



uOttawa

Faculté de génie
Faculty of Engineering

Deliverable H

User and Product Manual

Prepared by Group Z11

Full name	Student Number
Hawa Salamata Ndiaye	300092449
Ryan Rambali	300235460
Juan Torres	300198775
Layane El Merini	300288563
Geraud Ndagang	300105965

Presented to

Dr. Emmanuel Bouendeu, Ph.D, P.Eng

as part of the course

GNG 2101Z - Introduction to product development & management for engineers & computer
scientists

Faculty of Engineering

Department of Mechanical Engineering

23rd July 2023

Abstract

At the beginning of the course, our mission was to find a solution to assist our client, who loves playing video games and was unfortunately diagnosed with limited mobility and tremors. Our goal was to create a controller suited to him, with additional features, making it more effective. However, we had to work within certain constraints, particularly budget and time, which were the pivotal factors influencing the development of this product.

After careful consideration, we collectively decided on a potential solution that could meet our client's expectations. Employing various techniques and applying our acquired knowledge, we embarked on the journey of creating a high-quality product. This involved the creation of three different prototypes, with the third one ultimately becoming the final iteration.

The end result was « X-Ability » – an accessible and programmable remote controller that is compatible with both Xbox and PC for disabled gamers. The case was crafted using laser cutter technology and MDF wood material for durability and sturdiness. Additionally, we went the extra mile and included macro buttons for rapid fires and other useful functionalities.

By combining innovation, creativity, and problem-solving skills, we successfully delivered a product that met our client's requirements and improved his gaming experience.

Table of Contents

Abstract	2
Table of Contents	3
List of Figures	5
List of Tables	6
List of Acronyms and Glossary	7
1 Introduction	8
2 Overview	9
2.1 Conventions	10
2.2 Cautions & Warnings	10
3 Getting started	11
3.1 Configuration Considerations	11
3.2 User Access Considerations	11
3.3 Accessing/setting up the System	11
3.4 System Organization & Navigation	12
3.5 Exiting the System	14
4 Using the System	15
4.1 Game Mode	15
4.1.1 Gamepad Buttons/Joysticks	15
4.1.2 Macro Buttons	16
4.1.3 Trigger Modes	17
4.2 Program Mode	18
4.2.1 Reset EEPROM	18
4.2.2 Configure Macro Buttons	18
5 Troubleshooting & Support	19
5.1 Error Messages or Behaviors	19
Usage	19
Production	19
5.2 Special Consideration	19
5.3 Maintenance	20
5.4 Support	20
6 Product Documentation	21
6.1 Subsystem 1 of prototype: Physical	21
6.1.1 BOM (Bill of Materials)	21

6.1.2 Equipment list	21
6.1.3 Instructions	21
6.2 Subsystem 2 of prototype: Circuit Design	24
6.2.1 BOM	24
6.2.2 Equipment list	24
6.2.3 Instructions	24
6.3 Subsystem 3 of prototype: Software	28
6.3 Equipment list	28
6.3.3 Instructions	28
6.4 Testing & Validation	29
7 Conclusions and Recommendations for Future Work	31
8 Bibliography	32
9 APPENDIX I: Design Files	33
10 APPENDIX II: Other Appendices	34

List of Figures

Figure 2.1 Final prototype	8
Figure 3.1 USB Cable plugged into the controller	11
Figure 3.2 Game Button Layout	12
Figure 3.3 Macro Button Layout	12
Figure 3.2 Game Button Layout	14
Figure 3.3 Macro Button Layout	15
Figure 4.1 Option Button	16
Figure 4.2 Mode Button	17
Figure 6.1 concepts	21
Fig 6.2 - Overall view of the controller	22
Fig 6.3 - prototype 1, 2 and 3 respectively	22
Fig 6.4 - Complete Overview of Circuit	24
Fig 6.5 - Joystick Subsection of Circuit	24
Fig 6.6 - Arduino Subsection of Circuit	25
Fig 6.7 - 74HC166 PISO Shift-Register Subsection of Circuit	25
Fig 6.8- Push Button Subsection of Circuit	25
Fig 6.9 - functional Circuit Decomposition	26
Fig 6.10 - Prototype 1, 2 and 3 of the circuit subsystem	26
Fig 6.11 - Flow Chart of software functionalities	27
Fig 6.12 - Buttons correspondence in the code	28
Fig 6.13 - Validation	29
Fig 6.14 - Mario Kart	30

List of Tables

Table 1. Acronyms	6
Table 2. Glossary	6
Table 3 BOM Physical	20
Table 4 Equipment Physical	20
Table 5 BOM Circuit	23
Table 6 Equipment Circuit	23
Table 7 Equipment Software	27
Table 8 Testing and Validation	29
Table 9. Referenced Documents	32

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
EEPROM	Electrically Erasable Programmable Read-Only Memory
LED	Light Emitting Diode
MDF	Medium-density fibreboard
PCB	Printed Circuit Board
PISO	Parallel In - Serial Out

Table 2. Glossary

Term	Acronym	Definition
Directional pad	D-pad	form of directional control
Emulator		hardware or software that enables one computer system to behave like another computer system.
Input Lag		time delay between pressing a button and the assigned action occurring
Macro		Sequences of events (such as keystrokes, mouse clicks, and delays) that can be played back to help with repetitive tasks.
Type A Universal Serial Bus	Type A USB	data cable with thin, rectangular cross-section
XInput		Library for handling controller input

1 Introduction

An Xbox controller is a compact electronic device used for gaming purposes. Its primary function is to enhance the gaming experience by providing precise control and responsiveness. It allows gamers to interact, play, and fully engage in various activities. It is said that millions of gamers face challenges in gameplay, and only a quarter of them use a suitable device due to its limitations in everyday gaming situations (Statista Canada)

Most Xbox controller users don't often encounter issues with precision and usability, but for gamers with disabilities there are significant obstacles, like accessibility. There are accessible controllers available on the market, however some gamers may be reluctant to invest in those due to cost concerns, or owning a fully functional one but still desiring better functionalities.

This is where our remote controller comes into play.

This User and Product Manual (UPM) provides the necessary information for the gaming community, the Tetra Society or any party concerned to effectively use our product "X-Ability" and for prototype documentation. It was developed following the completion of our product as part of the course GNG2101Z - Introduction to product development & management for engineers & computer scientists, which started on May 1st, 2023 and ends on July 23, 2023 with the submission of the UPM. This project involved designing an accessible controller compatible with Xbox and PC for disabled gamers. This manual is structured in seven main axes and refers only to the use of the product «X-Ability», thus it cannot be used in any other way.

2 Overview

A member of the Tetra Society is currently seeking a solution to be able to play video games (an Xbox adapter controller) as he has significant tremors.

The current tool they use is the Elite controller series 2, however they are not able to accurately control the joysticks and the ability to 'rapid fires' is very difficult. Other accessible controllers do exist, like the Axis 1 Pro however they are either very expensive or don't quite meet the needs of the users. In response to this challenge, Tetra Society reached out to us for assistance in designing a robust, programmable and accessible controller, compatible with Xbox and PC for gamers with limited mobility.



Figure 2.1 Final prototype

To address this issue, our team decided to conceptualize a functional controller and included macro buttons as well as sticky triggers for the joysticks. The accessible controller perfectly meets the customer's expectations, given its responsiveness, ease of use and the number of buttons included. The product is characterized by a sturdy box made of MDF wood and a total of three joysticks and twenty buttons available to use, making it a comprehensive and practical solution for the customer.

2.1 Conventions

No convention is used in this report. The text is written in such a way that any user can follow the text without prior knowledge or knowledge of any conventions.

2.2 Cautions & Warnings

This product is not waterproof. It should not be exposed to water. It is also necessary here to let the user know that the tool uses an Arduino microcontroller.

3 Getting started

X-Ability is a fairly easy controller to access, as it requires minimal input from the user to be able to play any game they desire. All they need to do is to connect it to the gaming device of their choice and remap the buttons to their desired configuration, as we will explain below. After playing, the user can just unplug the controller, all the while the configured controls and macros will be saved for the next gaming session.

3.1 Configuration Considerations

To use this product, the user must have a gaming platform on PC (native or emulator) or an Xbox controller, as well as games to play on each device. They will also need a Type A USB cable to connect the controller to those devices. If playing on an Xbox, the user must also purchase an adapter USB key that will allow compatibility with the console. The exact setup will be explained in section 3.3.

3.2 User Access Considerations

This product is mainly targeted towards disabled gamers, especially those that have fine motor skill difficulties and coordination problems. However, many other users may benefit from XAbility, such as small children who are learning how to play video games, people with slower reflexes, and gamers in general as the programming features may appeal to them.

This product was designed to not have any restrictions in accessibility for disabled gamers. If this product is used by children, however, they may need assistance from a guardian to understand how program modes are operated.

It is also important to note that this controller is only compatible with Xbox and PC, therefore PS4/5 players should purchase the required adapter for it to work.

3.3 Accessing/setting up the System

Firstly, to access the controller, it should be plugged into a PC by using a standard USB cable (Type A). If using an Xbox, the controller should be plugged into an Xbox adapter key (sold separately) which would be connected to the console through a USB cable.

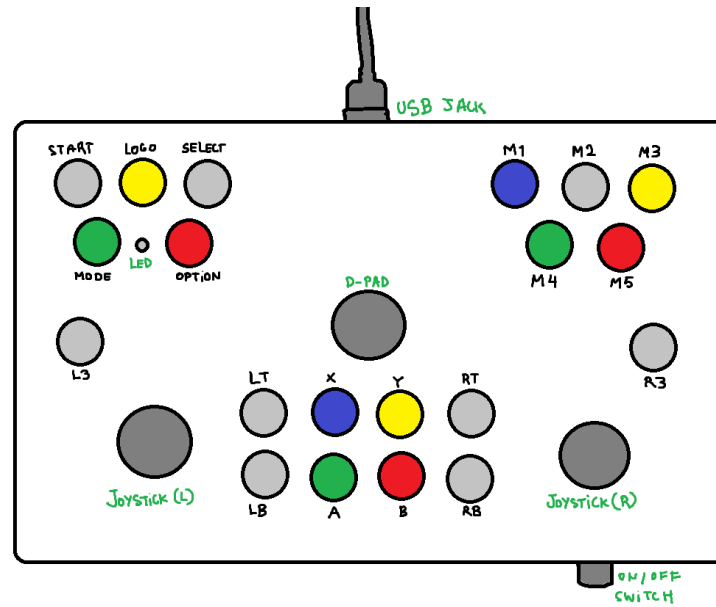


Figure 3.1 USB Cable plugged into the controller

Then, by opening the settings menu for the software used (Native game or emulator for PC or Xbox accessories app for Xbox), the user can assign commands to each button (mapping), including triggers ‘LT’ and ‘LB’, left and right joysticks and D-Pad. The user can start playing once they are satisfied with the mapping of the buttons.

The lights on the buttons can be turned on and off thanks to the switch on the front of the controller, if that disturbs their vision.

3.4 System Organization & Navigation

The controller consists of a wooden box and has 3 main sections: the game buttons, the macro buttons and the programming buttons. It also has an on/off button to turn on the lights and a USB jack at the back to connect it to various devices.

3.4.1 Game Buttons

This section consists of all the buttons (including joysticks) needed to play a game. There are eight in the lower part of the box’s top (closer to the user), 2 joysticks on each side and a D-Pad on top of them. There are also 3 at the top left corner that stand for ‘START’, ‘LOGO’ and ‘SELECT’, although they may be remapped to act out other commands. Their only purpose is to be pressed during the game.

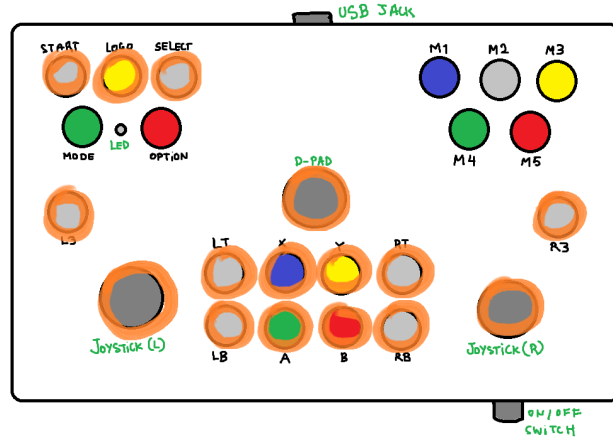


Figure 3.2 Game Button Layout

3.4.2 Macro Buttons

This section consists of the five macro buttons found on the top right corner of the box's top. Using the program mode, a sequence of up to 5 commands can be assigned to each button. By clicking a macro button, the sequence program will automatically play.

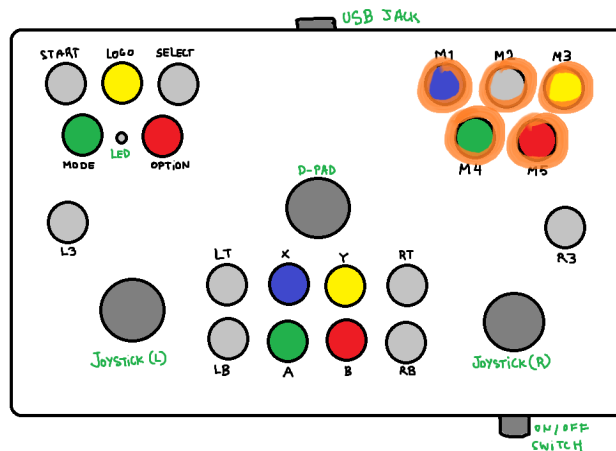


Figure 3.3 Macro Button Layout

To program a macro button, the 'MODE' button must be pressed. This will switch the mode to programming mode and the LED at the top left corner will switch to white. Then, in intervals of 2 seconds, the buttons of the user's choice should be pressed. The colour of the LED will change every time a command is input. Once the desired commands are inputted, the sequence should be confirmed by clicking the 'OPTION' button.

3.4.3 Programming Buttons

Aside from the macro functionality described above, the programming buttons ‘MODE’ and ‘OPTION’ can also be used to set up rapid fire and trigger modes. Both modes can be set up using the ‘OPTION’ button and they are part of a loop that goes like Normal-Trigger-Rapid Fire. This means that to set up, for example, a rapid fire button, the program will go through the Trigger mode first. The user must repeat the process a second time for the button to be assigned Rapid Fire mode.

3.5 Exiting the System

To turn off the controller, you only need to unplug it from the device it is connected to, as it is its power source. All programmed information should be saved after the current use.

4 Using the System

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

4.1 Game Mode

This is the default mode of the controller when it is first powered. The status LED will be green to indicate that you are in game mode. In game mode, you can use the controller to send commands to the game system you are connected to. You can also change trigger modes and activate macros. Finally, you can send signals using the joysticks and 8-way directional D-PAD stick.

4.1.1 Gamepad Buttons/Joysticks

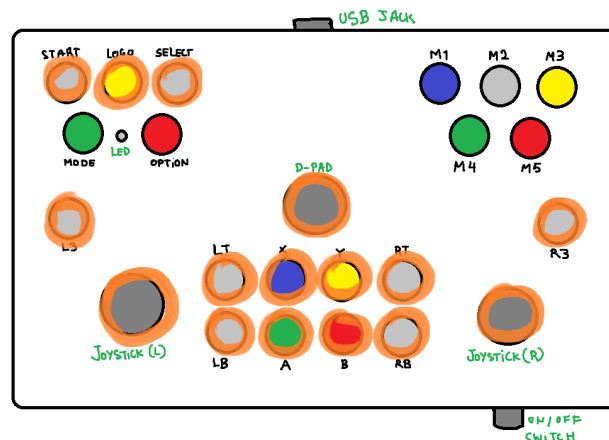


Figure 3.2 Game Button Layout

Figure 3.2 is repeated above and shows the buttons and joysticks that are referenced for this subsection. When you press any of the buttons circled above, it will send a game pad button signal to the game system's corresponding input. For example, let's say I pressed the X button on the controller. Then, the game system will receive that game signal as if I pushed the X button on a regular Xbox controller. This is consistent for all the gamepad buttons except for “START”, “SELECT”, “LOGO”. Those buttons are mapped respectively to what Xbox calls “MENU”, “VIEW”, “XBOX”.

As for the Left and Right joysticks, they are mapped according to the Xbox’s joystick layout, and in the same orientation. Lastly, for the D-PAD, it is mapped as an 8-Way Directional D-PAD, meaning not only does it have the directions, “LEFT”, “RIGHT”, “UP”, “DOWN”; but it

can hold press up to 2 of these directions at the same time, by moving the joysticks to one of the four “corners” of the movable area.

4.1.2 Macro Buttons

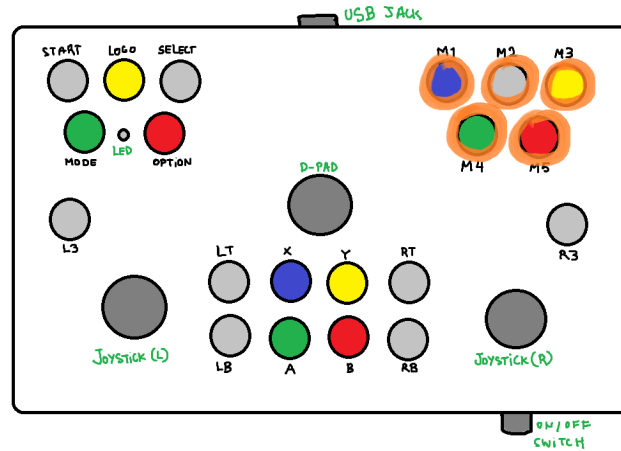


Figure 3.3 Macro Button Layout

Figure 3.3 is repeated above and shows the locations of the macro buttons. These buttons can be configured in Program Mode (explained in subsection 4.2.1). When you press one of these buttons, the controller will send the signals of up to 5 game pad buttons, sequentially, for 200 milliseconds per button. For example, say you have configured M4 to have X → Y. When you press M4 in game mode, first the game system, will receive a signal for the X button being pressed down. It will be pressed for 200 milliseconds. Then, the game system, will receive a signal for the Y button being pressed down. It will be pressed for 200 milliseconds. While the macro is executing, you will not be able to send more inputs from any other button or joystick.

4.1.3 Trigger Modes



Figure 4.1 Option Button

While in game mode, you can switch between 3 different trigger modes. The modes are in the following order: “Normal”, “Sticky”, “Rapid Fire”. You can switch between those modes by pressing the option button on the controller. The option button is indicated in figure 4.1 by the red arrow. Once you press the option button, the controller will cycle through the trigger modes in order, and when you are at the last mode, it will cycle back to the first. When you press the option button, the status LED will turn white for 2 seconds. During this time, you have to take your finger off of the button, and while the LED is white, the controller will not receive inputs.

In the “Normal” trigger mode, the trigger buttons “LT” and “RT”, will act like normal like any other buttons. When in “Sticky” trigger mode, and you press a trigger button, the status LED will turn white. This indicates the same thing as before, make sure you take your finger off of the button. Then if the trigger was “OFF” it will turn and stay “ON” and vice versa. Finally, in the “Rapid Fire” trigger mode, and you press a trigger button, the status LED will turn white. This indicates the same thing as before, make sure you take your finger off of the button. The associated trigger will rapidly be pressed and released in software. You can stop it, by pressing the associated trigger button again.

4.2 Program Mode

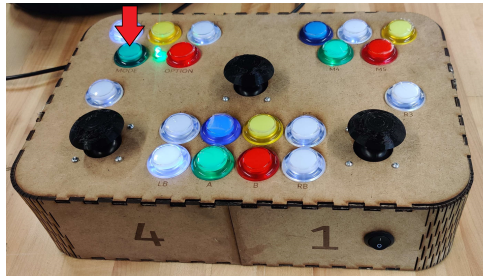


Figure 4.2 Mode Button

Program mode allows you to configure the macro buttons and reset the EEPROM (storage/memory for the configurations). To get the program mode, press the MODE button indicated in figure 4.2, the status LED will turn white. Make sure you take your finger off of the button. After 2 seconds, the status LED will RED, this indicates you are in program mode. To go back to game mode, press the MODE button again, the status LED will turn white. Make sure you take your finger off of the button. After 2 seconds, the status LED will green, this indicates you are in game mode.

4.2.1 Reset EEPROM

If you are in program mode and not currently setting a macro, when you press the OPTION button indicated in figure 4.1, the status LED will turn white. Make sure you take your finger off of the button. After 2 seconds, the status LED will RED. This means you have reset the EEPROM and your macro buttons will be reset.

4.2.2 Configure Macro Buttons

If you are in program mode, you can press any of the 5 M buttons in the top right, indicated in figure 3.3. The light will turn white, and 2 seconds later it will turn a different colour. Then you can press any gamepad button except for start select Logo. The light will turn white and then 2 seconds later it will change again. You can use up to 5 gamepad buttons. If you select all 5, when you press the fifth button, the colour after it changes to white, will be red. The macro will be set. If you want a shortened macro, when the colour has changed when you pressed your final game pad button after white, you can press the OPTION button. The LED will go white, then after 2 seconds it will be red, the macro has been set. Change back to game mode to try out your macro buttons.

5 Troubleshooting & Support

5.1 Error Messages or Behaviors

When using or producing the controller, you may run into unfavourable behaviours. Here are some you may encounter, they have been divided into two categories:

Usage

- If you change modes while setting macros before pressing the options button to end it, you will not be able to set more macros. This is easily fixed by simply unplugging and replugging the controller.
- When in program mode, or between trigger modes, if a button is held for longer than two seconds, the button will be triggered again even if you don't depress the button.
- If too much force is exerted on the joysticks, the cover may pop off. To fix this, simply push the button cover back onto the joystick module.
- If the controller is not giving outputs as soon as you plug it in, give it some time as it might need some time to boot up and function properly, also try pressing a trigger button.
- If the controller is not being recognised, check your device manager on your computer to make sure the usb device is being recognised as a game controller.

Production

- When uploading code to the arduino, double press the reset button on the arduino halfway through uploading.
- Make sure the correct port and board(Arduino Leonardo W/XInput) are selected.

5.2 Special Consideration

Here are some special considerations regarding the use of the controller and its behaviour

- If you want to use the controller as a ps4 controller on windows, you will need to download DS4Windows to let your computer read the XInput as PS4 input known as DS4
- To test controller input on a computer, go to gamepad-tester.com and press buttons on the controller or move the joysticks to test their input visually (the logo/home

button cannot be tested here). If you replug the controller, you will have to open a new tab of gamepad-tester.com to test again.

- **Do not flip controller upside down or on its side**

5.3 Maintenance

Maintenance of the controller will ensure it will run properly and look good for as long as possible. Here is some maintenance that should be done

- To clean the controller, wipe with a damp (not wet) microfiber cloth. You may use a multipurpose cleaner, but don't apply directly to the controller. DO NOT use abrasive or rough cleaners or scrubs.
- Avoid moving the controller too much or too rough as it may cause issues with the electronic components inside.
- When moving the controller, handle it with care.
- If joystick covers come off often, consider applying superglue inside the joystick cover to secure it to the module.
- To clean buttons, use cotton swabs with alcohol to wipe around and in between buttons.

5.4 Support

In the case of emergency assistance or support, contact our team through the makerlab. Email Justine at makerlab@uottawa.ca. In the email please state the name of the product, our team name (GameAbility -Z11), steps taken to produce error, and your expected result and we will try to help you out to the best of our abilities.

6 Product Documentation

6.1 Subsystem 1 of prototype: Physical

6.1.1 BOM (Bill of Materials)

ID #	Name	Description	Unit	Quantity	Unit Cost	Extended Cost
1	LED	RGB light to display mode info	Unit	1	\$2.00	\$2.00
2	Arcade Button Type 1	Arcade gamepad button	Unit	10	\$3.83	\$38.30
3	Joystick Cover	3d-printed cover for accessibility	g	30	\$0.15	\$4.50
4	MDF Wood	wood panels to be laser cut for box	m ²	0.56	\$17.85	\$10.00
Total product cost (without taxes or shipping)						\$54.80
Total product cost (including taxes)						\$61.90

Table 3 BOM Physical

6.1.2 Equipment list

ID	Name	Usefulness
1	3D printer	Use for make 3D print of the joysticks
2	Laser cutter	Use to cut the MDF wood for the box

Table 4 Equipment Physical

6.1.3 Instructions

To be able to build this prototype, we have follow some few steps which are:

- Establish a list of customer needs and prioritize them.

- make problem statement and Benchmarking to see what already exist as adapt remote controller
- Establish different concepts on the way we can build the remote controller and select one of them

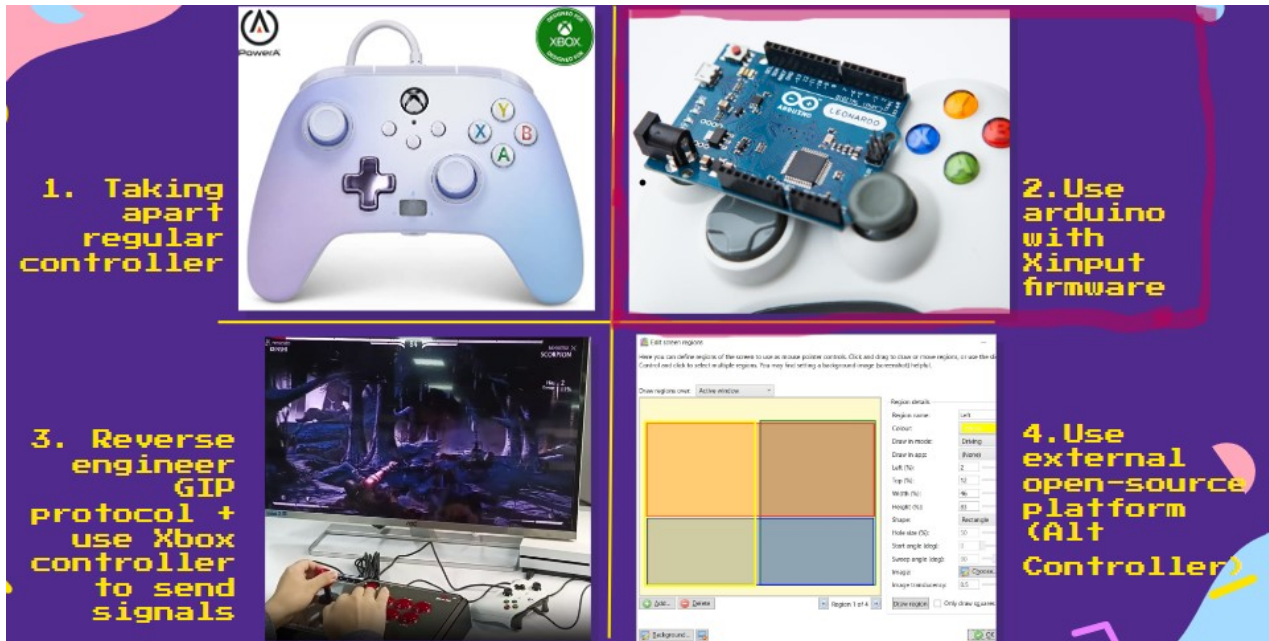
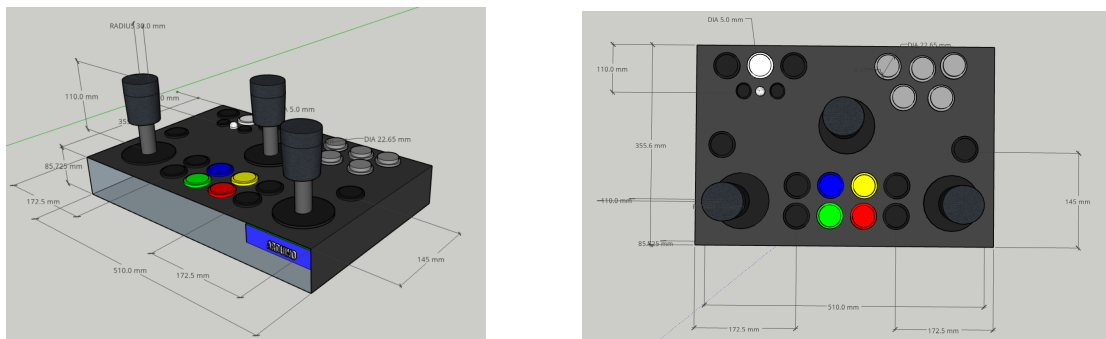


Figure 6.1 concepts

- Make a detailed design of our prototype



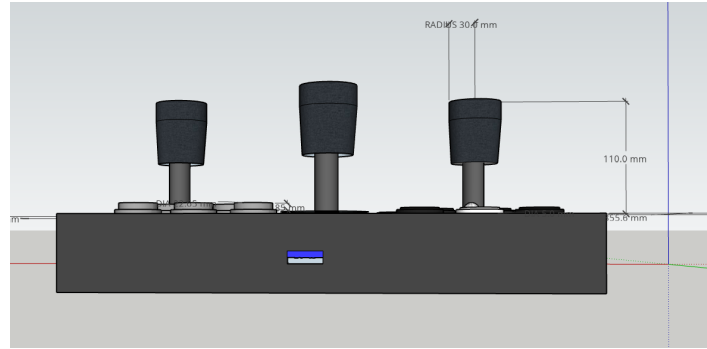


Fig 6.2 - Overall view of the controller

- After that we started making a prototype plan; where on the first prototype, we established an estimation of the 3D printing time for the Joystick. For the second Prototype, we work the case stability and add some few buttons. Finally, for the final prototype we made a new case, a new joystick and brought the different components together.



Fig 6.3 - prototype 1, 2 and 3 respectively

6.2 Subsystem 2 of prototype: Circuit Design

6.2.1 BOM

ID #	Name	Description	Unit	Quantity	Unit Cost	Extended Cost
1	Arduino Leonardo	Main logic board	Unit	1	\$15.00	\$15.00
2	PCB	PCB to connect electronics to	Unit	1	\$5.00	\$5.00
3	Header pins	Pins to connect to the PCB	Unit	3	\$0.30	\$0.90
4	Resistors 10K	Resistors for buttons	Unit	20	\$0.01	\$0.20
5	Resistors 220	resistors for RGB LED	Unit	3	\$0.01	\$0.03
Total product cost (without taxes or shipping)						\$21.13
Total product cost (including taxes)						\$24.30

Table 5 BOM Circuit

6.2.2 Equipment list

ID	Name	Usefulness
3	Weller soldering	Use to connect wires to buttons
4	CNC PCB mill	Use to create the circuit on the PCB

Table 6 Equipment Circuit

6.2.3 Instructions

To be able to build this prototype, we have follow some few steps which are:

- Same as the physical subsystem, we established a list of customer needs and prioritized them. Then, we made problem statement and Benchmarking to see what already exist as circuit for adapt remote

- We make a detailed design of the circuit with some assumptions about the power for the microchips which will be enough, the minimal input lag, and the remappable macro

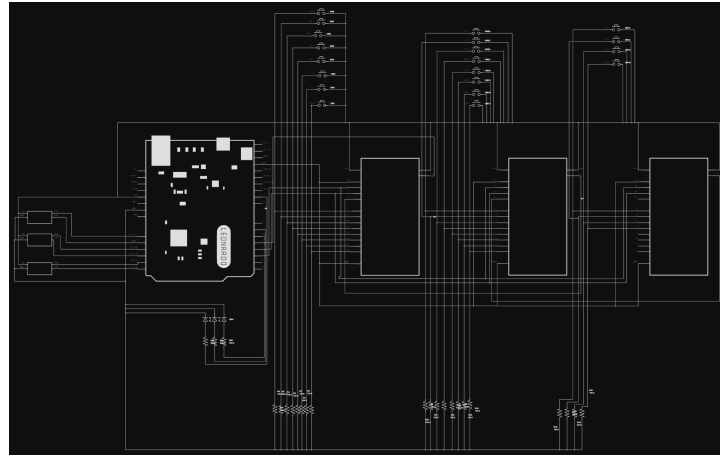


Fig 6.4 - Complete Overview of Circuit

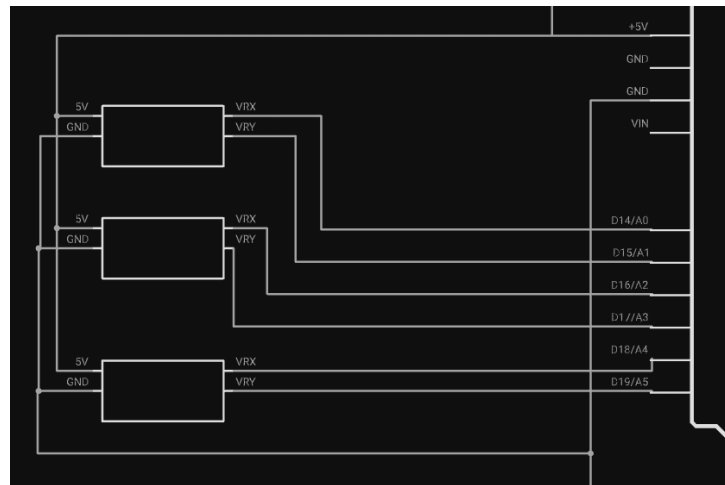


Fig 6.5 - Joystick Subsection of Circuit

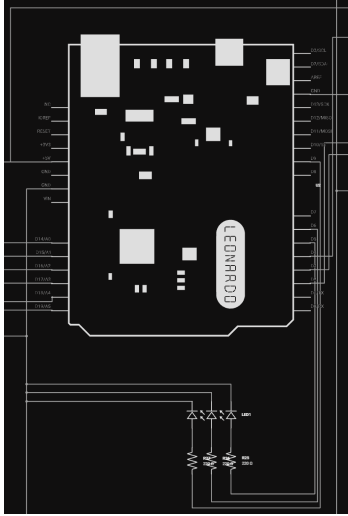


Fig 6.6 - Arduino Subsection of Circuit

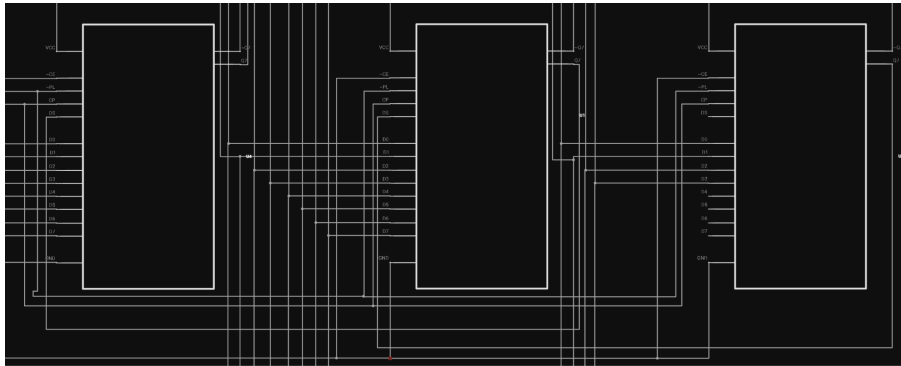


Fig 6.7 - 74HC166 PISO Shift-Register Subsection of Circuit

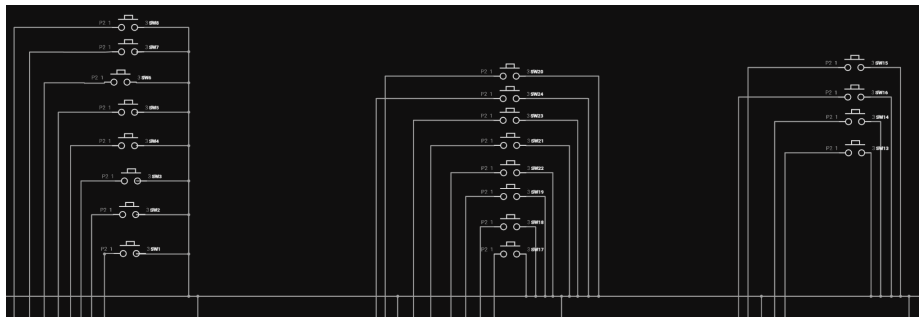


Fig 6.8- Push Button Subsection of Circuit

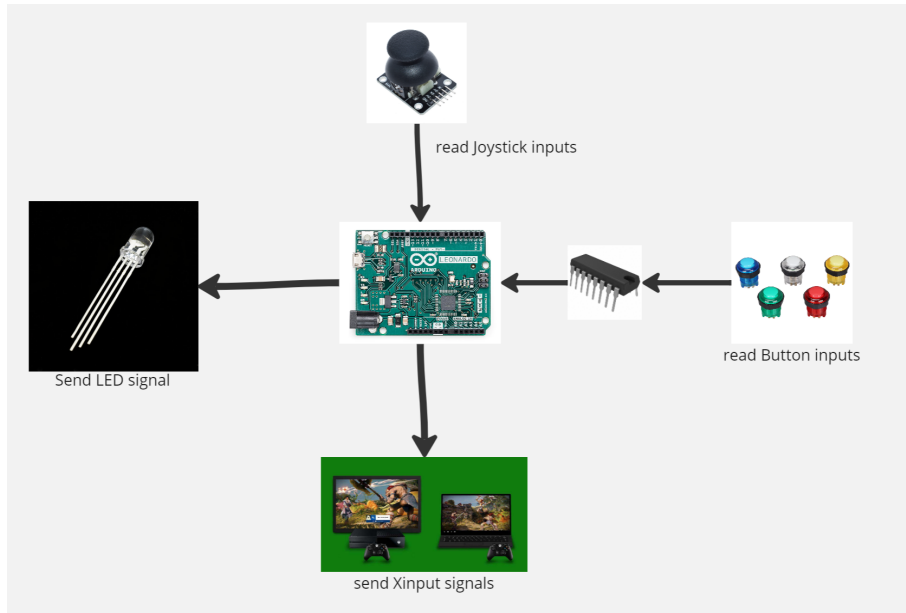


Fig 6.9 - functional Circuit Decomposition

- After that we made a prototype plan. For the first one we work on input lag timer, clock rate, power delivery for the shift registers, and the functionality of the pin on the shift register. For the second Prototype, we work on the functionality of 3 buttons, Joystick cover functionality. For the final prototype we work on the functionality of all the buttons.



Fig 6.10 - Prototype 1, 2 and 3 of the circuit subsystem

6.3 Subsystem 3 of prototype: Software

6.3 Equipment list

ID	Name	Usefulness
1	Windows laptop	a computer to write the code
2	Arduino IDE 2.0.	will get and keep the code to make the buttons and Joysticks works
3	Xinput Arduino AVR Firmware	To emulate our Xbox controller using an USB capable microcontroller and the ArduinoXInput library

Table 7 Equipment Software

6.3.3 Instructions

Concerning the Software part of our prototype, we start by making a flowchart to expand how our system will work. We wrote code where we associate each button to a constant and a functionality that will execute when pressing on it.

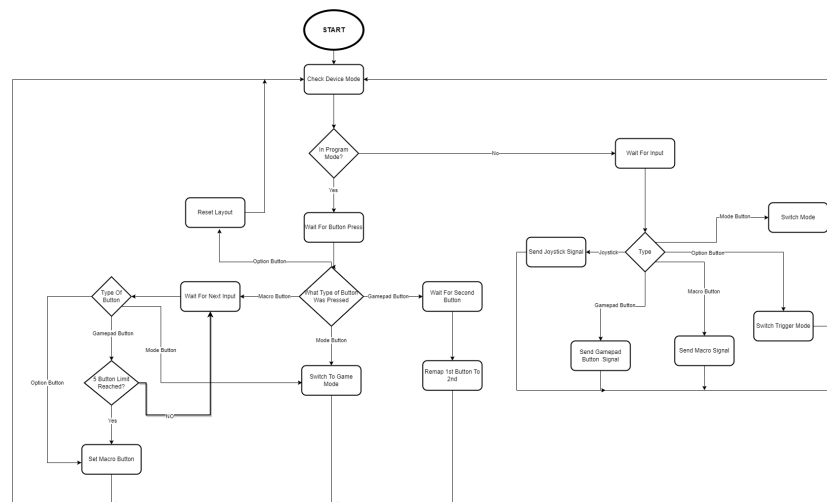


Fig 6.11 - Flow Chart of software functionalities

BUTTON_LOGO	0	BUTTON_LOGO = 0,
BUTTON_A	1	BUTTON_A = 1,
BUTTON_B	2	BUTTON_B = 2,
BUTTON_X	3	BUTTON_X = 3,
BUTTON_Y	4	BUTTON_Y = 4,
BUTTON_LB	5	BUTTON_LB = 5,
BUTTON_RB	6	BUTTON_RB = 6,
BUTTON_BACK	7	BUTTON_BACK = 7,
BUTTON_START	8	BUTTON_START = 8,
BUTTON_L3	9	BUTTON_L3 = 9,
BUTTON_R3	10	BUTTON_R3 = 10,
DPAD_UP	11	DPAD_UP = 11,
DPAD_DOWN	12	DPAD_DOWN = 12,
DPAD_LEFT	13	DPAD_LEFT = 13,
DPAD_RIGHT	14	DPAD_RIGHT = 14,
TRIGGER_LEFT	15	TRIGGER_LEFT = 15,
TRIGGER_RIGHT	16	TRIGGER_RIGHT = 16,
JOY_LEFT	17	JOY_LEFT,
JOY_RIGHT	18	JOY_RIGHT,

Fig 6.12 - Buttons correspondence in the code

6.4 Testing & Validation

objects/elements tested	Expected value	Actual value
Input lag test	< 12 milliseconds	9.3086 milliseconds
Game play test (Super Mario Bro)	No perceived lag	No perceived lag
Gamepad tester	All button work	All button work
Case stability	Case holds up under pressure	Case holds up under pressure
Button and joystick cover	Snug, No wobble	Snug, No wobble

fitment test		
Button function test	All 20 buttons working	All 20 buttons working
Program functionality	All feature work	All feature work

Table 8 Testing and Validation

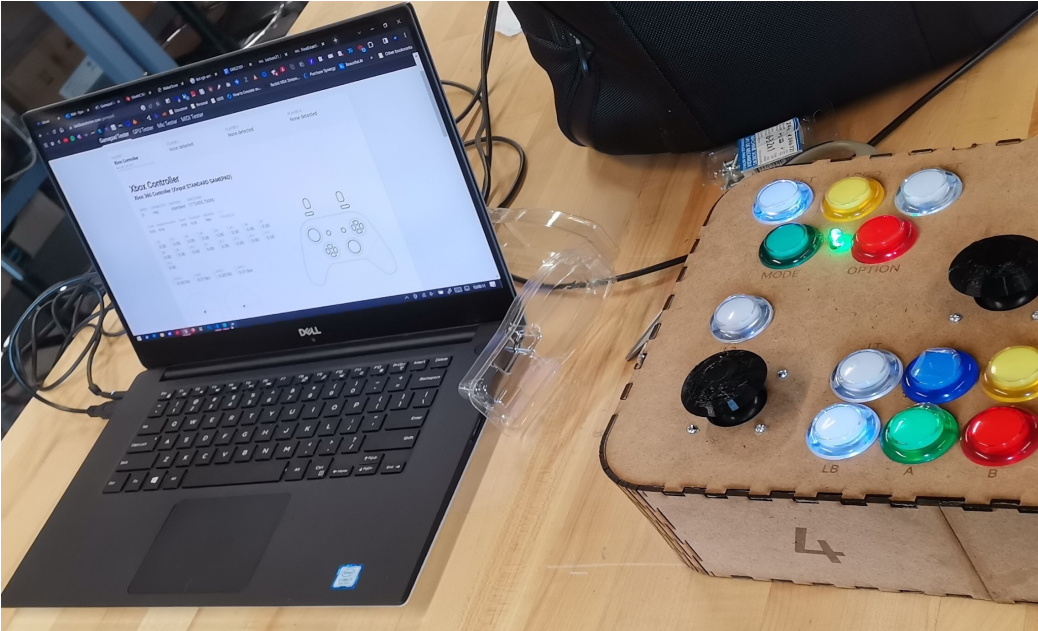


Fig 6.13 - Validation

Input lag test and Button test



Fig 6.14 - Mario Kart

7 Conclusions and Recommendations for Future Work

In order to complete this project, we used conceptual thinking and learned to work as a team to adapt to unforeseen circumstances through the implementation of a project plan and milestones. We also tried to remain motivated throughout the duration of our project in order to achieve a satisfactory end product. Note that our final product always requires some improvements and testing to ensure better results.

There were many lessons learned throughout the design process. We learned to start working as early as possible in order to develop the best product and settle any unexpected problems. Make sure to list and organize tasks in some sort of planning software to keep on track. It is also important to evenly distribute tasks. Good communication among team members is essential and prototyping is a very useful way to communicate with the client to be certain that all of their needs are satisfied.

With additional time, the team would have spent more time on the soldering and completed the PCB. Because of time constraints, we had to replace the PCB with a simple breadboard last minute because it was not working properly. Other future work include the design of a better case and for universal compatibility (PS4/PS5, Nintendo). This would also involve the conception of more suitable mounts for the joysticks as well as the option of single button remapping.

8 Bibliography

- [1] [Statista Canada](#)
- [2] [Enhanced Wired controller Xbox](#)
- [3] [How to emulate an Xbox controller Arduino](#)
- [4] [GIP Controller](#)
- [5] [Joystick dimensions](#)
- [6] [Input test portable](#)
- [7] <https://ablegamers.org/>
- [8] [Getting started with x-input](#)
- [9] [Macro commands and dups](#)
- [10] https://assets.nexperia.com/documents/data-sheet/74HC_HCT166.pdf

APPENDICES

9 APPENDIX I: Design Files

Maker Repo Link : [GameAbility | MakerRepo](#)

YouTube Video: <https://youtu.be/Aom4I5WfPVQ>

Table 9. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
MakerRepo	GameAbility MakerRepo	07/16/2023
Controller.ino Arduino Code	https://github.com/rambolps/Xbox-Adaptive-Arduino-Controller/blob/main/controller/controller.ino	
Axis 1 Pro controller design	Axis 1 Pro	
SNx4HC166 8-Bit Parallel-Load Shift Registers	SNx4HC166 8-Bit Parallel-Load Shift Registers datasheet (Rev. E)	

10 APPENDIX II: Other Appendices

- [A guide to EEPROM | Arduino Documentation](#)
- [▶ How to Add Multiple Inputs to an #Arduino using a Shift Register - The Lea...](#)
- [▶ Is Microsoft... the Good Guy? - Xbox Adaptive Controller](#)
- [How to Emulate an Xbox Controller with Arduino \(XInput\) - Parts Not Included](#)
- [GitHub - dmadison/ArduinoXInput: XInput library for USB capable Arduino boards](#)
- [GitHub - dmadison/ArduinoXInput_AVR: AVR boards package for the ArduinoXInput project](#)
- [Github - rambolps/Xbox-Adaptive-Arduino-Controller](#)