GNG 2101

Design Project User and Product Manual

Accessible X-Box Controller

Submitted by:

Quake B4.3

Gabriella El-Hajjar, 300293340

Sami Chartrand, 300283848

Raiyan Aziz, 300294770

Hanna Paik, 300310147

Saifullah Shaikh, 300293265

November 19 2023

University of Ottawa

Table of Contents

[Table of Contents ii](#_Toc153122189)

[List of Figures iii](#_Toc153122190)

[List of Tables v](#_Toc153122191)

[List of Acronyms and Glossary vi](#_Toc153122192)

[1 Introduction 7](#_Toc153122193)

[1 Overview 8](#_Toc153122194)

[1.1 Conventions 10](#_Toc153122195)

[1.2 Cautions & Warnings 10](#_Toc153122196)

[2 Getting started 13](#_Toc153122197)

[2.1 Configuration Considerations 13](#_Toc153122198)

[2.2 User Access Considerations 13](#_Toc153122199)

[2.3 Accessing/setting up the System 13](#_Toc153122200)

[2.4 System Organization & Navigation 14](#_Toc153122201)

[2.5 Exiting the System 14](#_Toc153122202)

[3 Using the System 14](#_Toc153122203)

[3.1 Xbox Controller Features 15](#_Toc153122204)

[3.2 Box 1 Features 15](#_Toc153122205)

[3.3 Box 2 Features 16](#_Toc153122206)

[3.4 Layout Customization 16](#_Toc153122207)

[3.5 Instructional Video 16](#_Toc153122208)

[4 Troubleshooting & Support 17](#_Toc153122209)

[4.1 Error Messages or Behaviors 17](#_Toc153122210)

[4.2 Special Considerations 17](#_Toc153122211)

[4.3 Maintenance 17](#_Toc153122212)

[4.4 Support 17](#_Toc153122213)

[5 Product Documentation 18](#_Toc153122214)

[Equipment list 18](#_Toc153122215)

[Instructions 19](#_Toc153122216)

[5.1 Testing & Validation 32](#_Toc153122217)

[6 Conclusions and Recommendations for Future Work 34](#_Toc153122218)

[7 Bibliography 35](#_Toc153122219)

[APPENDICES 37](#_Toc153122220)

[8 APPENDIX I: Design Files 37](#_Toc153122221)

[9 APPENDIX II: Other Appendices 38](#_Toc153122222)

List of Figures

[*Figure 1: QWAC* 10](#_Toc152877433)

[*Figure 2: Visual Model of QWAC* 11](#_Toc152877434)

[*Figure 3: QWAC* 14](#_Toc152877435)

[*Figure* 4*: Block Diagram of QWAC* 15](#_Toc152877436)

[*Figure 5: Labelled diagram of box 1 depicting all the button locations and the rapid fire.* 17](#_Toc152877437)

[*Figure 6: Diagram of box 2 depicting all the rapid-fire buttons for X, Y, A, and B* 17](#_Toc152877438)

[*Figure 7: Top for Box 1* 22](#_Toc152877439)

[*Figure 8: Base for Box 1* 22](#_Toc152877440)

[*Figure 9: Top for Box 2* 23](#_Toc152877441)

[*Figure 10: Base for Box 2* 23](#_Toc152877442)

[*Figure 11: MCU board on Controller* 29](#_Toc152877443)

[*Figure 12: MCU Board Top Wiring Diagram* 29](#_Toc152877444)

[*Figure 13: MCU Board Bottom Wiring Diagram* 30](#_Toc152877445)

[*Figure 14: Wired Controller* 30](#_Toc152877446)

[*Figure 15: 5 Button Protoboard Soldering Diagram* 31](#_Toc152877447)

[*Figure 16: Arduino I/O Soldering Diagram* 32](#_Toc152877448)

[*Figure 17: 4 Button Protoboard Soldering Diagram* 33](#_Toc152877449)

[*Figure 18: Contained Electronics* 33](#_Toc152877450)

[*Figure 19: Final Product* 34](#_Toc152877451)

[*Figure 20: Breadboard Design* 35](#_Toc152877452)

[*Figure 22: 4 Button PCB 3D Model* 40](#_Toc152877453)

[*Figure 21: Arduino I/O PCB 3D Model* 40](https://uottawa-my.sharepoint.com/personal/gelha045_uottawa_ca/Documents/User%20Manual.docx#_Toc152877454)

[*Figure 23: 5 Button PCB 3D Model* 40](#_Toc152877455)

List of Tables

[Table 1. Acronyms vi](#_Toc152876117)

[Table 2. Glossary vi](#_Toc152876118)

[Table 3. BOM (Bill of Materials) 20](#_Toc152876119)

[Table 4. Referenced Documents 38](#_Toc152876120)

List of Acronyms and Glossary

Table 1. Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| QAWC | Quake Accessible Wireless Controller |
| PCB | Printed Circuit Board |
| PLA | Polylactic Acid monomer (plastic) |
| PCR | Post Consumer Resin (plastic) |
| AXC | Accessible X-Box Controller |
| MCU | Microcontroller unit |
| LED | Light emitting diode |
| PC | Personal Computer |
| GND | Ground |
| VCC | Power supply voltage |

Table 2. Glossary

|  |  |  |
| --- | --- | --- |
| **Term** | **Acronym** | **Definition** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Introduction

This User and Product Manual (UPM) provides the information necessary for individuals with disabilities, particularly adaptive gamers who may experience limitations in their gaming performance due to conditions such as cerebral palsy, to effectively use the Accessible X-Box Controller (AXC), and for prototype documentation.

We created an accessible X-Box controller so that gaming can be a diverse experience for all users. We wanted everyone to have a fun and engaging experience while playing and not feel limited due to their condition. For a smooth user experience, the AXC prototype keeps the original X-Box controller's appearance and feel. We go into more detail about our prototype later in the manual, however a brief explanation should still give you an image of what we made. We added two boxes/cases connected to the X-Box controller and added extra and specific controls on them.

The purpose of this manual is to guide users and provide detailed insights into the Accessible X-Box Controller (AXC) prototype. It includes information on its features, how to use it, safety precautions, construction methods, materials utilized, testing protocols, and suggestions for future improvements. This document's purpose is to act as an all-inclusive resource for users and upcoming developers who take over the project. This user manual is organized into the 6 sections, in the order shown: an overview of the Accessible X-Box Controller (AXC), Features and Usage Instructions, safety Considerations, Building Process and Materials, Testing Procedures and Results, and Recommendations for Future Development.

To conclude, the users who will gain from the Accessible X-Box Controller (AXC) and future developers wishing to carry on our project can both refer to this User and Product Manual as a reference.

# Overview

Because of his cerebral palsy, our client finds it challenging to play the video games he enjoys. They may find it especially difficult to utilize the conventional Xbox controller because of its rear trigger buttons. His ability to easily reach and handle these triggers is limited by the motor restrictions associated with cerebral palsy, which negatively affects his gaming experience. Playing video games can be very difficult for those with cerebral palsy, and normal controllers may make it impossible for them to participate in this common activity. To advance inclusion and guarantee that people with motor disabilities have equal access to the enjoyment of gaming, this issue should be addressed.

Accessibility, comfort, customization, functional equivalents, and finally stability are the fundamentals for our client.

* Accessibility: To guarantee that the client can easily access all required functions, the controller must be made to take into account his physical restrictions.
* Comfort: Considering the possible difficulties caused by cerebral palsy, the controller should be cozy for extended periods of usage. Ergonomic design should be used to reduce weariness or discomfort.
* Personalization: Considering the distinct requirements of every person with cerebral palsy, the controller should be to facilitate customisation to accommodate motor skills and inclinations.
* Functional Equivalents: The extra controller buttons on the side should be to serve as the user's seamless means of executing all in-game operations. These should be functional equivalents to the conventional trigger buttons.
* Stability: To guarantee that the controller doesn’t tip over or move around while the client is using it, we added stompers at the bottom of the controller and the two extra boxes/cases that we 3D printed for the extra buttons/controls.

We deduced from his setup that he positions his controller between his legs for stability, indicating that he would feel more at ease using an actual controller as opposed to one in a box. Therefore, instead of creating a controller from scratch, we thought of updating and modifying an original controller. Since our client is unable to grip the controller, another one of our designs was inspired by the way he plays on the ground. We therefore devised a floor-flat design in which the controller's height remains the same as it did originally, but a base for the buttons is added next to it. This design decision guarantees our client can interact with the controller in a comfortable manner without requiring complicated configurations, while also improving stability.

Second, as was previously indicated, we deduced from his setup that the buttons and additional controls would be located in a separate box on the side that is attached to the controller. We also concluded that his condition prevented him from reaching the triggers in the back, which is why we are adding more buttons and controls. This is an essential addition. These buttons are arranged in a way that makes them more accessible by acting as functional substitutes for the conventional trigger buttons.

Finally, we would have to modify the controller we are updating because of the conditions set out by our customer. This offers a customized gaming experience by enabling our client to adjust button assignments to suit their particular motor skills. As an illustration, we provided the rapid-fire control function he asked for one of the additional controls we are adding based on the game he plays frequently.

Through creative planning and customizable settings, our product seeks to encourage players by reaching the unique needs of people with cerebral palsy and giving them a fun and simple gaming experience.

Final prototype is shown below:

A video game controller and a box

Description automatically generated

*Figure 1: QWAC*

The key features and functions of our product were aimed at increasing accessibility for clients with specific needs. For starters, the buttons that were added on both cases/boxes were specifically designed to replicate the same functionalities of a standard X-Box controller. For the first box/case, which is at the top of the controller, there are five buttons, each of them being a rapid fire one. The first white button from the left acts as a left button, which moves a player to the left, same functionality as the right joystick when moved towards the left. The red button acts as the main rapid-fire shooting button, once that one is pressed once it will automatically shoot several shots at once without the need to keep pressing down on the button. The first yellow button to the right of the red one, that one acts as the right button, which moves a player to the right, same functionality as the right joystick when moved towards the right. The second white button acts as the left trigger, this one allows a player to aim the gun/weapon they are using, or to use while drifting in a car game (depending on the game or controls). Finally, the second yellow button acts as the right trigger, allows a player to shoot one at a time, if he doesn’t want to use the rapid-fire control (for the car game this one also depends on which game and controls). For the second box, squarer shaped, it has four buttons, these buttons, however, act the exact same as the ABXY buttons on the X-Box controller. They are placed in the same order as they would be on a regular X-Box controller. The only difference being that the green (A) button is white due to our limited colored buttons.

In terms of architecture, the original X-Box controller is attached to two 3D-printed cases. The left button, rapid-fire trigger, right button, left trigger, and right trigger, going from left to right, are all on the rectangle shaped box/case as mentioned above. The buttons for ABXY are located in the second box/case on the right. Constructed from black construction paper, the cases have an aesthetically pleasing look. These boxes/cases are linked to the controller via cables that are organized and covered in white electrical tape for safety. This enables the smooth integration of the customized controls. A protoboard, also known as a breadboard, is located inside each box/case, and is equipped with the buttons and wires required to connect controllers and wires and integrate necessary programming, all of which contribute to a gaming experience that is both user-friendly and functional.

For a more visual and straight forward explanation here is a block diagram of our final prototype:

A diagram of a game controller

Description automatically generated

*Figure 2: Visual Model of QWAC*

## Conventions

For both Sections 2 Getting Started and 2.3 Accessing/setting up the System, there are a few steps mentioned where the reader would need to take action. It is indicated by a number list.

## Cautions & Warnings

Most of our warnings are very similar to the original Microsoft X-Box controller due to us modifying the original controller. However, our project does come with a few warnings from the added features, such as sharp edges, and wires.

WARNING: Sharp edges

Considering that our project has a single controller with two 3D printed enclosures to its right and top. Even though we tried our best to sand the rough edges of the boxes, there's still a potential that someone using them could get hurt.

WARNING: Shocking hazard

There are numerous connections connecting the three boxes, allowing us to program the correct instructions into the controller and the other two. Before we wrapped electrical tape and another material around them, these wires were visible. But, if the tape isn't applied correctly, comes off, or anything gets spilled on the controller close to the wires, there's still a chance these wires could hurt the client.

WARNING: Chocking hazard

This device may contain small pieces that could possibly choke a child or any small animals. One of the small items would be the tactile buttons we used to code. It’s very easily removable if the user wants to change the color or clean it.

WARNING: Keep out of reach of small children

Children, and others, can harm themselves or the device due to the wires connecting each box/case and controller are a bit sensitive if played aggressively. Make sure that the device and everything connected to it are out of children’s reach when done using it.

WARNING: General battery safety

The device uses disposable batteries. When disposable or rechargeable batteries are used improperly, they can explode, catch fire, leak battery fluid, overheat, or cause damage to property. They can also cause fatalities or serious harm.

Both poisonous and corrosive battery fluid can be found. If eaten, it could be dangerous or deadly and induce burns.

To lower the possibility of harm when utilizing rechargeable or disposable batteries:

* Make sure children can't access batteries.
* Dispose of the batteries safely.
* Don’t overuse them.
* If any liquid seep through it, avoid direct contact with your skin. Use a cloth to throw it away.
* Do not leave in direct sunlight.

# Getting started

1. Make sure the controller has batteries. Batteries are not included.

## Configuration Considerations

A video game controller and a box

Description automatically generated

*Figure 3: QWAC*

This prototype is a controller with three compartments. There is the main controller and two box extensions that emulate some of the main controller buttons and add rapid fire versions of some of the buttons from the main controller.

## User Access Considerations

This controller is only made to be used by one player. It is made to be played while supported on the ground or any stable surface. This product is made for those who prefer a controller that is supported by a flat surface and play first person shooter games as this controller has rapid fire features made to enhance the players shooting experience.

## Accessing/setting up the System

1. To turn the controller on hold down the logo button until the button begins to glow.
2. Make sure the controller is loaded with either AA batteries or rechargeable batteries.
3. Press and release the ‘pair’ button on the X-Box
4. Within 20 seconds press and hold the controller’s pair button until the controller’s X-Box button stays flashing, it will now pair with your X-Box and the logo should stay glowing.

## System Organization & Navigation

A diagram of a game controller

Description automatically generated

*Figure* 4*: Block Diagram of QWAC*

This product is composed of 3 parts. The Xbox controller, Box 1 and Box 2.

The Xbox controller is the main controller that contains all the possible buttons and other inputs.

Box 1 is the first extension from the controller that holds the front bumpers, triggers and the rapid-fire right trigger buttons.

Box 2 is the second extension from the controller, connected to Box 1, that holds the rapid-fire ABYX buttons.

## Exiting the System

To properly put away the system, you’d first need to turn off the controller. To do so, you’d press and hold on the Guide button, which is the button with the X-Box logo, down for six seconds. To know if it’s off or not, when the controller is on the Guide button will be a solid white color, when it’s off it reverts to its grey color. The guide button should be in the middle at the top of the controller.

# Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the QAWC.

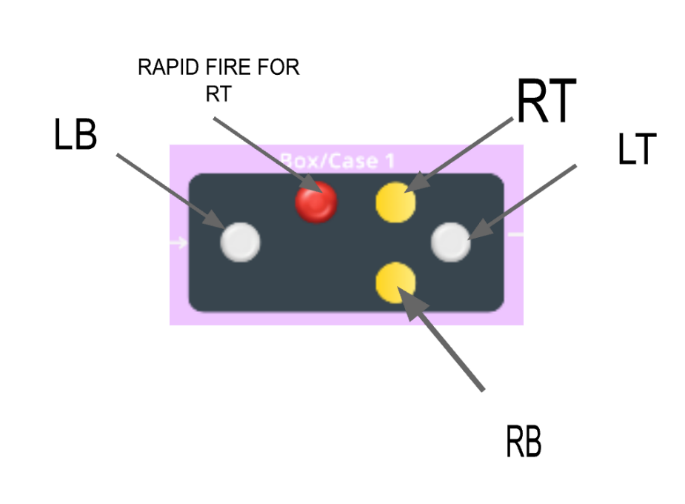
* Features of the X-Box controller
* Features of Box 1
* Features of Box 2

## Xbox Controller Features

* Features of Xbox Controller

## Box 1 Features

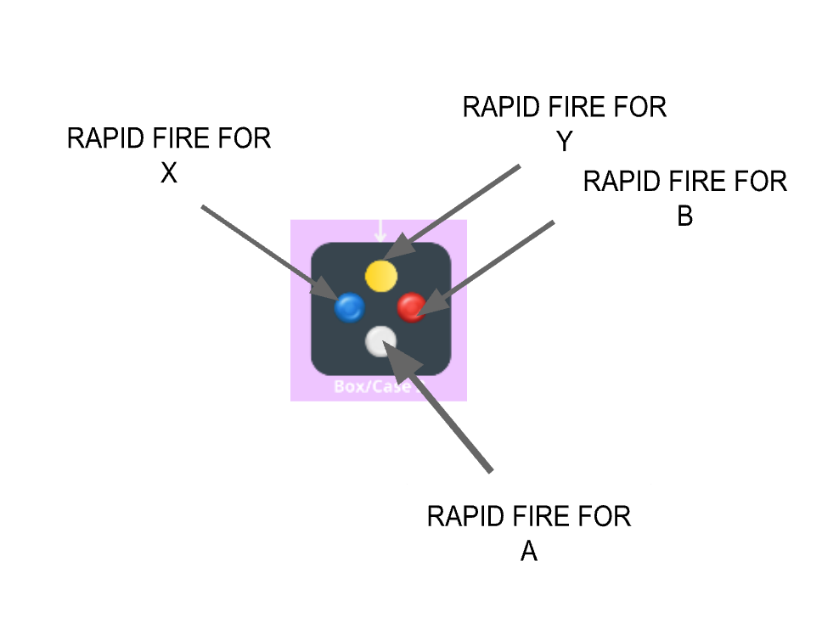
* Regular buttons
  + A crucial element within box 1 involves the repositioning of Xbox controller back buttons—LB, LT, RB, and RT, now situated in box 1 as depicted in figure 3.1. Despite this relocation, these four buttons seamlessly retain their original functionality, mirroring their behavior in a standard Xbox controller.
* Rapid fire trigger button
  + Another essential feature within box 1 is the rapid-fire trigger button, enabling users to emulate the RT button at a frequency of 8 hertz. Displayed in figure 3.1, the red button is strategically positioned at the center of the box for optimal user convenience. To activate this function, users simply press the red button once; thereafter, it automatically replicates the RT button's clicks at 8 hertz indefinitely until the red button is pressed again to deactivate the feature.



*Figure 5: Labelled diagram of box 1 depicting all the button locations and the rapid fire.*

## Box 2 Features

* Additional Rapid fire button features
  + Box 2 is designed to host rapid-fire buttons for the Xbox controller's X, Y, A, and B buttons. Operating autonomously, this feature avoids any interference with the controller's native buttons, behaving similarly to the RT rapid-fire button in figure 3.1. This implies emulation of the respective buttons at 8 hertz, with activation requiring a single press and deactivation with a subsequent press. Refer to Figure 3.2 for the visual representation of Box 2.



*Figure 6: Diagram of box 2 depicting all the rapid-fire buttons for X, Y, A, and B*

## Layout Customization

* Customizable layout
  + An essential aspect of the QAWC is its layout flexibility. The strong cable housing connecting the boxes is designed to be both robust and flexible, allowing users to customize box placements. In Figure 3, you can see the connection between the boxes through white cables, demonstrating their adaptability to various arrangements.

## Instructional Video

<https://youtu.be/KbPIwpr50Mw>

A qr code with a few squares

Description automatically generated

# Troubleshooting & Support

In the situation that the controller does not turn off the rapid-fire buttons or they don’t turn on

* + Turn off the X-Box controller by holding down on the Xbox logo button.
  + If that doesn’t work. Replace the batteries of the Xbox controller.

Refer to section 4.3 for more support.

## Error Messages or Behaviors

The wire connections between box 1 and box 2 are weaker than the rest of the container because they are jumper wires. If there are buttons on Box 2 that activate the wrong rapid-fire buttons or simply don’t work at all, it is most likely because the jumper wires are disconnected.

Refer to section 4.3 for assistance.

## Special Considerations

In the situation that the controller makes contact with water make sure that the controller is left to dry for 24 hours before reuse.

## Maintenance

Controller should be cleaned every few months with any household multipurpose cleaning product to ensure the health of the user. And debris that could get between the buttons and hinder the ability to push down on the tactile buttons should be cleared before each use.

## Support

Call Microsoft support for emergency assistance with the main controller or email [hpaik@uottawa.ca](mailto:hpaik@uottawa.ca) for assistance on the modifications.

# Product Documentation

Table 3. BOM (Bill of Materials)

|  |  |  |  |
| --- | --- | --- | --- |
| Material | Cost (tax included) | Links | Used or not |
| X-Box Controller | $35 | Facebook Marketplace | Used |
| Tactile Buttons | $10 | <https://www.amazon.ca/dp/B09V2BWFFL?psc=1&ref=ppx_yo2ov_dt_b_product_details> | Used |
| PCB Board (Version 1) | $10 | <https://makerstore.ca/shop/ols/products/copper-cladded-board>/ | Not used |
| Solid Electric Black Wire (50 ft) | $12 | <https://www.amazon.ca/URBEST-Solder-Coated-Wire-Wrapping-Celsius/dp/B01LWI20M0/ref=sr_1_5?crid=VNZBTPV26XJ1&keywords=pcb%2Bwires&qid=1697246890&sprefix=pcb%2Bwires%2Caps%2C115&sr=8-5&th=1> | Used |
| Electrical Components for Rapid Trigger Version 1 (not Arduino based) | $20.60 | <https://www.digikey.ca/short/fh2ttw0c> | Not used |
| Adhesive Rubber Bumper Pads | $12.33 |  | Used |
| Braided Cable Sleeve | $13.55 |  | Used |
| Capacitors (10x) | $15.81 |  | Used |
| 12 x 8 cm Protoboard Boards (2x) | $14.11 |  | Used |
| Total cost | $143.4 |  |  |

The final cost of our final prototype: $112.8

However, a few of the materials we used were not purchased due to it being from our university:

* PLA
* Arduino
* Jumper wires
* Resistors
* Protoboards (2x)
* Solder materials

### Equipment list

* Ultimaker 2+ 3D Printer
* Arduino Uno
* USB A to USB cable (printer/Arduino cable)
* PC with internet connection

Wiring the Controller:

* T-8 and T-6 safety Torx screw drivers
* Prying tool (flat head screwdriver)
* The Xbox controller
* 10 30cm length solid electric black wires
* Solder
* Soldering iron
* Scotch tape
* Electrical tape
* Wire cutter
* Wire stripper

Wiring Protoboards:

* Solid electrical wire
* Solder
* Soldering iron
* Scotch tape
* Electrical tape
* Glue gun
* Glue sticks for glue gun
* Wire cutters
* Wire stripper
* Protoboards
* Xbox controller from wiring controller
* Arduino with code
* 12 male to male jumper wires
* 12 female to female jumper wires

Final touches:

* Construction paper
* Electrical tape
* Sanding or grinding tools.
* Scissors
* Wire wrapping

Testing:

* Oscilloscope
* Multimeter
* Breadboard
* 10 330-ohm resistors
* 10 LEDs
* 12 jumper wires
* Micro USB to USB cable
* USB A to USB cable (printer/Arduino cable)
* Arduino with code
* PC with internet connection

### Instructions

Manufacturing the Casing:

Important note: Must download the UltiMaker Cura software before starting.

\*Download at your own risk\*

<https://ultimaker.com/software/ultimaker-cura/>

Printing Instructions:

1. Ensure that the UltiMaker 2+ printer is selected.
2. Import .stl file into UltiMaker Cura.
3. Ensure that **Material** is set to PLA and **Nozzle Size** is set to 0.8mm
4. Set **Draft Quality** to 0.2mm, **Infill Density** to 20%.
5. Ensure that **Generate Supports** is turned off and set **Build Plate Adhesion** to skirt.
6. Use the tab on the left to rotate the part and lay flat on the plate. (Set the same as pictures)

A computer screen shot of a computer

Description automatically generated

*Figure 7: Top for Box 1*

A computer screen shot of a computer

Description automatically generated

*Figure 8: Base for Box 1*

A computer screen shot of a computer

Description automatically generated

*Figure 9: Top for Box 2*

A computer screen shot of a computer

Description automatically generated

*Figure 10: Base for Box 2*

1. Once the part is set as desired, press the **Slice** button and it will build a preview and estimated printing time.
2. It is recommended to also go to the **Preview** tab at the top, this tab will show the entire printing process. It is recommended to use this feature to find mistakes before printing.
3. Insert a full-size SD card into computer, once inserted, Cura will automatically detect it and will allow to export the file onto the SD card.
4. Eject the SD card and remove from computer, then insert the SD card into the UltiMaker 2+ 3D printer.
5. Default settings on the printer are recommended for best printing results.

Important note: Please monitor the entire printing process as printing errors may occur.

Arduino Code:

1. Get an Arduino Uno and upload the following code:

const int buttonPinR2 = 3; // Pin where the button is connected

const int buttonPinY = 4;

const int buttonPinX =5;

const int buttonPinA = 6;

const int buttonPinB = 7;

const int bulbPinR2 = 8;

const int bulbPinY = 9;

const int bulbPinX = 10;

const int bulbPinA = 11;

const int bulbPinB = 12; // Pin where the bulb/LED is connected

bool bulbStateR2 = false; // Current state of the bulb

bool lastButtonStateR2 ; // Assume switch is open because of pull-up resistor

unsigned long lastDebounceTimeR2 = 0; // Last time the button input was toggled

unsigned long debounceDelayR2 = 50;

bool bulbStateY = false; // Current state of the bulb

bool lastButtonStateY; // Assume switch is open because of pull-up resistor

unsigned long lastDebounceTimeY = 0; // Last time the button input was toggled

unsigned long debounceDelayY = 50;

bool bulbStateX = false; // Current state of the bulb

bool lastButtonStateX; // Assume switch is open because of pull-up resistor

unsigned long lastDebounceTimeX = 0; // Last time the button input was toggled

unsigned long debounceDelayX = 50;

bool bulbStateA = false; // Current state of the bulb

bool lastButtonStateA; // Assume switch is open because of pull-up resistor

unsigned long lastDebounceTimeA = 0; // Last time the button input was toggled

unsigned long debounceDelayA = 50;

bool bulbStateB = false; // Current state of the bulb

bool lastButtonStateB; // Assume switch is open because of pull-up resistor

unsigned long lastDebounceTimeB = 0; // Last time the button input was toggled

unsigned long debounceDelayB = 50; // the debounce time in milliseconds

bool readingR2 = 0;

bool readingX = 0;

bool readingY = 0;

bool readingA = 0;

bool readingB = 0;

void setup() {

pinMode(buttonPinR2, INPUT); // Initialize the button pin as input (external pull-up resistor is used)

pinMode(bulbPinR2, OUTPUT); // Initialize the bulb pin as an output

lastButtonStateR2 = digitalRead(buttonPinR2);

pinMode(buttonPinX, INPUT); // Initialize the button pin as input (external pull-up resistor is used)

pinMode(bulbPinX, OUTPUT); // Initialize the bulb pin as an output

lastButtonStateX = digitalRead(buttonPinX);

pinMode(buttonPinY, INPUT); // Initialize the button pin as input (external pull-up resistor is used)

pinMode(bulbPinY, OUTPUT); // Initialize the bulb pin as an output

lastButtonStateY = digitalRead(buttonPinY);

pinMode(buttonPinA, INPUT); // Initialize the button pin as input (external pull-up resistor is used)

pinMode(bulbPinA, OUTPUT); // Initialize the bulb pin as an output

lastButtonStateA = digitalRead(buttonPinA);

pinMode(buttonPinB, INPUT); // Initialize the button pin as input (external pull-up resistor is used)

pinMode(bulbPinB, OUTPUT); // Initialize the bulb pin as an output

lastButtonStateB = digitalRead(buttonPinB);

Serial.begin(15000);

}

void loop() {

// Read the state of the button into a local variable:

readingR2 = digitalRead(buttonPinR2);

readingX = digitalRead(buttonPinX);

readingY = digitalRead(buttonPinY);

readingA = digitalRead(buttonPinA);

readingB = digitalRead(buttonPinB);

Serial.print(digitalRead(buttonPinR2));

Serial.print(digitalRead(bulbPinR2));

//if button press is not the same as

if (readingR2 != lastButtonStateR2){

lastButtonStateR2 = readingR2;

if (readingR2 == LOW){

if (bulbStateR2 == true){

bulbStateR2 = false;

}

else{

bulbStateR2 = true;

}

}

}

if (readingX != lastButtonStateX){

lastButtonStateX = readingX;

if (readingX == LOW){

if (bulbStateX == true){

bulbStateX = false;

}

else{

bulbStateX = true;

}

}

}

if (readingY != lastButtonStateY){

lastButtonStateY = readingY;

if (readingY == LOW){

if (bulbStateY == true){

bulbStateY = false;

}

else{

bulbStateY = true;

}

}

}

if (readingA != lastButtonStateA){

lastButtonStateA = readingA;

if (readingA == LOW){

if (bulbStateA == true){

bulbStateA = false;

}

else{

bulbStateA= true;

}

}

}

if (readingB != lastButtonStateB){

lastButtonStateB = readingB;

if (readingB == LOW){

if (bulbStateB == true){

bulbStateB = false;

}

else{

bulbStateB = true;

}

}

}

// If the bulb is supposed to be blinking, handle the blinking

if (bulbStateR2) {

// 8 Hz blinking means the bulb should toggle every 125 milliseconds

// (1000 milliseconds divided by 8)

pinMode(bulbPinR2, OUTPUT);

if ((millis() % 125) < 62) { // On for half the time

digitalWrite(bulbPinR2, LOW);

} else {

digitalWrite(bulbPinR2, HIGH);

}

} else {

// Make sure the bulb is off if it's not supposed to be blinking

pinMode(bulbPinR2, INPUT);

}

if (bulbStateX) {

// 8 Hz blinking means the bulb should toggle every 125 milliseconds

// (1000 milliseconds divided by 8)

pinMode(bulbPinX, OUTPUT);

if ((millis() % 125) < 62) { // On for half the time

digitalWrite(bulbPinX, LOW);

} else {

digitalWrite(bulbPinX, HIGH);

}

} else {

// Make sure the bulb is off if it's not supposed to be blinking

pinMode(bulbPinX, INPUT);

}

if (bulbStateY) {

// 8 Hz blinking means the bulb should toggle every 125 milliseconds

// (1000 milliseconds divided by 8)

pinMode(bulbPinY, OUTPUT);

if ((millis() % 125) < 62) { // On for half the time

digitalWrite(bulbPinY, LOW);

} else {

digitalWrite(bulbPinY, HIGH);

}

} else {

// Make sure the bulb is off if it's not supposed to be blinking

pinMode(bulbPinY, INPUT);

}

if (bulbStateA) {

// 8 Hz blinking means the bulb should toggle every 125 milliseconds

// (1000 milliseconds divided by 8)

pinMode(bulbPinA, OUTPUT);

if ((millis() % 125) < 62) { // On for half the time

digitalWrite(bulbPinA, LOW);

} else {

digitalWrite(bulbPinA, HIGH);

}

} else {

// Make sure the bulb is off if it's not supposed to be blinking

pinMode(bulbPinA, INPUT);

}

if (bulbStateB) {

// 8 Hz blinking means the bulb should toggle every 125 milliseconds

// (1000 milliseconds divided by 8)

pinMode(bulbPinB, OUTPUT);

if ((millis() % 125) < 62) { // On for half the time

digitalWrite(bulbPinB, LOW);

} else {

digitalWrite(bulbPinB, HIGH);

}

} else {

// Make sure the bulb is off if it's not supposed to be blinking

pinMode(bulbPinB, INPUT);

}

}

Ensure that the Arduino Uno has the required number of pins specified in the code block above.

Wiring The Controller:

1. Use a prying tool to separate the grip covers from each side of the controller.
2. Remove the battery cover and take a T-8 safety Torx Screwdriver and remove the 5 screws holding the controller case together.
3. Remove the bottom of the case to reveal the bottom of the MCU board.

A close up of a circuit board

Description automatically generated



*Figure 11: MCU board on Controller*

1. Solder 3 30cm length solid electrical wires to the three points of the board as shown in the diagram below. Make sure that the wires are leading to the right.

A green circuit board with a black background

Description automatically generated



*Figure 12: MCU Board Top Wiring Diagram*

1. Use a T-6 safety Torx and remove the 2 screws on the lower side of the MCU board
2. Pull the MCU board out of its socket and gently without breaking the black and gray wires attached to the sides of the MCU board, turn it over to reveal the top of the MCU board.
3. Solder 7 30cm length solid electrical wire to the seven points of the board as shown in the diagram below. Make sure that the wires are leading to the right.

A green circuit board with many wires

Description automatically generated



*Figure 13: MCU Board Bottom Wiring Diagram*

1. Use electrical tape to line up the wires so they are coming out, to the right side of MCU board (left side of the controller when face up) and label each end of the wire with a piece of scotch tape so you know which wire is which.
2. Plug the MCU board back in and screw it back down. Screw back on the back casing and make sure to not pinch the wires. Also put back on the battery casing

A piece of paper with a piece of paper on a desk with a computer mouse

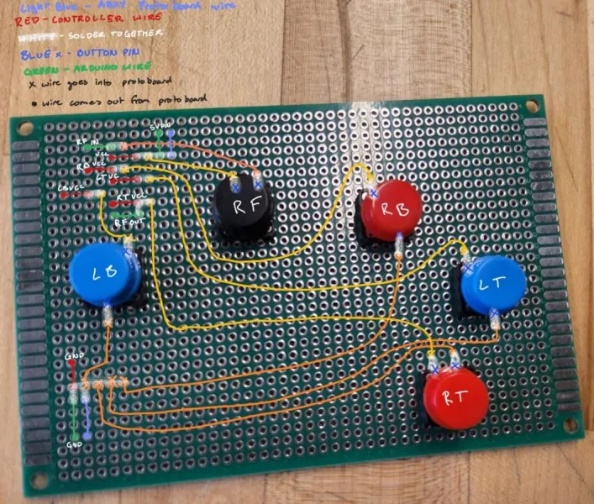
Description automatically generated

*Figure 14: Wired Controller*

1. Now wiring the controller is done.

Soldering the Protoboards:

1. Using the 3 diagrams below solder electrical components to complete the wiring of the controller.
2. All the wires should be the solid electrical wire except for the wire labeled PIN. There are 12 pin wires.
3. For these wires, take 12 male to male jumper wires and connect them to 12 female to female jumper wires so you get 12 double length jumper wires.
4. Use hot glue to secure the male and female wire jumper connections.
5. Cut the unattached female head off each of the wires and solder them to the PIN locations on the protoboards. The male head will be connected to the Arduino at its designated pin location.

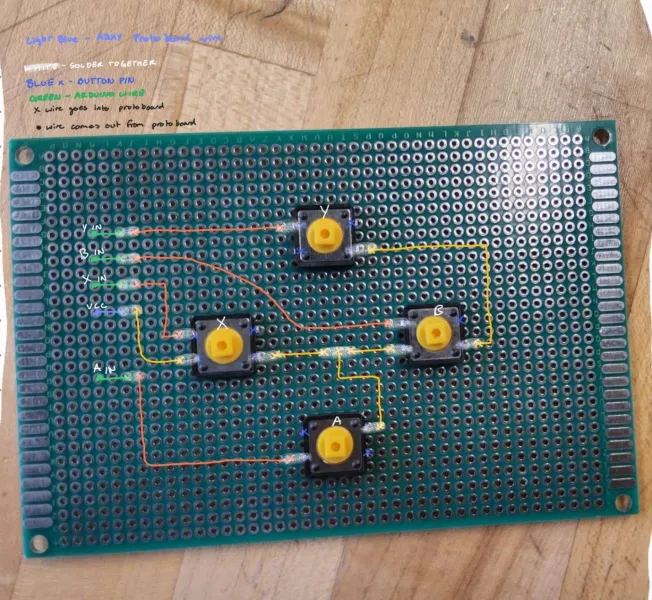


*Figure 15: 5 Button Protoboard Soldering Diagram*

A close-up of a circuit board

Description automatically generated

*Figure 16: Arduino I/O Soldering Diagram*



*Figure 17: 4 Button Protoboard Soldering Diagram*

1. Soldering the protoboards is now complete.

Placing the electrical components in the boxes:

1. Take the Arduino and the 4 button protoboard and place them in the box with the 4-button design. Use scotch tape to secure the protoboard to the supports of the case.
2. Align the wires to go out the left hole in the side of the box.
3. Take the Arduino protoboard and the 5 button protoboard design and place them in the box with the 5 buttons design.
4. Use scotch tape to secure the protoboard to the supports of the case.
5. Align the wires that go into the box to go out it’s left side wire hole and align the wires that go out of the box to go out of the box through the right-side wire hole.
6. It should look something similar to what is shown below:

A video game controller and a controller

Description automatically generated

*Figure 18: Contained Electronics*

1. Now containing the electronics is complete.

Final Touches:

1. Place the top of the cases on to the controller boxes. If the 3D printing did not create a smooth finish on the top of the casing, use construction paper to layer the outside of the casing and cut holes for the button caps.
2. After finishing the top casing place the button caps on to the buttons. Make sure you follow the correct colour coding. Then label the buttons using a sharpie.
3. Take white electrical tape and wire wrapping and wrap the lose wires to make them strong and held together.
4. The final product should look like this.



*Figure 19: Final Product*

## Testing & Validation

ARDUINO TESTING:

After Uploading the Arduino code use a bread board and set up a circuit seen in the figure below for 5 buttons. The input and output pins used on the Arduino go from pin 3 to pin 12. The inputs pins are from 3-7 and the output pins goes from pins 8-12. The Arduino code is step up in a away that input pin 3 is paired with output pin 8. Input pin 4 is paired with output pin 9. The next input pin is paired with the next output pin. For each button the output pin and the input pin should be paired.

Upon setting up the circuit on the breadboard, power the Arduino by connecting it to a PC. Then press each button once. Upon powering the Arduino, the output LEDs should be on. This means the output is supplying a 3.3 voltage. The Xbox controller registers a button press when the output is low. i.e when the output LED is off. When you initially press a button the input LED above the button should light up and the output pin should be flickering at an 8 hertz pulse.

Hook up an oscilloscope to the output wire and ground to measure and ensure the pulse is 8 Hertz. When a button is pressed for the second time the LED should stop flickering and stay on. If this is successful for every button, then the Arduino testing is completed. A screenshot of a computer

Description automatically generated

*Figure 20: Breadboard Design*

CONTROLLER WIRING TESTING:

After wiring the controller, test to see if the solder connections are secure by using a micro-USB to USB cable to power the controller and connect it to a PC. Go on a web browser and search for a controller tester. [Gamepad Tester - Check Controllers and Joysticks Online (hardwaretester.com)](https://hardwaretester.com/gamepad)

Run the controller tester and connect each button or trigger wire to the GND wire and see if the controller tester registers the action.

Use a multimeter to measure the voltage of the VCC wire. If it is measured to be 3.3V when the controller is on the connection is secure.

After soldering the protoboards together, connect the controller to a controller test on the PC like before and press each button on the protoboards and see if they register on the tester. If they don’t, use a multimeter to probe the buttons and solder connections to find where the issue is coming from. Hook up the negative end to the ground and the other side to the spot you want to probe.

What you should see on the multimeter:

* The buttons should be supplied with 3.3 volts while the triggers should have around 160mV suppled to their buttons.
* The VCC should have 3.3 Volts.
* The Arduino recognises input when they are high (3.3 volts) but the controller recognises input when they are low (grounded).
* When the output for the Arduino is flickering the voltage flickers from 1.5V to 160mV otherwise it is 3.3V.

If all buttons register on the controller tester the wiring testing is completed.

CASE TESTING:

After printing all 4 parts with the 3D printer, analyze each part to see any flaws that are visible. If the surface isn’t smooth, that’s not a problem, what should be of concern would be any fibers in the structure that are not combined properly. These defects may cause a failure to occur with the piece. If such problems are visible, recycle the part and print another one.

For physical testing, ensure that the top and bottom pieces fit together. It should be very tight but not impossible to fit. These tight tolerances are there for replacing any need for screws or other connections. A second test is to do a slight bend test, by pressing down on the center of each piece. If the parts are printed properly, there shouldn’t be a lot of flexibility.

Finally, to ensure that all parts are functioning, assemble the parts with the buttons to ensure that they fit inside the holes and don’t scratch or cause other issues. That concludes all testing for the case.

# Conclusions and Recommendations for Future Work

In conclusion,the main lesson we all learned as a group was time management, this was the most difficult and important part of our project. Due to all of us having a full and different schedule than each other, it was hard to set up a specific date and time for us to meet up each week. Whenever one teammate would have a busy week most of the group would also have a busy week. In this case some work would be pushed for a while and others there would be a little amount of people able to complete the work.

The work we had during our prototype was split into three different sections, software the coding part, electrical part, and designing part. Which we split within our team as well. During the software part, we created a code for the buttons/controls so they can be replaced by some of the controls from the X-Box controller. In the electrical section we opened the X-Box controller and see where we need to connect all the wires needed so the buttons can connect to them. This part also required us to solder all the wires in the controller, and on the protoboards for the buttons. Finally for the designing part we went through a variety of prototypes, from a box to a stand to upgrading the controller itself and adding the buttons the client needed on the side in two different boxes. The design part had also required us to format it a certain way due to how our client sits down and uses the controller. We came up with the idea to have three parts of the design, one being the controller the second being a box with the left and right triggers, and rapid-fire buttons at the top, and finally the last box being placed at the top right with the four main buttons of an X-Box controller (XYAB). All these sections were connected through wires which we had to wrap with electrical tape, and still have it so it’s flexible to move around.

The most productive avenues for future work that we would suggest to the other groups who want to continue and improve upon our work, would be accessibility testing, standards, materials, ergonomics, and finally cost optimization. We suggest performing detailed accessibility testing to make sure that the updated controller complies with accepted accessibility requirements. Since our client has the same disability, we also want to stimulate research into other accessibility features and alternatives that can enhance the user experience even more for people with different levels and types of cerebral palsy. Regarding materials and ergonomics, we might suggest looking at alternate materials or coatings that could improve the customized controller's durability, comfort, and appearance. Additionally, look at ergonomic designs that take into account various hand sizes and grip preferences in order to make the controller comfortable to use for extended periods of time. In conclusion, we would advise looking into methods to reduce costs without losing the improved controller's usability and quality, thus making it more accessible to a larger user base.

If we had a few more months, we would add and change a few things such as: getting a PCB design from China, adding modulo connections, increasing our button quality, having a more ergonomic design, key stability, resin casing (PCR), adding retractable cables, and heat shrinking wire wrapping. Compared to using protoboards, exporting the PCB design from China will offer cost-effectiveness and efficiency, enabling us to reduce production costs while keeping quality standards. Adding modulo connectors, such USB, and detachable cables improve user ease by making cable replacement easier. This would also help our prototype have more variety in our controller. Making ergonomic frame designs a top priority ensures user comfort throughout long hours of use, enhancing functionality in general and increasing the controller's appeal. By guaranteeing the stability and uniformity of key presses, essential for accurate input in gaming scenarios, key stabilizers improve performance. Improving the button's quality not only makes the prototype more durable, but it also makes pressing the button enjoyable and offers a better sense of touch. Polycarbonate resin (PCR) is used in casing to protect internal components while maintaining an aesthetically pleasing appearance, striking a balance between strength and longevity. Retractable cords improve portability by making it easier for users to store and avoid tangling or clutter. By using heat shrink wrapping for controlling cables, clean, safe cable protection is provided, which reduces strain and damage on the cables.

The item we chose to abandon because of lack of time was the use of PCB’s. We chose to opt for protoboards, or more commonly known as breadboards, even though PCBs are important to add since they are easier to use for a variety of reasons. With PCBs, lengthy wire connections and soldering are not necessary, providing a more professional and efficient option. They improve the prototype's overall appearance and streamline the wiring procedure, giving it a more dependable and effective base. Protoboards fulfilled an immediate need, but incorporating PCBs would result in a finished product that is more refined and cleaner.

# Bibliography

References for metrics:

“Xbox Adaptive Controller.” *Microsoft*, <https://www.microsoft.com/en-ca/d/xbox-adaptive-controller/8nsdbhz1n3d8>. Accessed 7 October 2023.

“Axis 4 Pro – Blue Tip Gaming.” *Blue Tip Gaming*, <https://www.bluetipgaming.com/product/axis-4-pro/>. Accessed 7 October 2023.

“SUBPAC M2X.” *SUBPAC*, <https://subpac.com/subpac-m2x/>. Accessed 7 October 2023.

“Optima Joystick - Video Gaming Assistive Joystick.” *Pretorian Technologies*, <https://www.pretorianuk.com/optima-joystick>. Accessed 7 October 2023.

“Single Handed Xbox One Controllers | Web Portal for Benjamin J Heckendorn.” *Ben Heck*, <https://www.benheck.com/xb1s_controllers/>. Accessed 7 October 2023.

“QuadStick FPS game controller — QuadStick.” *QuadStick*, [https://www.quadstick.com/shop/quadstick-fps-game-controller. Accessed 7 October 2023](https://www.quadstick.com/shop/quadstick-fps-game-controller.%20Accessed%207%20October%202023).

“Product safety warnings and instructions.” *Microsoft Support*, <https://support.microsoft.com/en-us/topic/product-safety-warnings-and-instructions-726eab87-f471-4ad8-48e5-9c25f68927ba>. Accessed 28 November 2023.

APPENDICES

# APPENDIX I: Design Files

Table 4. Referenced Documents

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Document Location and/or URL** | **Issuance Date** |
| 4.3 Accessible Controller | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |
| Base 1 | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |
| Base 2 | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |
| Box 1 | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |
| Box 2 | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |
| Arduino Sketch | <https://makerepo.com/saifshaikh/1751.43-accessible-controller> | Nov 13 2023 |

# APPENDIX II: Other Appendices

Possible PCB designs for future work on the controller design:

Using PCBs instead of protoboards would improve the designs manufacturability.

A blue circuit board with white text

Description automatically generatedA blue circuit board with yellow dots and numbers

Description automatically generated

*Figure 22: 4 Button PCB 3D Model*

A blue circuit board with white text

Description automatically generated

*Figure 21: Arduino I/O PCB 3D Model*

*Figure 23: 5 Button PCB 3D Model*