Computer engineering

TEJ4M1 1A

Game Controllers

By Sean Ye and Nicholas Kaldis

Description:

For this project we designed and built a game controller for video games using arduino.

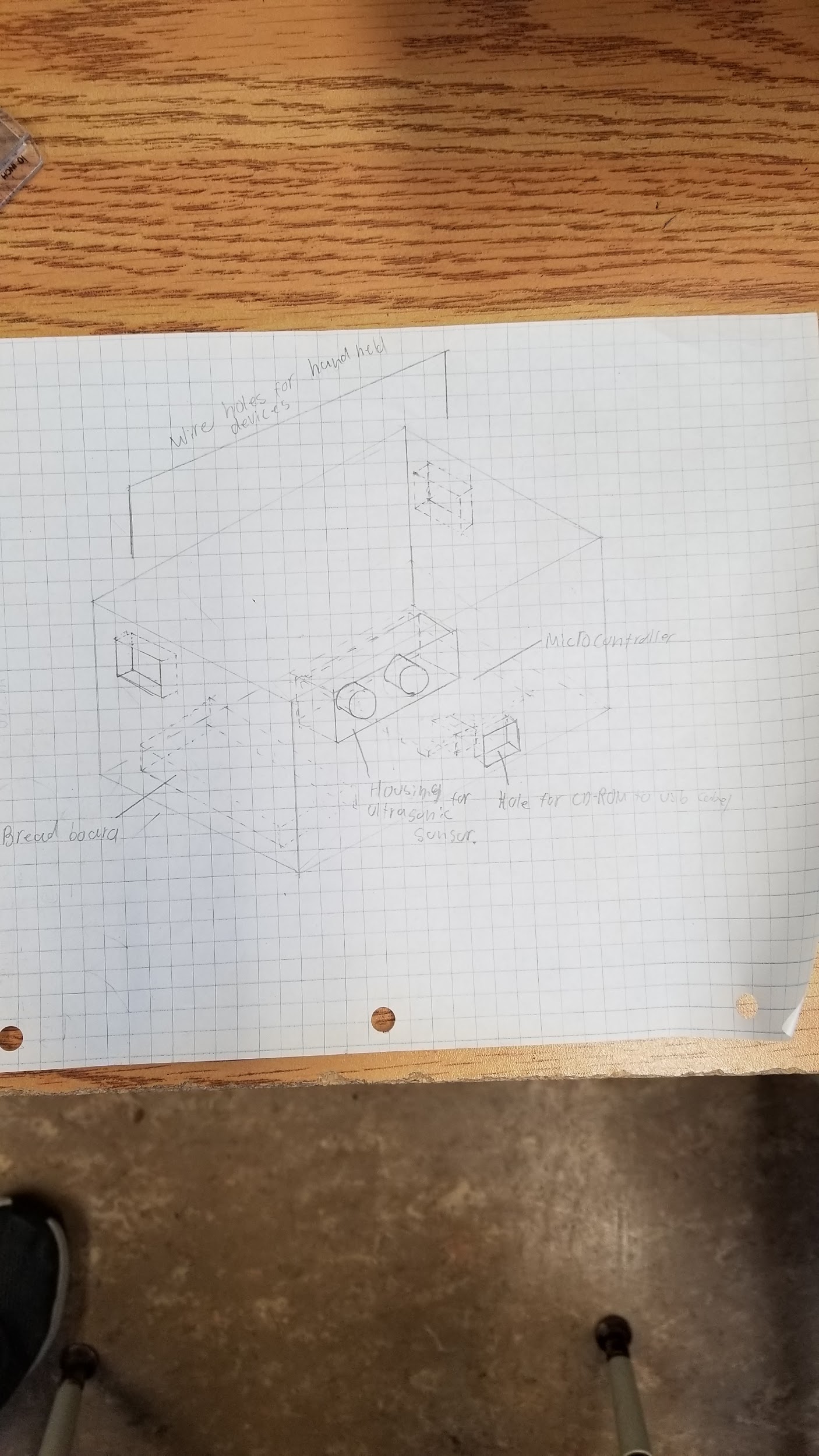
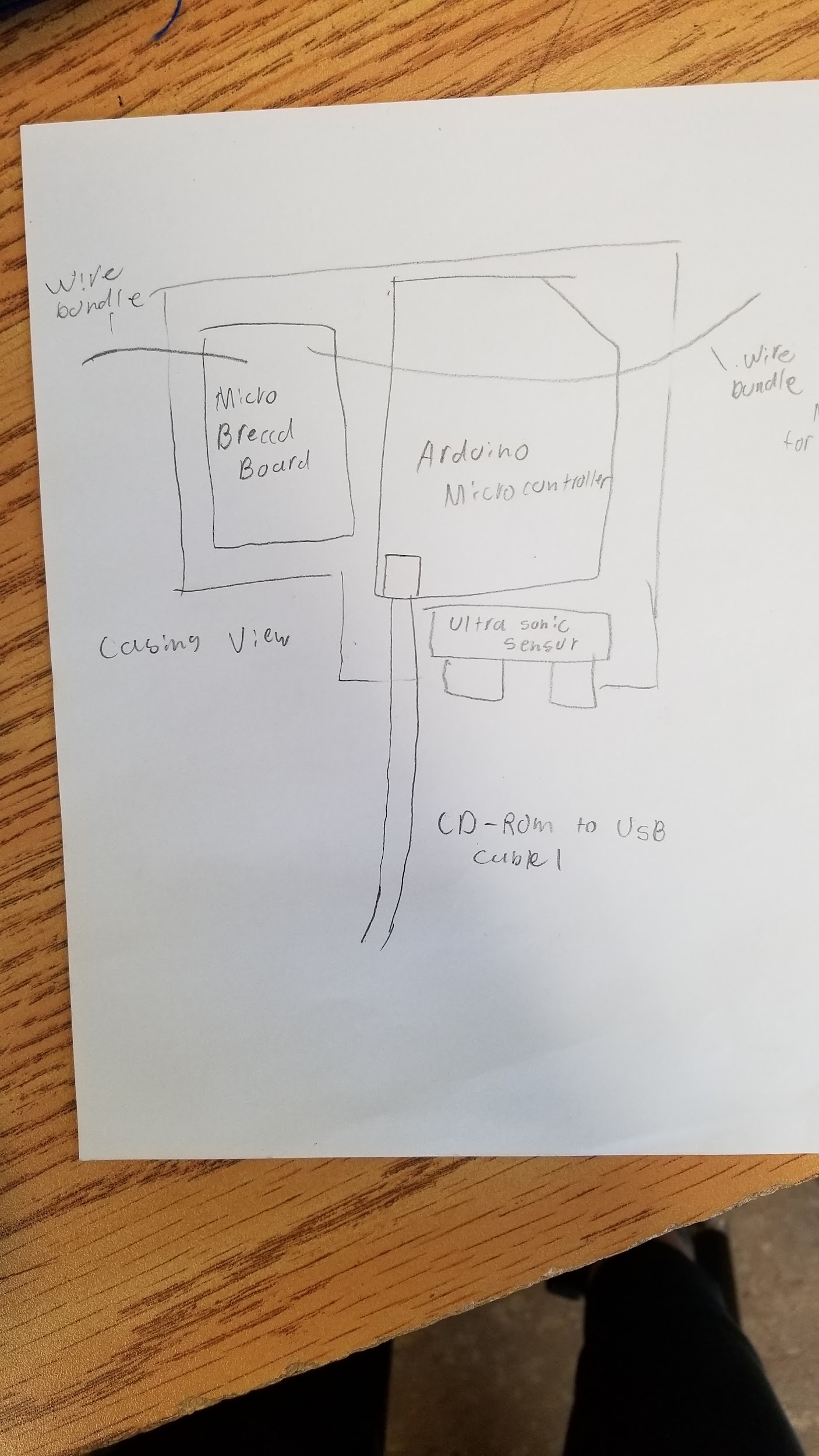
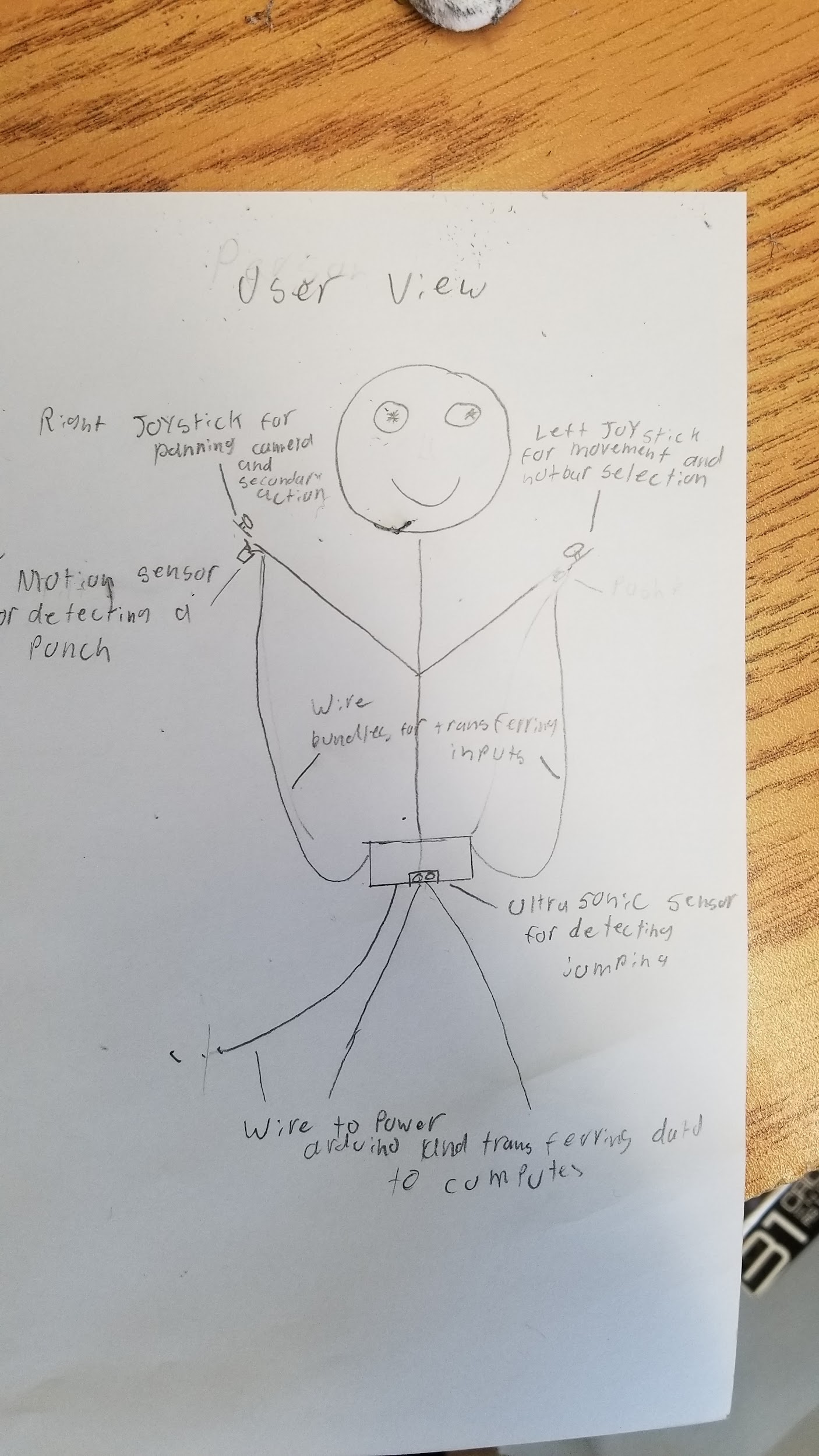
Build Process:

The most important part of the project was to control the actions on a computer. This made preparing the process in which the Arduino’s data could be read the first stage of the build. The process must translate controller data into various inputs such as mouse movements and key presses in the game. To solve this, we needed a way to read data from the microcontroller and used a software called [CoolTerm](https://freeware.the-meiers.org/) which writes serial data sent from the microcontroller into a file. We then set up the joystick with the board and successfully wrote it to the file. While testing the player movements in Minecraft we found that some controls were quite delayed and would make it impossible to enjoy playing the game. Originally we thought it wasn’t writing to the file correctly, but later we found that some pieces of data were only written every dozen calls, which meant large delays due to fewer updates. The reason was that Serial.print() takes longer than Serial.write() even when they were both sending numbers as the slower option was sending ASCII encoded strings while the second one could quickly send bytes, fixing the issue.

The next feature would be to implement the primary action which was important to punch and break blocks. The original plan was to detect when the user shook the right joystick using a motion sensor, but it was found to be useless and the tilt sensor was a viable replacement. The player inventory trigger and Hotbar selection were also implemented by adding unique control combinations using the current devices. The casing was created as most of the hardware was already added and was made from [Legos](https://www.lego.com/en-ca) for quick and durable modifications and prototyping. The case also provided protection to the hardware and had clamps which keep the external wires secure to the casing and breadboard. Jumping was added as the ultrasonic sensor was tested and mounted into the case. This posed a problem as the player would hit whenever the user jumped since the tilt sensor could not tell the difference between a jump and deliberate shaking. The solution we implemented was to have the user click the right joystick button first to trigger either a secondary or primary action and have the user shake to determine if the action is primary. This solved the problem as jumping no longer had an effect on the conditions of the action.

The final step was to create an application that would initialize the environment for the controller to run. The mouse and keyboard controller was currently written in C++ and the file interpreter helped the controller read the right line written in Python. This led to the application starting the file interpreter and the controller at the same time. The application also opened CoolTerm with the correct settings and checked if anything was written into the input file from CoolTerm before starting, to avoid getting the mouse stuck in permanent movement from the previous data.

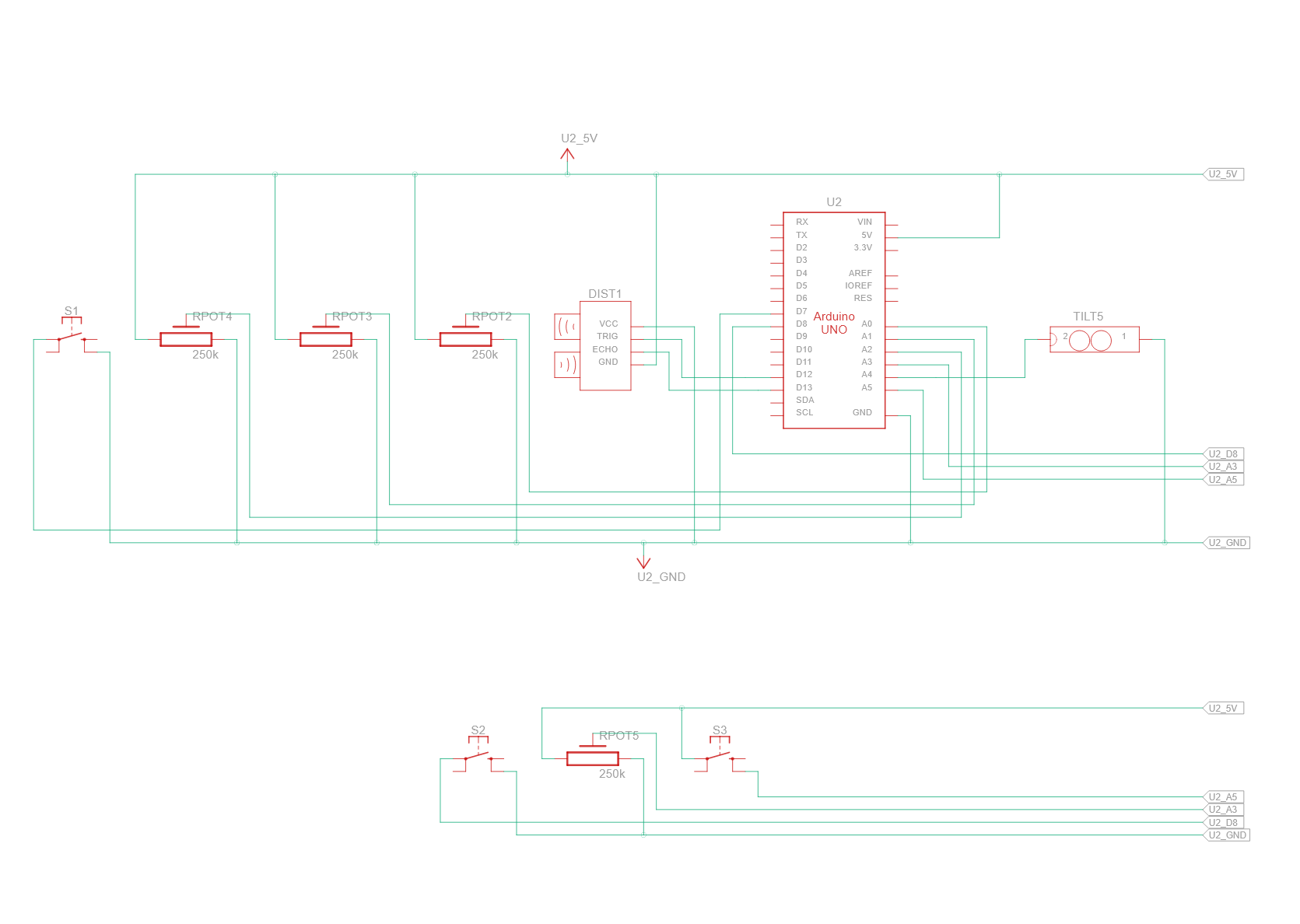
Drawings



Parts list

| Arduino |  |
| --- | --- |
| Push button |  |
| Joysticks |  |
| Wires |  |
| Long wires |  |
| Long USB PC Cable |  |
| Ultrasonic sensor |  |
| Mini Breadboard |  |
| Standard Breadboard |  |
| Lego |  |

Wiring diagram



Summary:

In conclusion after a lot of trial and error our project was able to accomplish all the things that we needed it to. We can control player movement such as player orientation, jumping, and walking. We could also open player inventory, cycle through the players hot bar and even crafting. Our design for our controllers function the way that it’s supposed to. Most of our wires and circuitry is kept in a protective box made of lego to allow the user to have free control and not feel restricted and this also limits the chance of someone getting hurt or anything breaking. Sean and I both worked on the controller and the report, Sean mostly worked on the coding for the controller.