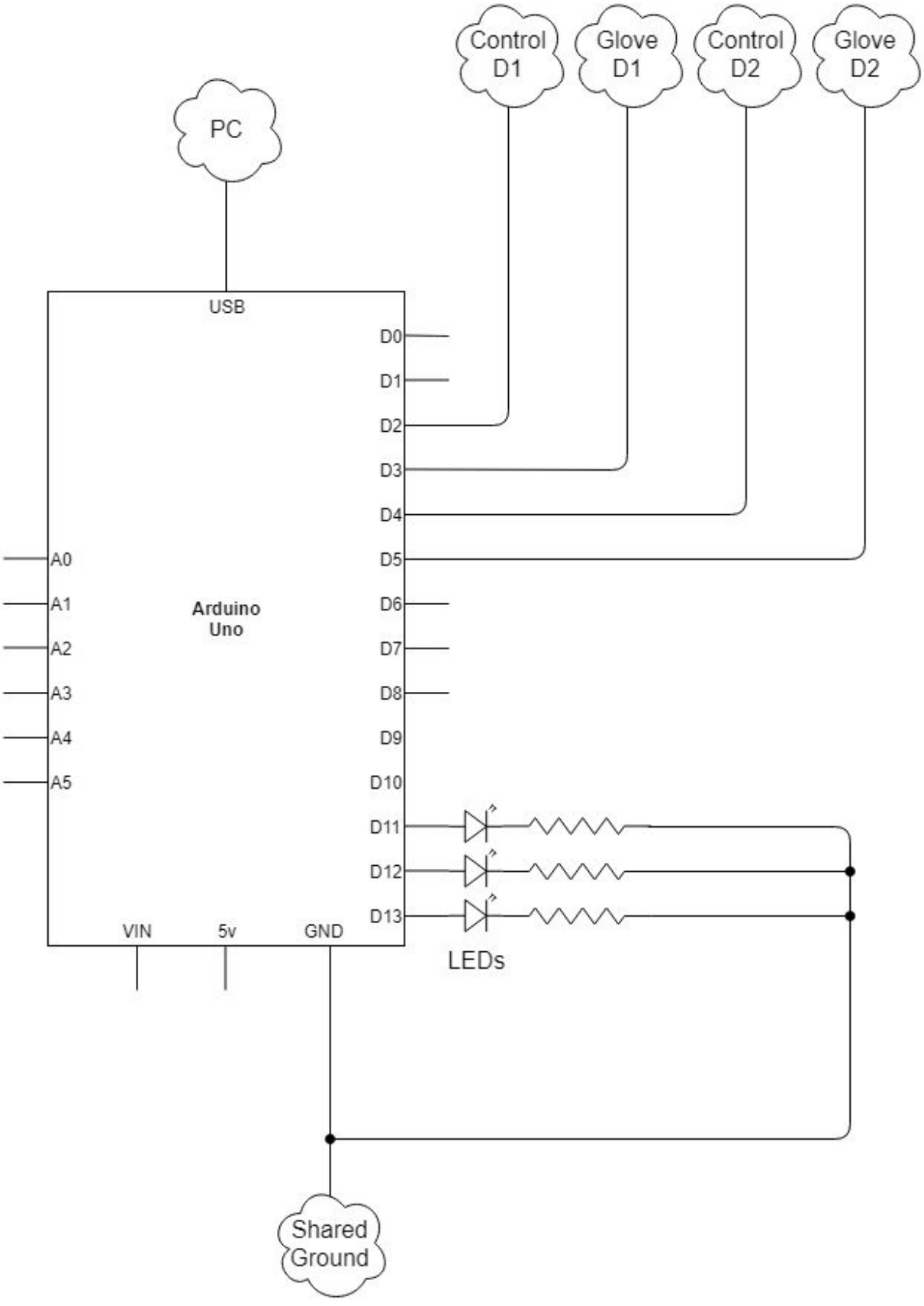


Original controller drawing.

Some parts that we ordered online.


Master
D2
Master
D4
USB
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
A0
A1
A2
A3
A4
A5
Arduino
Uno
Buttons
VIN
5v
GND
9V
Battery
Shared
Ground
Joystick


Master
D3
Master
D4
Button
USB
Flex Sensors
A0
A1
A2
A3
A4
A5
Arduino
Uno
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
VIN
5v
GND
LED -
LED+
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
E
R/W
RS
V0
VDD
VSS
LCD
Screen
Dial
9V
Battery
220 Ω
Shared
Ground


Control D1
Glove D1
Control D2
Glove D2
PC
USB
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
A0
A1
A2
A3
A4
A5
Arduino Uno
VIN
5v
GND
LEDs
Shared Ground