VR Disability Accessibility Design Document

Team 27: Evan Miller (emm2), Justin Zhou (jezhou2), Vinith Raj (vinithr2)
ECE 445 Design Document: Fall 2020
TA: Dean Biskup

Introduction

Problem and solution overview

As the world keeps progressing, virtual reality (VR) devices are becoming more and more sophisticated. There are full body capture devices being released on the market, as well as haptic feedback. As VR becomes more high powered and consumer friendly, there is a good potential for developers to make more gaming, resource, and teaching applications. As it stands right now, however, VR is inhospitable for people with disabilities. This is especially true of people with disabilities of the arms or hands.

The issue of accessibility for people with disabilities is something that can be solved through multiple avenues. The way that we are proposing specifically tackles this issue for people with the disabilities of the arms or hands. A controller will be made for feet that are controlled by using pressure sensors to allow for both continuous and quick inputs. A USB driver will be necessary for parsing and identification of signals, and saving and normalizing a person's weight and pressure can be done via software to allow for calibration. The sensors will have their own battery pack(s) to provide power and reduce cord tangle, and potentially a single Bluetooth device will be used to wirelessly send the data, at the cost of power. To gather and send the signals, an ATMega328 microcontroller and a USB-to-Serial adapter will be used, and if Bluetooth is used, then a Bluetooth-to-Serial adapter will be used, however the project will continue under the assumption of the former. There are multiple approaches to adjustability and accessibility for users that are explained further in the subsystems section of the proposal. Testing is necessary to know what is most accessible, but either a station to keep the feet in the same place that acts as a hub for the electronics and power, a strapped on sensors/battery packs combination, or a fusion of both with a set of sensors on a pad (like the pad for the game *Dance Dance Revolution*) that has physical markers to allow a user to guide them while using the controller will be developed for user input. For this project, we will be testing and making designs for the first and third input options.

There are a number of disability related controllers on the market like the QuadStick FPS game controller, and the Microsoft Adaptive Controller, however they are expensive and not developed to be used in a VR environment [1][2]. There is, however, only one foot controller for VR, the 3D Rudder Foot Motion Controller. There are a number of complaints about it, namely that it is hard to control and slides around on the ground, which we intend to address with this design. It should also be noted that there is praise, largely from people with upper body disabilities, but this support can be attributed to the fact that it is the only device that addresses their issues on the market [3].

Developing for VR also comes with it’s own issues, which helps explain the lack of suitable controllers. One issue is that VR controllers have a different SDK than other controllers that exist for well established systems (PS4, XBox, etc.), so when a controller is made, it has to be compatible with the headset. On the hardware side, the controller has to be designed so that someone can use it without being able to see it, and should be stable and comfortable for prolonged use. If someone cannot navigate the controller without sight then it is unusable, due to them having the headset strapped to their face. There are also some general issues with developing for VR, such as keeping things feeling natural for users to increase immersion, and keeping peripherals strong in the case of excess movement on the part of the user due to getting too engrossed, which includes paying attention to how wires and other parts of the peripheral can get caught or hit other things.

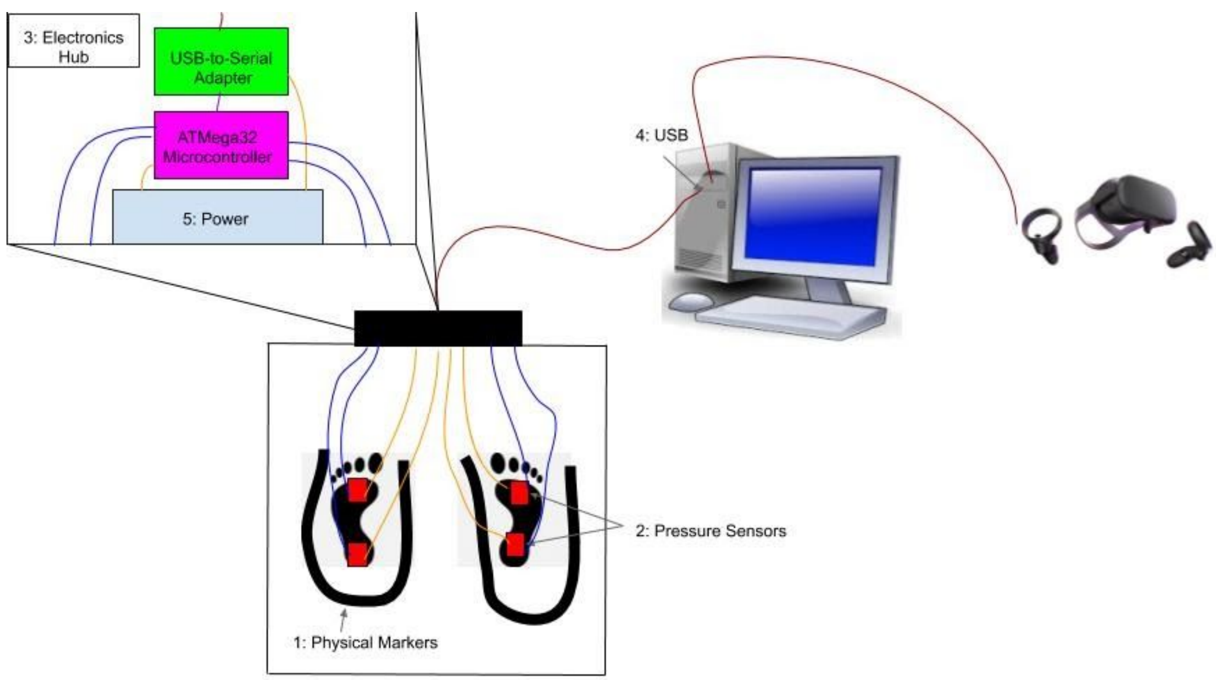

Figure 1: Representation of a sensor pad based approach to controller design with location of subsystems. Each wire has its own type defined by color: Blue - Pressure sensor output, Orange - Power Supply, Purple - Serial data from microcontroller, Maroon - USB data. The physical markers are symbolic placeholders for the real foot guidance design.

High Level Requirements

The high level requirements for this project are as follows:

- **Microcontroller Development**
  The code developed for the ATMega controller is fast, bug free, and accommodates asynchronous inputs from the amplifiers.
- **Signal Identification**
  The driver this project develops identifies the signals for each user input and makes them available in a usable format for other PC programs.
- **Navigation**
  The physical markers guidance components can be navigated by first time users while wearing a VR headset such that while on the controller they are able to reposition their feet to the correct corresponding sensing location 95% of the time if they move out of position.

Design

Block Diagram

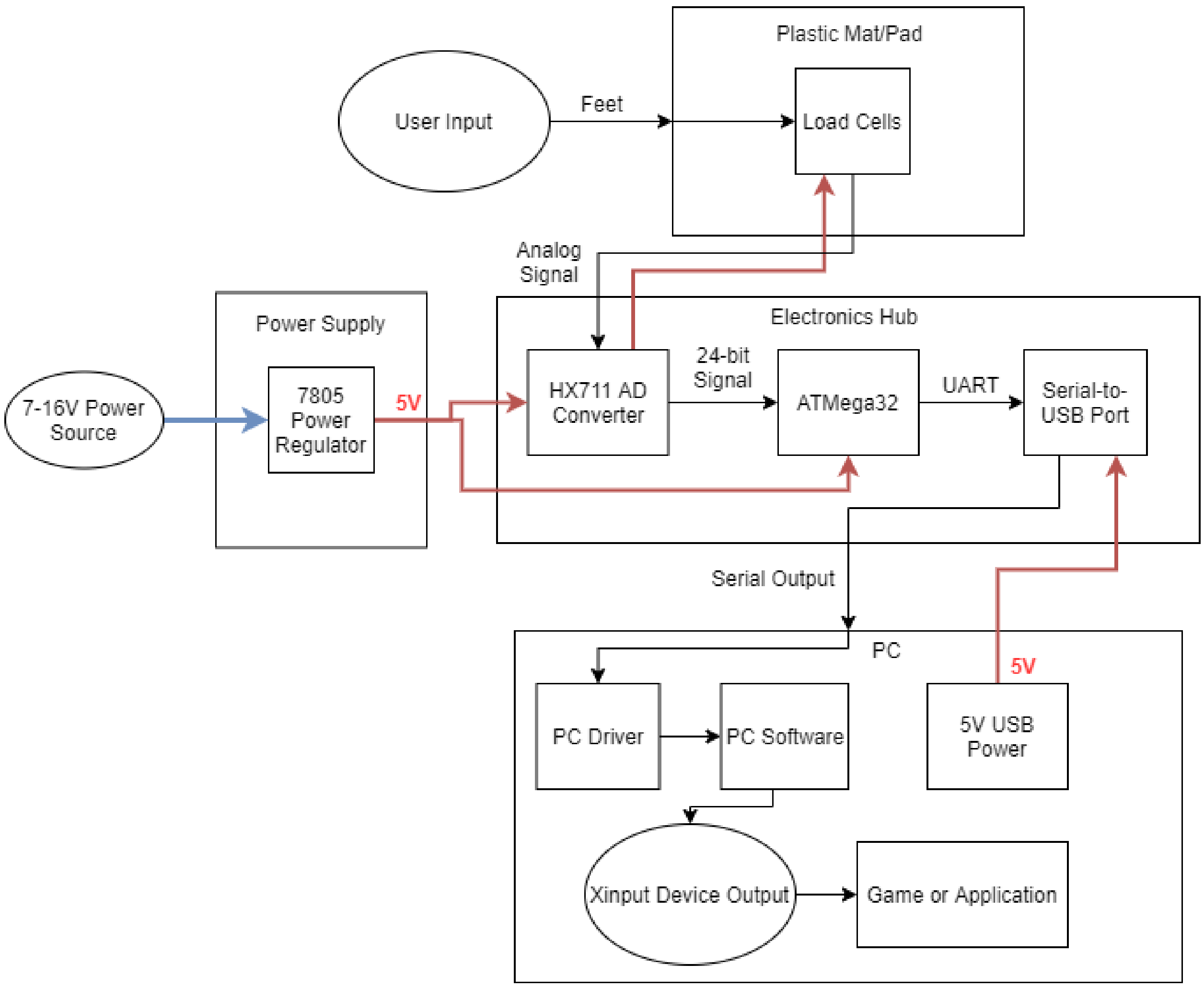

Figure 2: High-Level Block Diagram

This block diagram is a high-level overview of the device. It takes input from the user stepping on a pressure sensor, which sends an analog signal to the HX711 AD Converter which converts the signal into a usable 24-bit digital signal that the ATMega328 Microcontroller can process. The microcontroller will then process the signals into UART and send them to the USB to Serial converter which converts the signals to the USB protocol. The PC driver and software then processes this USB signal into an XInput signal where it can then be used in any PC game or application. This block diagram satisfies all the high-level requirements because it is able to detect a user’s input and process it into a signal that can be used as an input to a game or program.

Block Descriptions

- **Load Cells**
  - The load cells are the primary way the user will interface with the device. There are four (4) 50kg load cells connected to the HX711 AD Converter in the Electronics Hub subsystem. The load cells will send analog signals to the HX711 and receive 5V power from the HX711. These load cells are what the user will step on in order to send inputs to the PC. Without this subsystem, there is no way for the user to send inputs to the PC, which is the purpose of this device.
  - **Inputs:** 5V +/- 0.1V power input
  - **Outputs:** Analog voltage output to HX711 AD Converter

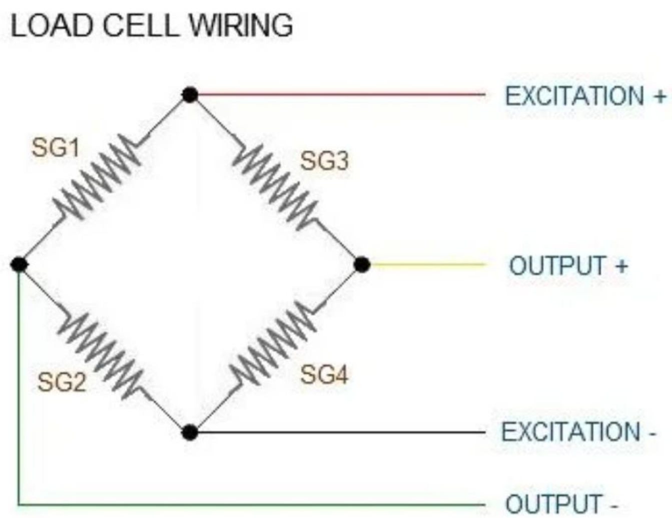

Figure 3: Load Cell Wiring Diagram

Electronics Hub

- This subsystem consists of an HX711 AD Converter, an ATMega328 Microcontroller, and a USB to Serial adapter. The HX711 takes input from the load cells and converts it into a 24-bit digital signal. This signal is sent to the ATMega328 Microcontroller where the data for each input is parsed and sent to the USB to Serial adapter via a UART signal. The Serial to USB adapter converts this signal into a USB Protocol which the computer can then process. The Serial to USB adapter is connected to a PC via a USB cable. This subsystem is important because it is what processes the signals to be sent to the PC so that the drivers can properly parse the signals.

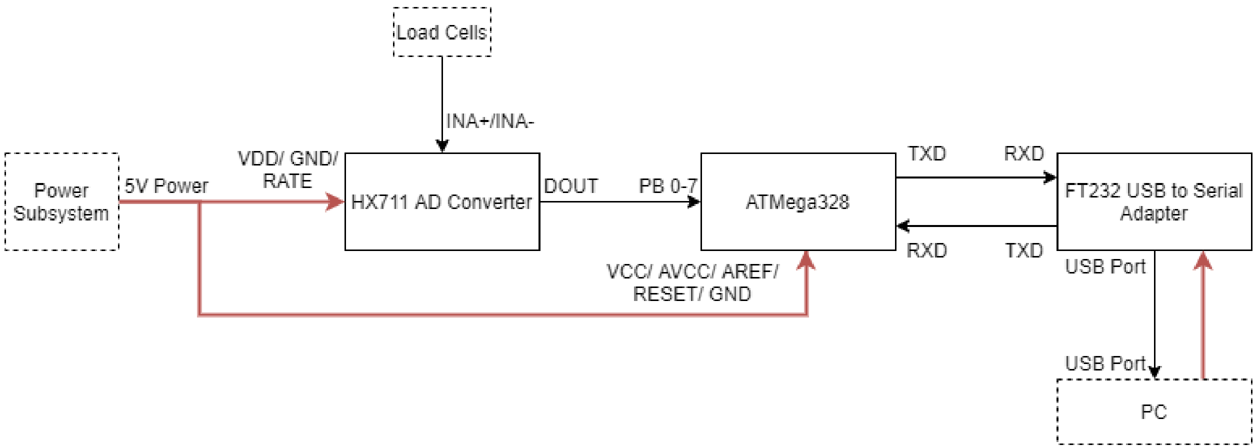

Figure 4: Electronics Hub High-Level Block Diagram

HX711 AD Converter

- **Inputs**: INA+/- from Load Cells, 5V+/- 5% from the power subsystem
- **Outputs**: Serial data output to ATMega328
- The HX711 is a part of the Electronics Hub subsystem and converts the analog signals from the load cells into a 24-bit digital signal which the ATMega328 can process.

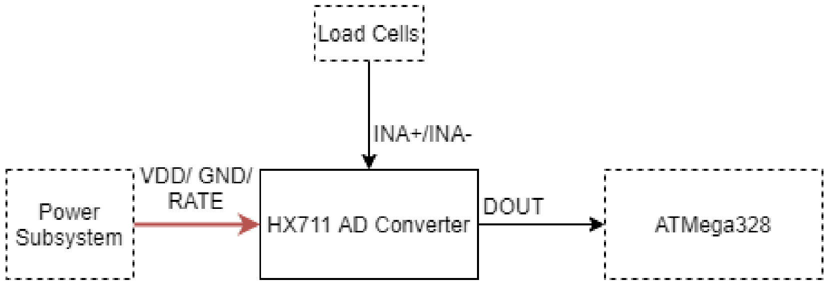

Figure 5: HX711 High-Level Block Diagram

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output（NC when not used）
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input（connect to AGND when not used）
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used）
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1: HX711 Pinout

ATMega328

- **Inputs**: A 24-bit digital signal from the HX711 AD Converter, serial input from USB to Serial Adapter, 5V +/-5% power input
- **Outputs:** Serial output to the USB to Serial Adapter
- The ATMega328 is a part of the Electronics Hub subsystem and it takes the inputs from the HX711 AD Converter and processes it so that it can be sent to the FT232 USB to Serial Adapter.

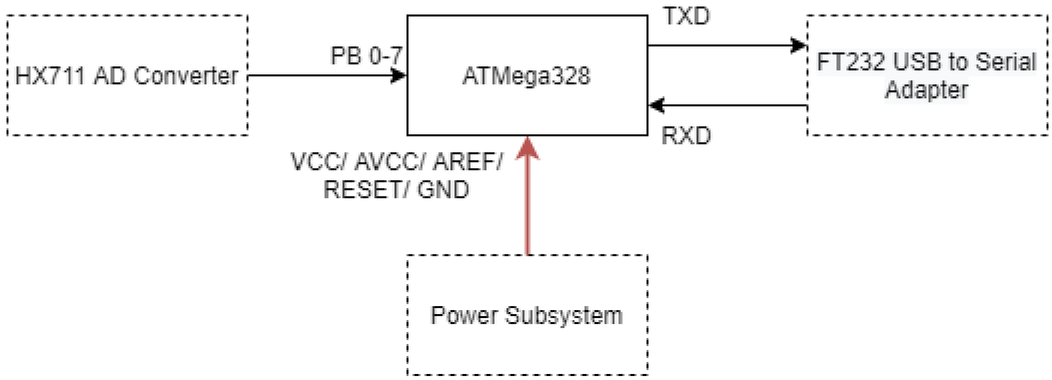

Figure 6: ATMega328 High-Level Block Diagram

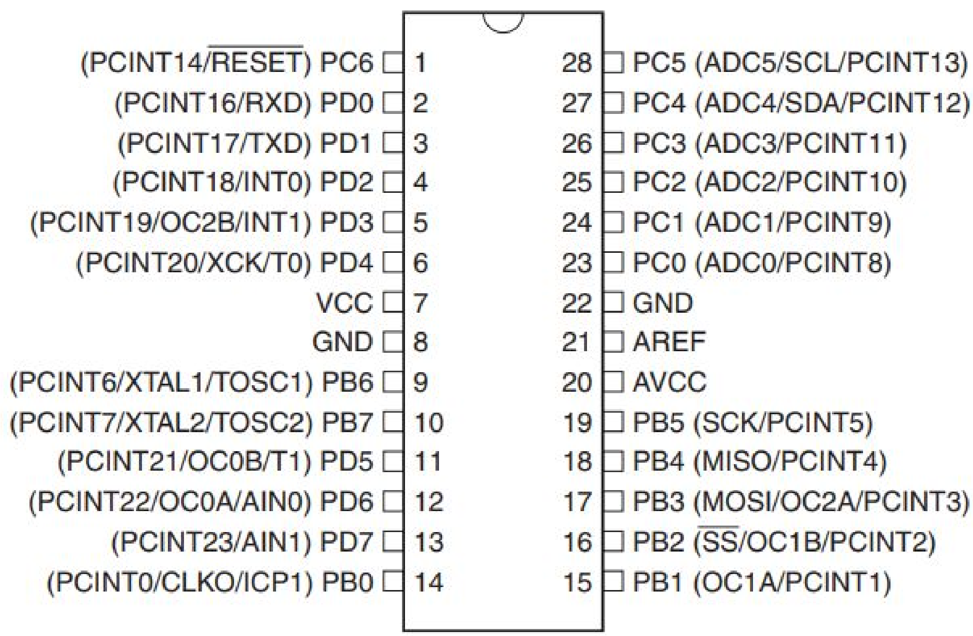

Figure 7: ATMega328 Pinout

FT232 USB to Serial Adapter

- **Inputs**: A USART Serial TTL signal from the ATMega328, data and 5V +/-5% from the PC USB port
- **Outputs:** A serial signal to the ATMega328, a USB data signal to the PC USB port containing the preprocessed button signals
- The FT232 USB to Serial Adapter is a part of the Electronics Hub subsystem and it converts and sends the processed signals to the PC so that it can be used as an input for games or applications.

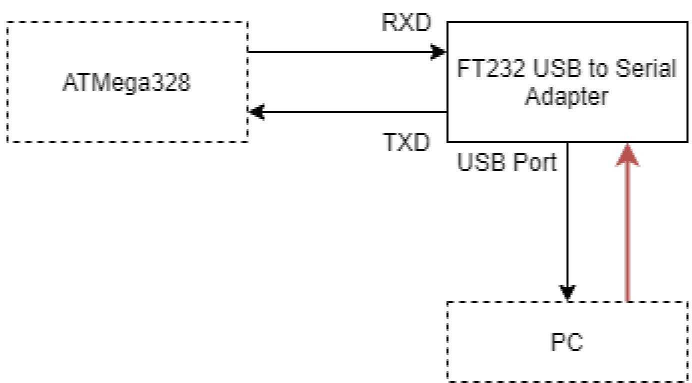

Figure 8: FT232 High-Level Block Diagram

Pin No.	Name	Type	Description
1	TXD	Output	Transmit Asynchronous Data Output.
2	DTR#	Output	Data Terminal Ready Control Output / Handshake Signal.
3	RTS#	Output	Request to Send Control Output / Handshake Signal.
5	RXD	Input	Receiving Asynchronous Data Input.

Table 2: FT232 Pinout

Power Supply

- **Inputs**: 9-Volt Battery or any 7-16V power source
- **Outputs:** 5V +/-5% for ATMega328 Microcontroller and HX711 AD Converter
- Every electronic subsystem in the device is capable of being run off of a 5V +/-5% power supply, so in order to supply this power, a 7805 Voltage Regulator will be used to bring the power from a 7-16V supply down a usable voltage. A 9-volt battery is chosen as the main power source because of its portability and ease-of-use.

The power requirements for each component is shown here:

Device	Voltage/Current Requirements
HX711 AD Converter	2.7~5.5V
ATMega328 Microcontroller	Maximum Operating Voltage: 6.0V Typical Current Consumption at Vcc=5V: 4.0mA

Table 3: Power requirements for all components on the device

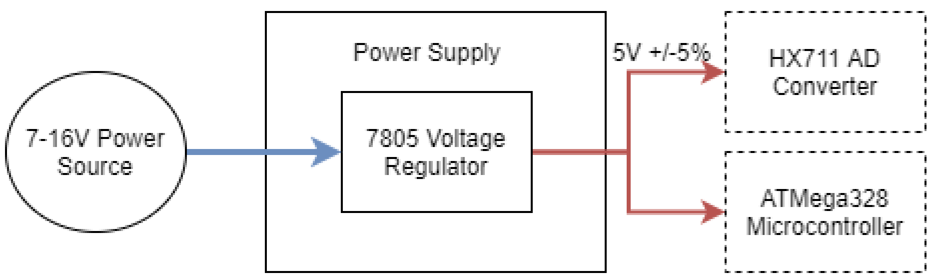

Figure 9: High-Level Power Supply Block Diagram

USB Driver and Software

- Inputs: A USB signal from the USB to Serial Adapter
- Outputs: A USB signal to the USB to Serial Adapter
- The purpose of the USB driver and software is to parse the incoming data from the microcontroller and sensors and process it into data which programs on the PC can use. This driver must be able to properly read the data sent by the microcontroller and it must expose it so that a secondary piece of software can use it to pass proper inputs into games or other software. This software must be able to set activation thresholds for each one of the pressure sensors so that users of all weight ranges can comfortably use the device.

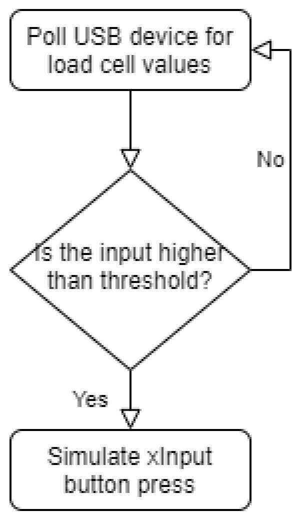

Figure 10: Software/Driver Flowchart

Requirements and Verification

Load Cells

Requirements	Verification
Each load cell must be able to accurately measure up to 50kg, with a tolerance of +/- 1kg.	Equipment: Scale, 5x10kg weights, Arduino, HX711 bridge Test Procedure: 1. Load the Arduino with code to parse the HX711 signals and connect it to the PC. 2. Connect the load cell to the HX711. (Excitation+ to VDD, Excitation- to GND, Output+ to INPA, Output- to INNA) 3. Connect the HX711 to the Arduino. (VDD to 5V, GND to GND, SCK to D5, DT to D6) 4. Place the 5x10kg weights on a scale and record the reading. 5. Place the 5x10kg weights on the load cell and record the reading output by the arduino. 6. Verify that the output from the Arduino matches the weight measured on the scale to within +/- 1kg. 7. Repeat steps 2-6 with all load cells. Presentation of Results - The results will be presented in a table which shows the measured vs actual weights for each load cell.
Each load cell must be able to withstand loads of up to 75kg for up to 30 seconds without breaking or deforming	Equipment: 3x25kg weights, Arduino, HX711 bridge Test Procedure: 1. Verify that the load cell is accurate using the previous procedure. 2. Place the 3x25kg weights on top of the load cell for 30 seconds. 3. Remove the weights and test the load cell using the previous procedure and ensure that the measured weight is

	within 1kg of the control measurement. Presentation of Results - The results will be presented in a table which shows the measured weights before and after the 75kg load was placed on it.
All wires and connectors coming from the plastic mat/pad must be able to withstand loads of up to 75kg for up to 30 seconds without breaking or deforming	Equipment: 3x25kg weights, Arduino, HX711 bridge Test Procedure: 1. Verify that the wires and connectors are functional by connecting them to the HX711 bridge and the Arduino and ensuring that they output correctly. 2. Place all the wires and connectors on a flat surface. 3. Place 3x25kg weights on top of the wires for 30 seconds. 4. Remove the weights and connect the wires to the HX711 and Arduino again and verify that they output correctly. Presentation of Results - The results will be presented in a table which shows the measured weights before and after the 75kg load was placed on it.

Table 4: Requirements and Verification Table for Load Cells

ATMega328

<table>
<tr><td>Requirements</td><td>Verification</td></tr>
<tr><td>All input signals are being received accurately from the HX711 AD Converter, serial input from USB to Serial Adapter</td><td>Equipment:
Test Procedure:
- Verify that all wires and connectors are functional and inserted in the appropriate slots
- Record input signal values
- Record output signal values
- Ensure behavior is as expected</td></tr>
<tr><td>Ensure about 5V are being received</td><td>Equipment:
Test Procedure:
- Use a voltmeter to test voltage amount
- Test if voltage stays relatively constant, within 5% of 5V, no matter the overall load</td></tr>
</table>

Table 5: Requirements and Verification Table for ATMega328

Tolerance Analysis

The critical part of this design is the communication between software and hardware. Inputs into a controller are time-sensitive because added latency when the user tries to send an input to the computer can be very off-putting and can break immersion when the user is in VR. In order to reduce latency from a user input to when the software processes it, there must be a consistent throughput of data to the PC. The biggest bottleneck in the device is going to be the HX711 AD Converter. The output data rate can be either 10Hz or 80Hz. This means that the minimum input delay between when a user sends the signal to when the computer receives it is 1/80Hz = 0.0125s, or 12.5ms.

Covid-19 Contingency Planning

With regards to the difficulties posed by Covid-19, there is a plan in place to develop the system outside of the lab. Each of the group members have their own computers, and as such are able to put together the systems on their own if necessary. One person will focus on the 3-D printing and modeling aspect of the project, since they are not in the vicinity of the campus. Because this project does require the hardware to test the software side, the other two people can focus on making the hardware subsystems work, with one person picking up the power and sensing, and the other person focusing on programming the microcontroller. When each person is done with their parts, the hardware and software will be delivered to the person on campus, where the team can zoom call and put together and bugfix the pieces. Because we only need 5v power supplies and we have our own soldering tools, we are confident that we can continue building the project without the lab equipment. To test, we also have our own multimeters to ensure that the parts are getting the required voltages and currents.

Cost and Schedule

Cost Analysis

Total Hours of Development	Cost per Hour	Total Man Hour Cost
Hour breakdown per person: 10 hours confirming documentation of parts 15 hours to finish and build design 20 hours for bug fixing of parts and design 10 extra hours of time to allow for flexibility ____________ Total hours per person: 55	$20	$20 * 55 hours * 3 people = $3300.00

Table 6: Hour Cost Analysis

Part	Part Number	Unit Cost	Package and Package Cost	Supplier	Quantity	Total Cost for Parts
ATXMega32 Microcontroller	43237-2	$15.00	______	Amazon	1	______
USBASP Programmer	______	$1.71	______	Ebay	1	____
Weight Weighing Sensor	______	______	$7.98 (4 sensors 2 HX711 amplifiers)	Amazon	2	____
Amplifier	HX711	$5.99	______	Amazon	2	____
USB-to-Serial Adaptor	FT232RL	$11.99	______	Amazon	1	____

USB Extension Cable		$6.13		Amazon	1	
Wires		$15.00		Amazon	1	
						$77.84

Table 7: Part Cost Analysis

Hour Cost	$1380.00
Part Cost	$77.84
Total Cost	$3300.00 + $77.84 = $3377.84

Table 8: Total Cost

Project Schedule

Week	Evan Miller	Vinith Raj	Justin Zhou	For all members
October 5th				Design and review PCB control signals and power supplies
October 12th	Begin designing/editing USB-to-Serial driver	Start working on model for 3D printing	Begin making code for ATMega	Review physical orientation component. Buy components. Order PCB.
October 19th	Send test signals to simulate ATMega inputs for bug fixing and validation	Finish 3D printing design for feet. Begin design for electronics hub.	Receive ATMega and programmer. Begin testing out code.	Receive Parts
October 26st		Helping other members with bug fixing over Discord.		Continue bug fixing individual parts and running validation tests.
November 2st	Physically bring parts to Justin to begin wiring together. Also bring soldering and multimeter if necessary.	Send design to Justin for printing	Receive parts from Evan and design from Vinith. 3-D print hub, and physical markers.	Further bug fixing of electronics and verification. Prepare for demonstration
November 9th				Final testing of parts. If extra time, make a short Unreal Engine program that works with our design. Prepare for demonstration.
November 16th				Demonstrate
November 23rd				Break
November 30th				Prepare for presentation

				and present
December 7th				Work on and turn in final paper

Table 9: Weekly Schedule

Ethics and Safety

Due to the project dealing with VR, some obvious health concerns do come to mind. Virtual reality headsets are known to cause nausea and anxiety, especially after extended usage. People with disabilities might be even more at risk of being affected by these problems. In addition, VR headsets can also cause eye strain, and even anxiety from all the stress of being in the virtual environment. There are all issues we need to be aware of as we develop this technology, and make sure our project doesn't make these issues any worse through both accidental and intentional misuse. With regards to testing specifically, only the project members will be testing the technology. Since there is a low development time, we can't be certain that the project is ready for beta testing with the target audience. Before reaching that point, we will thoroughly test the product on ourselves. For putting together our parts, the main danger is soldering. To mitigate the dangers of accidental burns, we will have reminders around the soldering equipment to keep it unplugged when not in use, and we will put it in a holder to mitigate the risks of fire or being brushed against.

With regards to ethical issues relevant to the project, IEEE Code of Ethics Section 1.1 [4] discusses the importance of safety and health of the public, and we believe that we should ensure that the project is safe for any user. Considering our target audience is people with disabilities, we need to be even more considerate of their needs to be safe and healthy while using this project. Furthermore, for testing, we need to be clear about the effects of the technology and the associated risks, and ensure we receive proper consent from testers via waivers. ACM Code of Ethics Section 2.7 [5] is also quite relevant to our project because it is important for the public to understand the technology we are using and how it works. This allows for projects like this to gain even more attention. Virtual reality has a lot of potential, and it could yet be expanded to help disabled individuals. By educating the public through proper conduct and safety precautions, as well as helping them to understand the consequences of this technology on society, we can collectively learn and continue our efforts in this field.

References

[1] "QuadStick FPS Game Controller." ***quadstick.com*** **QuadStick, 2020, https://www.quadstick.com/shop/quadstick-fps-game-controller. September 17th, 2020.**

[2] "Xbox Adaptive Controller." ***microsoft.com*** **Microsoft, 2020, https://www.xbox.com/en-US/accessories/controllers/xbox-adaptive-controller. September 17th, 2020.**

[3] "3D Rudder Foot Motion Controller for VR and PC Games and Applications with VR Mode, Foot Keyboard Mode, Foot Mouse Mode, Foot Joystick Mode." ***amazon.com*** **Amazon, Dec 29, 2016, https://www.amazon.com/3dRudder-Foot-Motion-Controller-Applications/product-reviews/B01MS26PFK. September 17th, 2020.**

[4] "IEEE Code of Ethics" ***IEEE, https://www.ieee.org/about/corporate/governance/p7-8.html.*** **September 17th, 2020.**

[5] "ACM Code of Ethics and Professional Conduct" ***ACM, https://www.acm.org/code-of-ethics.*** **September 17th, 2020.**