Developing a modular gaming handheld for gamers with muscular dystrophy

Bachelor Thesis Creative Technology by:

Cedric E. S. Omtzigt

Supervisors:

Dr. Ir. Edwin Dertien and Dr. Ir. Robby W. van Delden

Abstract

This thesis describes the continued development of a modular gaming handheld/controller for gamers with Duchenne Muscular Dystrophy (DMD). The prior research specifically went into the design of the controller and creating an initial solution for the PlayStation 4 (PS4) console. However, this implementation only worked for a single game due to the use of a single hard-coded menu for switching between control function groups (CFGs), limiting the player from playing other games with their friends using the special controller. The goal is to develop a system that works with a multitude of games where the player is able to define their desired button scheme per game, and is able to personalize the CFGs per game using an SD card that can be configured on a computer. There are multiple variations of the final pipeline that could be chosen to implement such a controller, but each variation has its own consequences in terms of time to develop, functionality, costs and the resulting volume of the system's pipeline. Based on the research it can be concluded that for the appropriate development for proprietary hardware, such as the PS4, requires the acquirement of the appropriate product specification (Product Spec) so that the engineer knows what to build to using detailed key requirements from the documentation. Reverse engineering is also a possible procedure given that the development stays within its legal bounds. The concept has potential and should be developed further and eventually tested on a broad scale to truly test its effectiveness.

Acknowledgments

I would like to thank the following people for their support through time and materials during this thesis.

A warm thank you to Edwin Dertien and Robby van Delden for their support from within the University of Twente.

Next to that, I would also like to thank Jordi van Heeskwijk from the Sint Maartenskliniek, Laura Koot from Yumen Bionics and Ruud van der Wel from the NMA Gaming foundation for their general support and lending the devices that allowed me to perform the research in this paper.

Table of Contents

List of Acronyms

DMD	Duchenne Muscular Dystrophy
PS	PlayStation
PS4	PlayStation 4
DS4	DualShock 4
HID	Human Interface Device
USB	Universal Serial Bus
JSON	JavaScript Object Notation
CSV	Comma Separated Values
D-pad	Directional pad
CFS	Control Function Switching
CFG	Control Function Group
LED	Light-Emitting Diode
LCD	Liquid Crystal Display
BR	Button Remapping
T4.1	Teensy 4.1
T3.5	Teensy 3.5
IDE	Integrated Development Platform
POV-hat	Point of View hat
PCB	Printed Circuit Board
FPS	Frames Per Second

1. Introduction

The life expectancy of a person with Duchenne Muscular Dystrophy (DMD) lies around 25 years [1], [2]. During this time, motor capabilities progressively deteriorate making it increasingly difficult to perform bodily functions. The decrease in mobility and joint deformations also greatly affect the independence and quality of life [3], [4]. As it becomes more challenging to undertake activities outdoors it may become more comforting to pursue indoor hobbies. As a result, people with muscular dystrophy may also experience trouble with their social environment. An example is connecting with their peers, as they have limited physical activity abilities. In addition, their condition is bound to worsen over time increasing their limitations further. To go beyond their physical limitations, video games are used to connect with communities and have social interactions over the internet. This allows them to converse and interact with others who share the same passion.

Custom-made controllers allow people with muscular dystrophy to continue playing their favorite games. However, the accessibility of these custom controllers is limited and needs experts to configure and tune the controller to the needs of the individual in question. The controller setup is not straightforward either and often requires custom-made solutions that can be adapted on the fly. Hence it is important to find out to what extent a general template can be constructed so that anyone who requires such a controller can more easily configure one.

1.1 Context and Relevance

The experience when gaming is important when the goal is to get enjoyment out of the activity. The client of this project is Marc, who has DMD and needs a custom controller to optimally play all his favorite games. He was featured in the Dutch television program ‘We Gaan Het Maken’ where a team of accomplished experts from different fields of expertise, including Dr. Ir. Edwin Dertien helped create a controller for the PlayStation 4. Hence, the current focus is on creating a working solution for those with muscular disabilities on the PlayStation 4/5. The controller consisted of two nun chucks, each featuring 2 low-resistance buttons that are easier to press compared to standard DualShock 4 buttons, and a used joystick because they are lighter to move around compared to new joysticks. One button would cycle through a menu that arranges different control function groups to the controller. The resulting device was designed to be played with one game, limiting his ability to play other games with his friends due to each game having different control schemes and crucial buttons.

The goal of the project is to design and build an input-intervention unit for custom game controllers where the buttons can be assigned per game more easily with the use of editable profiles. These profiles hold the information on which signal emulates which button, and can be edited, saved, and used with

an as simple as possible mechanism. This has the potential to help people who do not possess full-range motor control to not only switch between games more easily but also allow them to play games that have limited accessibility support and lack remapping features, improving the overall gaming experience.

1.2 Research Questions

The goal is to design a controller for people with muscular dystrophy that includes the ability to cycle through button remapping profiles configured for different games. It is the aim of the study to answer the following main research question:

RQ: What are the characteristics of an effective and practical profile menu and controller, aimed at users with muscular dystrophy?

With the sub-research question:

SubRQ1: What is the state of the art in adaptive game controllers?

2. Background Research

2.1 Literature Review

The literature review aims to find out what information is available on muscular dystrophy and game controllers, and pinpoint what prior knowledge one might need to set one up. The literature review consists of three parts that are equally important to the gaming experience with DMD. The first part will touch upon the accessibility options on current gaming hardware, the second part will explain a set of guidelines and the options for custom controllers, and the third part presents a range of solutions with current assistive technologies for posture and custom inputs. Finally, the literature review ends with a conclusion.

2.1.1 Methods for literature review

The literature presented was found through Google Scholar, Science Direct, and mainly Scopus. The journal articles and conference publications were chosen based on relevancy to the subject matter and RQ1. Keywords for the found literature include: Gaming, Disability, Accessibility, Duchenne Muscular Dystrophy, Muscular Disease, Controller, Custom, Assistive, Technology, Technologies.

2.1.2 Relevance of this research

The issue of accessibility in gaming is ever-so-relevant because there could be more support shown from the bigger corporations. The target group may be small and not net the profits compared to the regular gaming target audience, but it is a group that should not be forgotten because this group depends on the ability to play video games. For them, gaming is their socializing because it is difficult for those kids and young adults to go outside. Therefore it is important to bring this issue to light and improve the general awareness around not only accessibility in gaming, but also in general.

2.1.3 Accessibility Options

A key detail that comes forward through existing literature is that between DMD and those with full-motor capabilities, the difference in performance for motor capabilities increases when more fingers are involved [5]. A standard Sony PlayStation DUALSHOCK™ 4 controller offers 17 buttons, a touchpad,

and 2 joysticks each with 360 degrees of directional control [6]. If a person is unable to reach a button on the controller with a crucial function, it is common to remap the button with one that is reachable but has a function that can be sacrificed. Unfortunately, this can often only be done for one custom profile across the entire console, limiting the number of games that are accessible with the remap.

According to a survey among disabled gamers conducted by Beeston *et al.* [7], key remapping is the most popular accessibility option along with subtitles, followed by text enlargement and contrast or color changes. As for hardware, the most popular choice was 'other' where respondents elaborated on various situations that were not on the list such as using a converter to play with a keyboard and mouse on a console, or an on-screen keyboard. The second most popular but most voted single technology was the use of custom controllers, which brings us to our next point.

2.1.4 Custom Controllers

Gaming controllers and input devices often have to be adapted by experts to the wants and needs of disabled gamers. A set of guidelines that resulted from an analysis of gaming devices for people with disabilities in the upper limbs with special attention to limited finger dexterity by [8] indicated the following:

- The ability to use joysticks is a must;
- The resulting controller should be able to lay flat on a surface while being held;
- Buttons should be able to be remapped;
- The controller should function as a standard Human Interface Device (HID) class USB component.

The HID class is a communication protocol that configures application-specific behavior to a USB device, which can be seen in depth in [9].

Secondly, Iacopetti *et al.* [10] argue that the adapted game controllers are often modifications made to a commercialized controller in a "do-it-yourself fashion." The fact that these customizations do not meet the necessary safety certifications and are built on existing products means they cannot be commercialized on a big scale. Because of this, the adapted controllers are only used by a niche group of people and do not reach the entirety of the target audience that would also benefit from the solution [10].

Finally, there is an apparent trend in the literature about the creation of an adapted game controller for disabilities in upper limbs. As Hassan *et al.* [11] describe, it is common for there to be a central interface device into which either standard controllers, or custom sensors, joysticks, switches, and buttons are connected [10], [12], [13]. These custom inputs allow for a personal gaming experience because they can be tailored to the abilities of the person, think of the positioning of inputs and force needed for said inputs. The Xbox Adaptive Controller made by Microsoft, shown in Figure 1, seems to be the only commercialized system that allows custom input with a 3.5mm jack for each function on the classic Xbox One controller [14].

Figure 1: Photo showing the Xbox Adaptive Controller in use (white box with 2 black circles)

2.1.5 Assistive Technologies

There is a range of 'preventative' and 'reactive' solutions to the problems gamers with DMD face. Preventative solutions act as an intervention that aims to decrease the rate at which boys with DMD degrade in strength and range of motion. This is preferably done early on to support the functionality of the limbs in the long term. Active elbow support as an exoskeleton is a common intervention being tested for boys with DMD. Examples of such a device are the dynamic arm support developed by Heutinck *et al.* [15], ACTIVE-seated by Lowes *et al.* [16], and The Yumen Arm by Janssen *et al.* [4], which developed similar devices and used similar testing methods to test the statistical significance. All indicated median to great 'improved elbow range of motion', 'extension strength', and 'less fatigue' through training with their device. An exoskeleton can reduce the effect of gravity acting on the muscles and can relieve the stress on them, to aid in daily activities and not only gaming [17].

Secondly, reactive solutions are those that aim to give back the lost function through more direct kinds of assistive technologies. Consoles and controllers between different companies each have different communication protocols. Thus, a special device is required to use actuators and sensors, and transmit the correct byte sequence for the input to be registered. This is done with a specialized Universal HID Actuator that makes the console think it is receiving signals from a native controller [18]. A native controller is needed next to the preferred controller for authentication and specific features for each console [19]. This allows for a plethora of different sensors and actuators to be configured by an expert, who can tailor the setup to the person with DMD's needs and abilities. Examples of such devices currently on the market are the Titan One [20] and Titan Two [19]. These devices allow official wired controllers, unofficial wired and Bluetooth controllers, keyboard and mouse, and sensors and actuators to be used. The latter combination requires an extra device that

collects the input of the different sensors and transmits it through USB. An alternative is to use the Xbox Adaptive Controller and have the different buttons and joysticks wired into it using the 3.5mm jack connections to use on a Nintendo Switch, or PlayStation device.

2.1.6 Conclusion

This literature review aimed to discover what technology is out there to aid Duchenne Muscular Dystrophy gamers. The prior knowledge needed for the desired setup has several factors to be deemed operational and successful: accessibility options, custom controller, and assistive technologies.

First, accessibility options are important for allowing people to play the games with the assistance they need, and with button remapping being the most wanted feature it is mostly limited to one button remap profile across the entire game system. If a player wanted different controls between games they would first have to edit the button remapping profile in the system's menu.

Second, the custom game controllers that are available are divided into two groups, for which one is a do-it-yourself area with custom solutions by local experts and the other is the concept of a hub that takes custom inputs, for which Xbox has the only commercialized version for their current generation consoles.

Finally, assistive technologies range from solutions that will aid the well-being of the body in the long term and ones that improve the gaming experience directly by allowing a multitude of controller inputs and devices using specialized HID USB components. The key takeaway in this section is the ability to use the Xbox Adaptive Controller using the HID USB component like the Titan One or Two on a console like the PlayStation 4 and 5. The active elbow supports and exoskeletons that have been tested allow the gamer to maintain or preserve a healthy posture and its integrity in the long run. A combination of the preventative and reactive assistive technologies discussed in 2.1.5 would allow the gamer with DMD to play on multiple consoles with the same controller setup too, which is an improvement to using the classic controller for each console with different inputs, weights, and form factor.

2.1.7 Discussion

The literature review provides an examination into the most sought-after custom controller requirements and accessibility options in gaming, in addition to the existing assistive technologies that have been developed with improving the quality of life in mind. This information can all be applied in the context of the research project because it solidifies the claim for a need of such a controller to exist for each existing gaming console. When creating the custom controller it should be noted that the button

remapping is one of the most important features according to the survey, and the functionality thereof could be a determining factor for whether or not a person would want to try the controller. The process should be made as simple as possible where the user would have to do as little as possible, in order to make it feel as if the button remapping per game offers better functionality than current systems do. This would require the creation of a software on the computer that would be able to alter the button remapping profiles per game, and load them back into the device instead of having the user use low level code like C++, or specific formatting conventions like JavaScript Object Notation (JSON) or 'comma separated values'(CSV) formats use.

In addition, there should be more research done into the ergonomics of weight distribution of game controllers. A common issue for gaming with DMD is that the controller becomes too heavy to hold, which is why not only the button placement is a problem that should be noted. The reason why current console manufacturers like PS and Microsoft do not overhaul their controllers for better accessibility or ergonomics may be due to various reasons, but is most definitely also linked to brand recognition. It is impossible to please an entire population, hence why it is important to offer options. Modular controllers would offer such options, where there is a difference between models where the handles are different sizes, or one has more buttons than the other to suit the needs of as many gamers as possible.

Finally, it would be interesting to see how the assistive technologies that aim to improve the longevity of the ability to maintain a good posture like the Yumen Arm could be combined with the special controller, and the benefits or effects it could have with long term usage. Further research could be done on whether the combination could be tested and how the effectiveness could be evaluated.

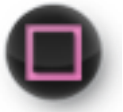
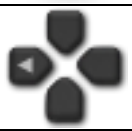
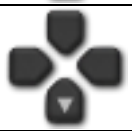
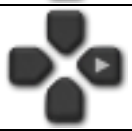
3. Development of Prototype / Design / Method

3.1 Concept

3.1.1 Initial Project Notes

The reason behind this project is to build upon the existing work of Dr. Ir. E.C. Dertien and GitHub user lonnwolf031 in the hopes of decreasing the data pipeline of their solution, or creating a more accessible and feasible alternative of a custom gaming controller for the PS4. Their solution consisted of a dismantled DS4 controller where connections were soldered to the existing buttons and takes inputs from a 3D printed split handheld system using an Arduino Due, hence the name Splitcontroller. User lonnwolf031 has continued the project after edwindertien's last commit and is at this time 30 commits ahead, therefore references will be made to the latest forked [21] version instead of the original [22].

A standard DS4 controller consists of 18 buttons, of which 2 are analog triggers, and 2 analog joysticks with each an X and Y direction that can be seen as individual potentiometers per axis. The layout of the DS4 controller has been depicted in Table 1.

Reference Number	Icon	Name	Signal Description
1		Square	Digital signal, button
2		Cross	Digital signal, button
3		Circle	Digital signal, button
4		Triangle	Digital signal, button
5		Left D-pad	Digital signal, button
6		Down D-pad	Digital signal, button
7		Right D-pad	Digital signal, button

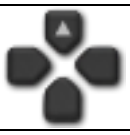
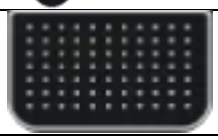
8		Up D-pad	Digital signal, button
9		L1	Digital signal, button/bumper
10		R1	Digital signal, button/bumper
11		L2	Digital signal, button Analog signal, trigger
12		R2	Digital signal, button Analog signal, trigger
13		L3	Digital signal, button when the left joystick is pressed down
14		R3	Digital signal, button when the right joystick is pressed down
15		SHARE	Digital signal, button
16		HOME	Digital signal, button
17		OPTIONS	Digital signal, button
18		Touchpad	Digital signal, button
19		Left stick	Analog signal, joystick
20		Right stick	Analog signal, joystick

Table 1, List of standard DS4 buttons with name and signal type

Icons of numbers 5 through 8, and 13 and 14 that represent the Directional pad and L3 + R3 images originate from the psdevwiki for the DualShock 3 [23]. The Touchpad icon 18 originates from Dominik Geuer's controller icon pack [24]. The remaining button icons originate from the psdevwiki for the DualShock 4 [25].

There are two important aspects to this project that may sound similar but should not be mixed up given that each one serves a different purpose. The terms are Control Function Switching (CFS) and Button Remapping (BR).

3.1.2 Control Function Switching

One button on the controller will cycle through a menu that arranges different control function groups for the custom controller as illustrated in Figure 2. In this scenario button, F has been pressed to change the assigned functions of the buttons. The idea is to not only use the control function groups on buttons but also the joystick axis where if you would hold the joystick forward on the Y-axis a button could be pressed. Another important feature is to implement combination buttons with a single button press. Combination buttons can be hard to press when the buttons are on the same hand for gamers with DMD but are a prominent feature in gaming. FIFA 21 is an example of a game on the PS4 where combination buttons are commonly used to perform special moves, but their accessibility options [26] do not allow users to map combination buttons to a single button. One attacking move called the 'Super Cancel' even requires all bumpers and triggers, L1 + R1 + L2 + R2 to be pressed simultaneously [27].

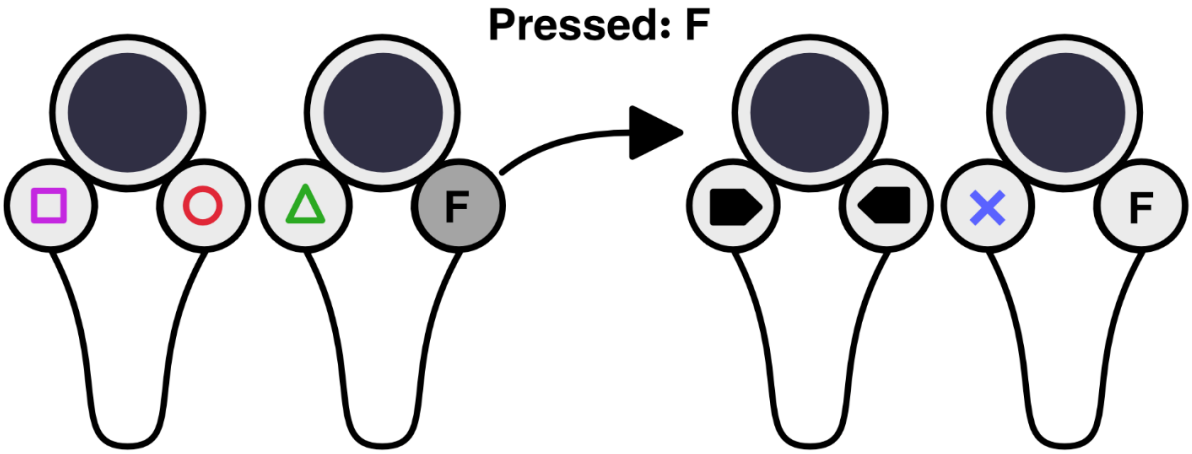


Figure 2, Control function group switching scenario

The reason behind CFS is to provide the same level of functionality that a standard DS4 controller has to offer but in a lighter and easier-to-use package, when it comes to controller weight, the distance between buttons and the placement thereof, and the force needed to press those buttons.

3.1.3 Button Remapping

If a person is unable to reach a button on the controller with a crucial function or simply dislikes the standard control scheme, it is common to remap the button with one that is reachable but has a function that can be sacrificed. At this time most games on the PS4 do not allow the user to remap the buttons. Instead, a singular system-wide remapping opportunity is given in the PS4's system settings in the Accessibility section as shown in Figure 3. This section can be accessed through the PS4's home screen and going into *Settings* > *Accessibility* > *Button Assignments* > **click checkbox* Enable Custom Button Assignments* > *Customize Button Assignments*.

Figure 3, PS4 Single use system wide button remapping menu

This is detrimental to those who prefer or even require a different control scheme from the standard one across multiple games. This is why it is important in terms of accessibility to offer the ability to have a button remapping scheme for multiple games instead of a single system-wide change.

3.2 Overview of available techniques

3.2.1 Controller converter

The custom controller will need a device that understands the input it is receiving and outputs this in such a way that the receiving PS4 knows what to do with this data and uses it properly. In addition, due to the proprietary nature of the PS ecosystem, a controller converter has to be used to circumvent the 'security handshake' that is necessary for the controller inputs to be recognized by official PS systems. The systems that will be used for this project are the Titan Two, Figure 4, and the CronusMAX PLUS, Figure 5.

Figure 4, Titan Two controller converter [40]

Figure 5, CronusMAX PLUS controller converter [41]

These systems can cost up to 100 euros but are scarcely available and may take several months to arrive. For this project, a Titan Two was lent out by Maartenskliniek and a Cronus Max by NMA Gaming for research and development of the custom controller.

To create the security circumventing technology yourself would be an enormous task, one that was too great to tackle within the projected timeframe. Extensive knowledge about reverse engineering and USB protocols will probably be required so it may be a better decision to simply purchase one of these devices and wait for them to arrive. More on the security element that can be found in licensed third-party controllers can be found on GitHub, where user dogtopus created somewhat detailed documentation [28].

3.2.2 Microcontrollers

The input from the controller will have to be processed before it can be sent through to the PS4 with the controller converter. Microcontrollers are a great way to handle specific outputs after being programmed, based on certain inputs. Requirements that will have to be met are:

- The ability for CFS
- Loading control function profiles from SD cards

- Handling multi-button output
- Allowing the implementation to stay compact
- Visual feedback of actions

The Arduino series is a popular microcontroller for hobbyists that allows for the basic functionality of inputs and outputs. The microcontroller that was chosen for this project was the Teensy 4.1 because it offers numerous advantages over the standard Arduino lineup.

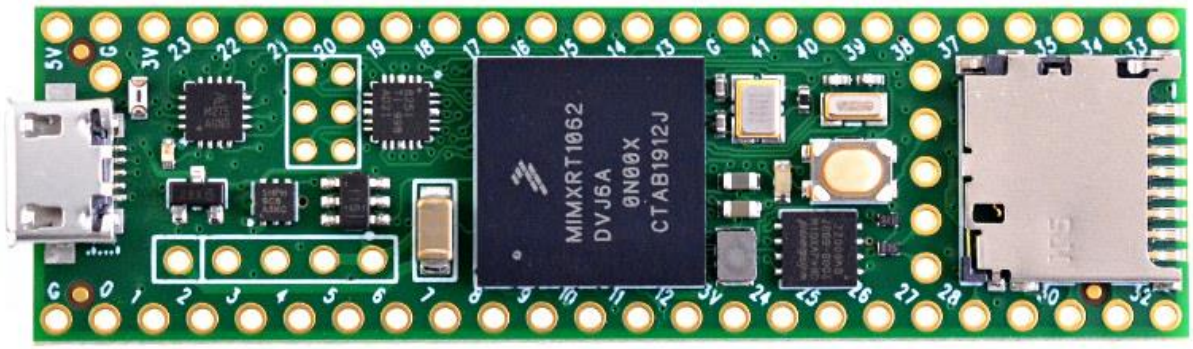


Figure 6, Teensy 4.1 microcontroller [29]

Official Arduino boards do not feature the required SD card slot and would need a shield to extend the functionality of the board, whereas the T4.1 shown in Figure 6 features one by default. Other advantages of the T4.1 for this project include:

1. ARM Cortex-M7 600Mhz CPU
2. 42 easily accessible I/O pins, 55 in total
3. Act as a USB Device through the main USB port (micro USB to USB A)
4. Act as USB Host allowing USB devices to be connected to the T4.1
5. Dimensions of 61.66mm long and a width of 17.78mm(+- 0.6mm)
6. Native SD card port (4bit SDIO)

The T4.1 will offer more than enough processing power over comparable microcontrollers [29, 30].

The ability the make the T4.1 function as a USB device also allows it to act as a Flight stick, Joystick, Controller, Mouse, or Keyboard through its dedicated libraries. The signal that the T4.1 then outputs could, in theory, be interpreted by the controller converter because of it acting as a standard HID. If that would not work there is also the possibility to transmit Raw HID which allows for 64-byte messages to be communicated. The T4.1 would then be configured as a USB controller of its own rather than mimicking an HID device through one of the device libraries.

3.2.3 Configuration options

The custom controller is built with 2 buttons and a joystick per controller, of which each hand holds one. This input would then go into a T4.1 microcontroller with an SD card that can load controller configurations onto the board. From there on there are a variety of options with each their negatives and positives. Because the controller and microcontroller setup do not change, they will be represented within a dotted box. Here follow the configuration options:

Variation 1.

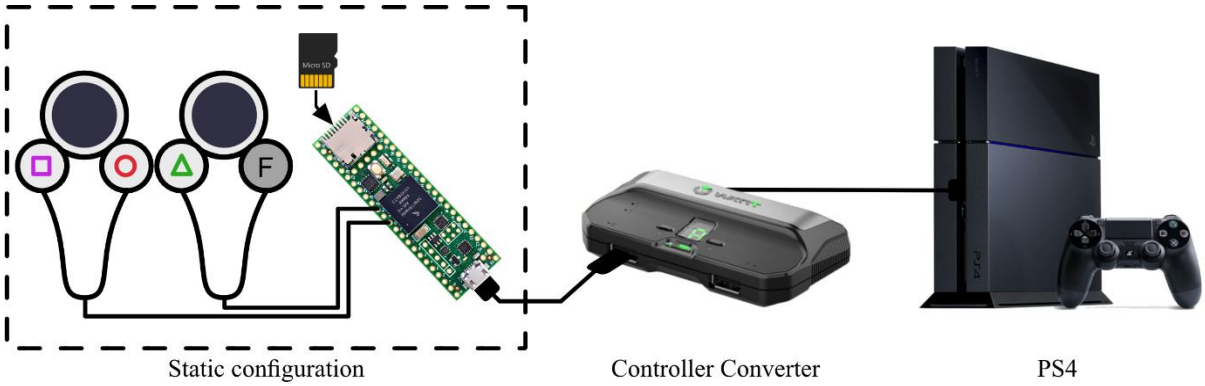


Figure 7, Controller converter directly connected to the PlayStation 4

Variation 2.

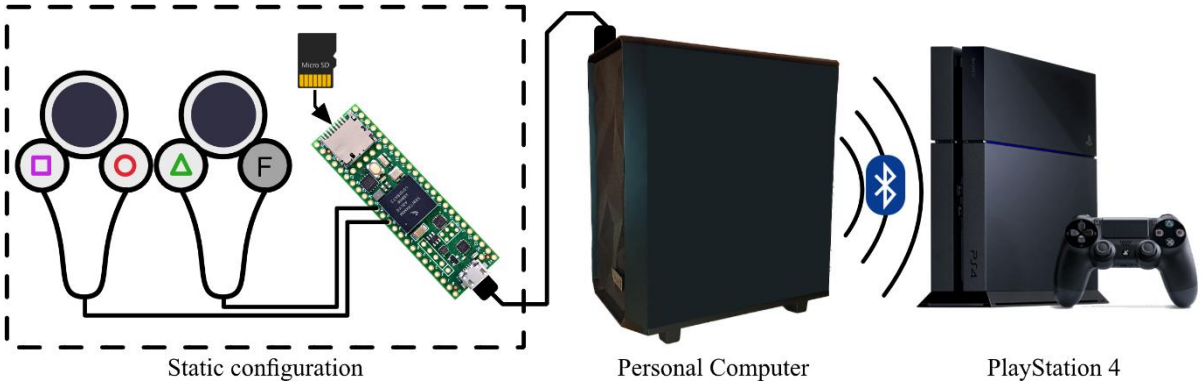


Figure 8, Personal Computer used to send signals to the ps4 with Bluetooth

Variation 3.

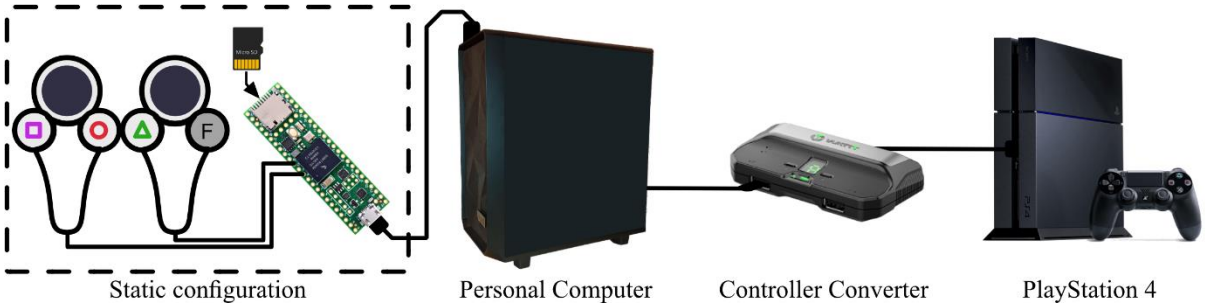


Figure 9, The personal computer is used in between the microcontroller and controller converter

Variation 4.

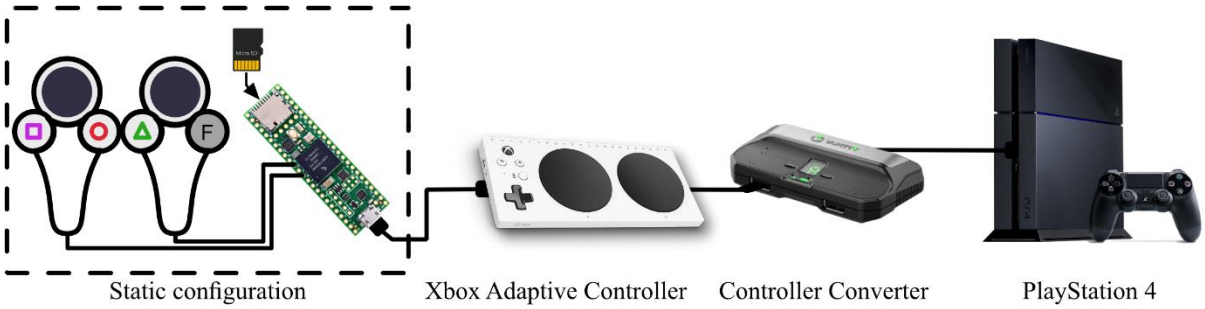


Figure 10, The Xbox adaptive controller is used to provide signals that are known to work to the controller converter

Variation 5.

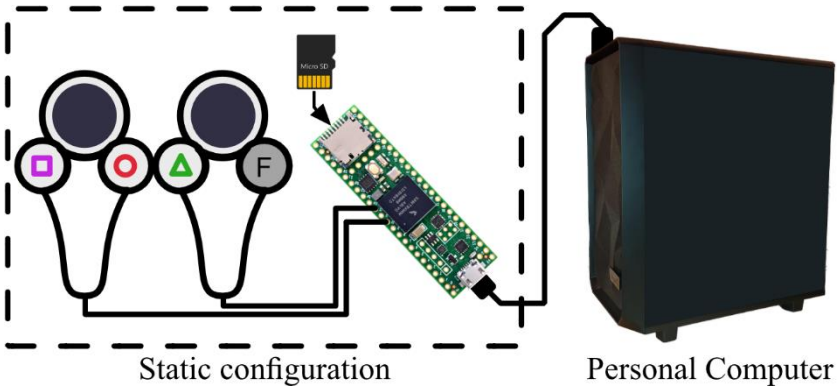


Figure 11, The solution made to work on a personal computer instead of a PS4

Variation 6.

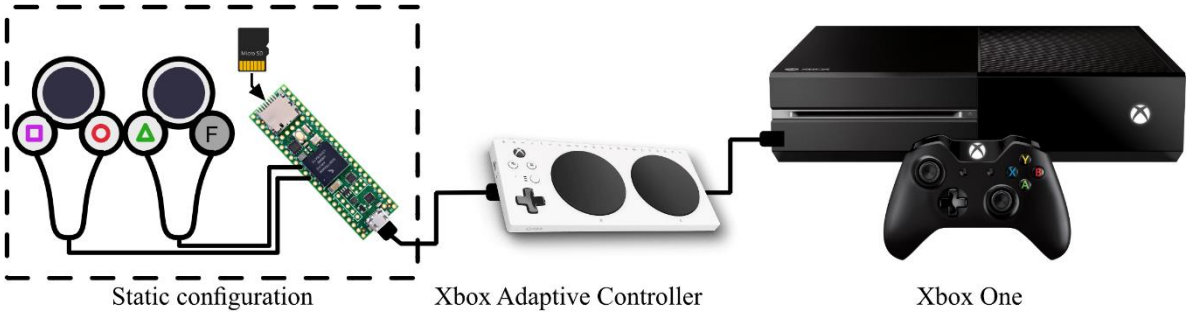


Figure 12, the solution made to work on an Xbox One instead of a PS4

The ideal configuration would be Figure 7, variation 1, where the microcontroller with joystick combination inserts into a controller converter and is usable with the PS4. If the microcontroller would be unable to provide the right connection to the controller converter, the idea of connecting through a computer may have to be explored where there is also the option for Bluetooth capabilities. Next to that, the controller converters are known to work with Xbox products and thus it may also be a viable option to connect the microcontroller to the 3.5mm jack inputs of the Xbox adaptive controller, which would then provide input that is understood by the controller converter. This solution would not reduce the pipeline and bring a lot of extra costs compared to the existing solution, given that each button will consist of its own wire with a 3.5mm jack and the joysticks would each consist of 4 wires, for power, x-axis, y-axis, and the button within the joystick. Therefore, if the solution for PS may not work it is worth the effort to attempt a solution that works on PC, for general proof of concept and working out the details and requirements of the final solution.

3.3 Requirements

The requirements the product should fulfill include various functions, both in the real world and digital world. For the real world, there should be visual feedback on what menu is being used, what profile is loaded, and the ability to determine if the controller is working or not. As for digital aspects, the controller should feature the controller, microcontroller, and receiving system which is a PS4 in this case. This has been visually presented in Figure 13.

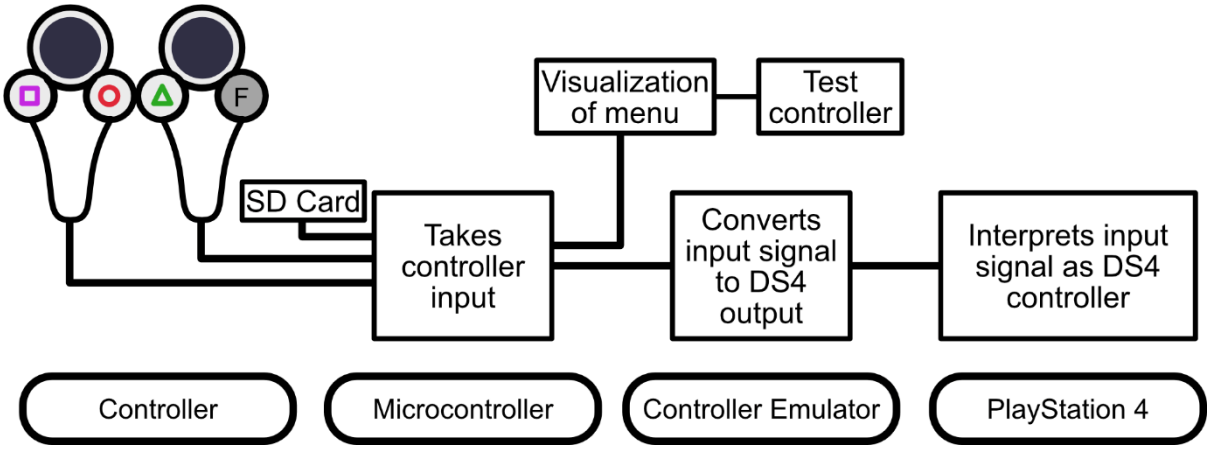


Figure 13, Design requirements mapped

3.3.1 System pipeline

The controller consists of two separate parts that are each held in a hand. The goal of the controller is to transmit real-world input to the microcontroller for further processing.

The microcontroller processes the input from the controller, for which the data is then manipulated through programming to provide the desired output to the controller emulator. The microcontroller also loads the button combinations per CFG that have been specified per game. Output also provides visualization of the menu and type of menu that was loaded through an LCD and LED rings. Visual feedback is important to make sure that the user is aware of this transfer taking place, providing feedback in the form of color rather than solely plain text on whether the controller is working and most importantly what CFG they are currently in.

The controller emulator will process the byte sequence that the microcontroller produces when buttons are pressed or joysticks are used and then output the corrected signal to the PS4, which will interpret the signal as if it came from an official DS4 controller.

3.3.2 Functional requirements

There is a certain level of functionality that the resulting setup must fulfill to be deemed an improvement over the Splitcontroller. These functional requirements are as follows:

The controller must be able to -

1. Take button input from the user and accurately reflect this in the digital domain, with minimal delay.

2. Read CFG and button remapping profiles per game from an SD card. This will make it easier for people using the end product to change the profile they are using, compared to hard-coding the menus into the microcontroller.
3. Be easy to set up and have the ability to be mounted on spaces, like wheelchairs where there is minimal setup possibility.
4. Be built in a modular fashion, so that in the case of repair the modules can be swapped out rather than having to throw away an entire controller.
5. Display what function group the user is in with a LED ring or matrix, with a color of their choice per CFG. There should also be an LCD present with more detailed information such as what button was pressed, what menu item they are currently in, and what game profile they currently have selected.

3.4 Hardware

The hardware used for the project includes the following:

3.4.1 Microcontrollers:

The datasheets of the T4.1 and T3.5 can be found in Appendix C.

Note: If you are planning on using libraries with the Arduino software, make sure to check the supported boards first. For example, the T4.1 does not work with the XInput library that will be mentioned in section 3.5, but the T3.5 and TLC are supported. The same goes for the USBHost_t36 where the T3.5 and TLC are not supported but the T4.1 is.

3.4.2 Other electrical components

Computer:

Desktop computer with at least 2 USB 2.0+ ports

OS: Microsoft Windows 10 Pro

Version: 10.0.19044 Build 19044

Monitor:

Feature HDMI output for the PS4 and either an additional HDMI slot or a DisplayPort slot for the PC

PlayStation 4

Software version 9.60, used up until 29/06/2022.

DualShock 4 controller:

One that is functioning accordingly, with no stick drift or other malfunctions.

HID Controller:

A standard controller that can be used on the PC. Should be able to be plugged into the T4.1's USB A to USB A connector.

In this case the Logitech Precision Gamepad Blue, Figure 14, which does not have joysticks, and the Nacon GC-100XF Black, Figure 15, which does have joysticks, and features a DirectInput and XInput switch, which are two different communication methods for the HID controller. They essentially do the same thing but the XInput implementation is more recent and up to date than the DirectInput API. More on this can be found on the official Microsoft documentation site [31].

Figure 14, Logitech Precision HID controller

Figure 15, Nacon GC-100XF HID controller

Controller Converters:

Had to switch controller converters throughout the project due to the lending period of the Titan Two ending halfway through. The CCs work with the same principle and did not behave differently from one another when tested. The devices used were the Titan Two, and after that device had to be returned a CronusMAX PLUS was acquired.

Basic electronic components:

Appropriate wires (capable of transferring data)

Breadboard

4x 10k Ohm resistors

4 Tactile pushbutton switches

2x Joysticks

Some USB A to Micro USB wires may not transfer data and are made for powering devices only. Rule this out before trying to test the system.

3.5 Software

3.5.1 Portable Arduino IDE

Arduino IDE Version used: 1.8.19

The Arduino IDE is an open-source IDE that allows the user the write, compile, and upload code for Arduino boards. The portable installation [32] allows you to keep several instances of the program which is beneficial when you are trying to install things that may go wrong. In this case, there were 3 instances made. One for Teensyduino, one clean install without any additional installations, and one for XInput with the Teensy-XInput patcher which is all later explained in section 3.5.3. Under download options on the Arduino IDE 1.8.19 Installation page [33], choose the ZIP file rather than the installation program for Win 7 and newer. It is advised to create an Arduino folder directly in the C:/ drive for easy access and to keep a short PATH. Other than that you should simply follow the installation instructions using Arduino's documentation [32].

3.5.2 Teensyduino

Teensyduino version used: 1.56

To run sketches on a Teensy microcontroller you need to install Teensyduino to your Arduino IDE [34]. The installation process [35] is straightforward, you only need to know where your portable Arduino IDE is installed to direct the installation to that path, shown in Figure 17. There is also a prompt to choose libraries to install. It is easiest to install all of them in case you want to try one that you did not plan on using when installing.

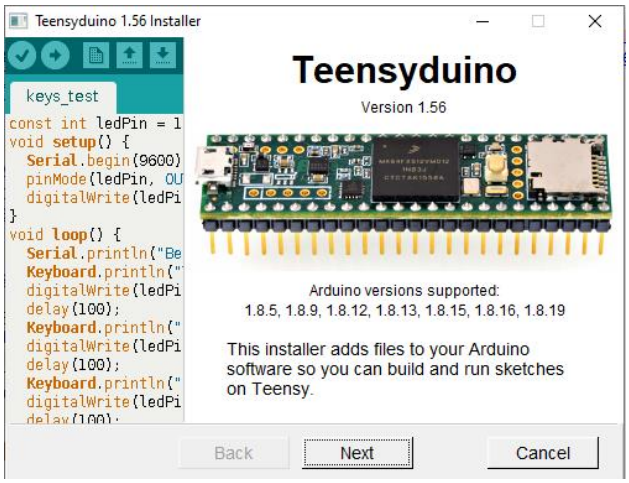


Figure 16, Teensyduino installer welcome screen

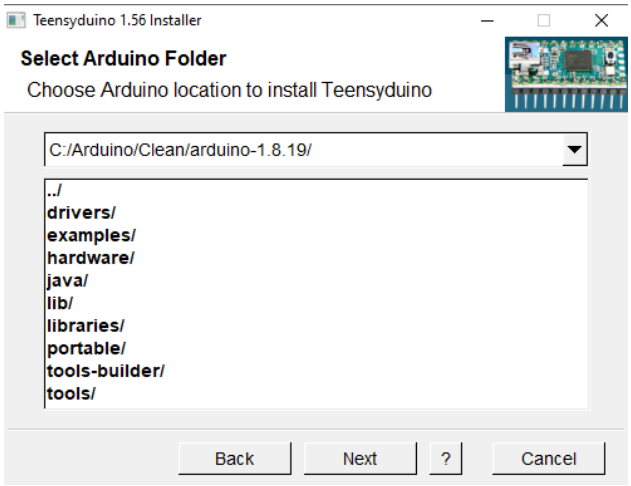


Figure 17, opening the Arduino installation folder for teensyduino to install itself

3.5.3 Libraries used

1. PJRC's Joystick library [36] build in with Teensyduino. Version number unknown (Date 30/06/2022)

This library ended up not working fully for the PS4 which is later explained in section 3.6. The examples that come with the library are a great help in understanding the functionality.

2. ArduinoXInput Version 1.2.5 ZIP through GitHub [37] or library install through Arduino IDE.
    a. Copy the files from the “teensy” repository in ArduinoXInput_Teensy [38] into the existing “hardware” folder of the Arduino IDE and replace any existing files.

This library worked fully for the PC.

Select the supported board of your choosing. In this case, it is the T3.5. Under USB Type select XInput.

***BEWARE: Once you upload to the T3.5 as an XInput device, Serial communication will become unavailable and the device will no longer be recognized on the ports. If you want to read out the data through the Serial Monitor you have to select a different USB type, like Serial. You need to know how to reset your board before attempting to upload code or change the board type. What worked for the T3.5 is to either:

- While connected to the PC, quickly give the reset button one tap
- While connected to the PC, hold it for 5, 10, or 15 seconds a few times
- Unplug the board, hold the reset button, plug it back into the PC and then let go of the reset button.

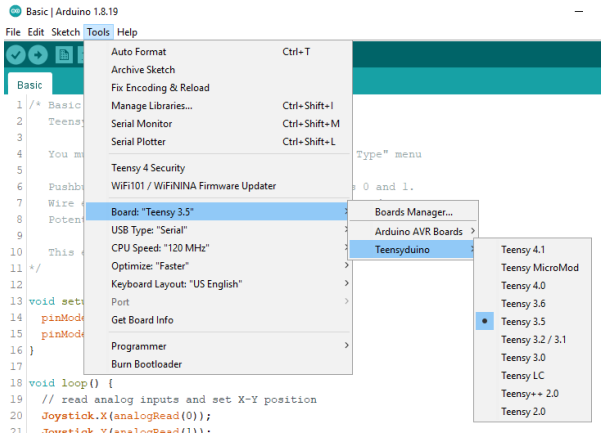


Figure 18, Choosing the board

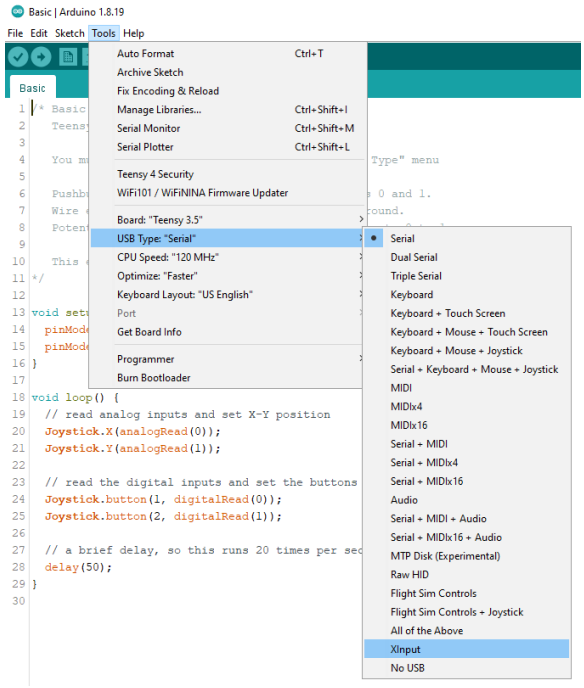


Figure 19, Selecting the XInput USB Type

PJRC's own Joystick library [36] for the T4.1 that comes with the Teensyduino installation was used in an attempt to make the controller function correctly on the controller converter. The library has a clear overview of functions and is well enough documented for basic implementation. Using this library and plugging the T4.1 into the CC with code that emulated 4 buttons and 2 joysticks showed that the T4.1 that is now a USB device was accepted by the CC with a green light, but input was not understood correctly. More details will be explained in 3.6.

3. USBHost_t36

This library is included in the Teensyduino installation if it was checked when going through the installation process, and was used in combination with the T4.1 with USB A header to USB A female connector. The specific focus is on analyzing the communication protocol and the bytes that are sent or changed with each button press or joystick movement of the HID controller.

3.6 Design process

The goal of the design is to simplify the Splitcontroller pipeline and make it function just as well, with support for CFGs and button remapping loaded through an SD card. The major issue is that the problem is exceptionally unique like David and Goliath, it is the designer and Sony. Therefore the focus at this stage is on creating a working implementation of the special controller for the PS4 rather than trying to redesign and user test the Splitcontroller again.

The first step was to find out what existing technologies could be used as a potential solution.

3.6.1 Joystick Library

The Joystick library from PJRC for the T4.1 looked promising and was investigated. After having coded a simple setup with a test for 4 buttons and two joysticks with full axis support, the controller was tested using the joy.cpl program in windows 10. In this program every button and axis responded correctly and as expected. When plugging the controller into the CC, the Titan Two indicated that it supported the device with a green LED. However, the signal that it produced for the PS4 was far from expected. It is likely but not certain that the T4.1 made itself recognizable as a standard HID that the Titan Two could recognize but proceeded to send byte sequences that had different functions linked to them

compared to the byte sequences that the Titan Two expected, or sequences that the titan two was unfamiliar with and tried to interpret in its own way.

In the incredibly lacking documentation [36] it is mentioned that the POV-hat, also known as D-pad, is controlled with degrees instead of numbered buttons, where 0 is up, 90 is right, 180 is down and 270 is left. When using the D-Pad you are meant to reset the direction of the button press by using *Joystick.hat(-1);*, otherwise the theoretical controller will not know that you let go of the button. What *Joystick.hat(-1);* does on the PS4 through the CC is send the signal of 'left' on the D-Pad, just like the degree that is supposed to represent the left button *Joystick.hat(270);*, without ever stopping. *Joystick.hat(90);* would send the D-Pad signal 'right' continuously with no option to turn it off and go back to the default state.

In addition, the left and right joysticks failed to be interpreted by the CC. This may have something to do with the way the joystick values are sent but it is currently unclear as this issue would need thorough investigation on its own. The logic for the Joystick library can be found in the Arduino folder where Teensyduino has been installed into an Arduino installation at: *Arduino-1.8.19 > hardware > teensy > avr > cores > teensy 4 > usb_Joystick.h*

The definitions that help set the other USB definitions in the other files for the joystick can be found in the same folder in *usb_desc.h* with the bytes that make up the descriptor listed in *usb_desc.c* line 281 for standard controllers and line 327 for large controllers, if this was defined in *usb_desc.h* by editing JOYSTICK_SIZE to 64. Kenton Hamaluik tried to use the standard Joystick library for a Unity project, but Unity would not recognize the controller. He decided to create his own HID joystick by altering the code in the previously mentioned locations for which he left adequate documentation [39].

What did end up working was the 4 buttons, Square – Cross – Circle, and Triangle. This is because they are simply linked to a button number 1 – 2 – 3 and 4 where the interpretation is universally understood between the systems. Even if all the buttons worked it is impossible to play mainstream games without the joysticks or d-pad functioning and therefore a different solution had to be explored before deep diving into writing a custom HID joystick.

3.6.2 XInput Library

The XInput library looked promising, with online sources stating they were able to create controllers that worked on the PC which indicated that their protocols may also be understood by a CC. Youtube user noycebru showcased a working prototype [40] and finished product [41] that worked flawlessly on a PC. Next to that it was also incredibly well documented [42] and accompanied by an extensive tutorial [43], which is why it was worth checking out the solution on PS4.

The supported board's list was what delayed the project, as the T4.1 was too new to be supported, and a TLC, T3.5, or T3.6 was not in possession. The TLC was the only board that was able to be acquired directly but did not feature an SD card slot, so the implementation on it would only have been for a proof of concept. Luckily a T3.5 was soon acquired and the code was already written for it to be tested.

The joy.cpl program showed that the buttons and joysticks worked as expected and indicated that the controller was recognized as an Xbox controller for PC, rather than just an Xbox controller which raised some suspicion. Upon plugging in the T3.5 into the CC as an XInput device, the light turned red on the Titan Two, indicating that it does not understand what kind of device is plugged in and will not try to interpret any of the inputs. Thus, this solution was also not destined for the PS4 implementation but the functionality and level of documentation make it a better library for developing a controller for the PC than the standard joystick library from PJRC.

3.6.3 HID Analysis

Provided that the two best options for libraries were specifically designed for PC, and not necessarily emulating an HID device, there is no other option other than creating a working HID controller by analyzing existing HID controllers and mimicking their communication protocols. These controllers will first have to be confirmed to work with the CC before they can be deemed useful to the efforts of this project. It should be noted that the timeframe of the project in its current state only allows for the setup of the analysis and not the reproduction of the communication. Some of the analyzed HID output can be found in Appendix D.

The setup includes the use of the USBHost_t36 library, using the HIDDeviceInfo example sketch with the T4.1 USB Type configured as a Serial device. This setup allows for a standard HID controller to connect and intercept the communication that would normally go directly to the PC. There were 2 HID controllers in Figures 14 and 15 that were able to be used with the CC, of which the one in Figure 14 did not have joysticks, and the other in Figure 15 that did have joysticks spammed the Serial port with a value that continuously changed, in such a way that the raw output data became unreadable. For the second controller with joysticks to feature useful output, the example code would have to be edited to either filter out the byte sequence of the spammed value or comment out/delete the part that is continuously printing the value.

The output of the HID controller with joysticks would produce Serial data like in Figure 20 when the RAW HID output was enabled. The value consistently switches between 512 and 513, which is around the halfway point of 1023. However, neither the joysticks nor D-Pad represented this 10-bit value making it unclear what did cause this continuous switching.

```
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=512
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=513
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=512
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=513
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=512
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=513
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=512
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=513
HID(10005): 00 00 0F 80 80 80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=FF00002C, value=512
*** Device HID1 - disconnected ***
*** HID Device hdc1 - disconnected ***
```

Figure 20, RAW HID Serial output of HID controller with joysticks

A button press was able to be recorded when quickly pressing and releasing the button in Figure 20, followed by disconnecting the device to prevent the Serial monitor from continuing to spam over the data entry.

The fully ordered output was clearer but lacked significant detail, only displaying the values and not the RAW HID byte sequences.

One unfortunate detail is that this simple setup used for the standard HID controllers does not work for either a PS4 or Xbox controller as they are not activated upon connection and will try to connect to their system over Bluetooth if their home button is pressed. The only kind of output that was able to be found is from Frank who hosted it on his wiki for some time but is now a broken link. Luckily it is still accessible through the Internet Archive's WayBackMachine [44] which is a service that captures web pages for future use without having to worry about the service disappearing. The DS4 HID RAW USB output [45] is now able to be analyzed and can be attempted to be reproduced for research and education purposes. GitHub user johndrinkwater also did some research into the DS4 and posted his RAW HID USB output in the form of a hex dump, where each byte is a two-digit hexadecimal number that is posted in rows [46]. The majority of the information that is shared between johndrinkwater and Frank's findings can be found on the PS4 dev wiki's DS4-USB page [25].

What did work was hooking up the DS4 controller to the PC and analyzing what buttons and axis the controller used according to Windows 10's joy.cpl program, which resulted in the following: the numbering of the buttons could be useful, especially the home button as that is quite an important one.

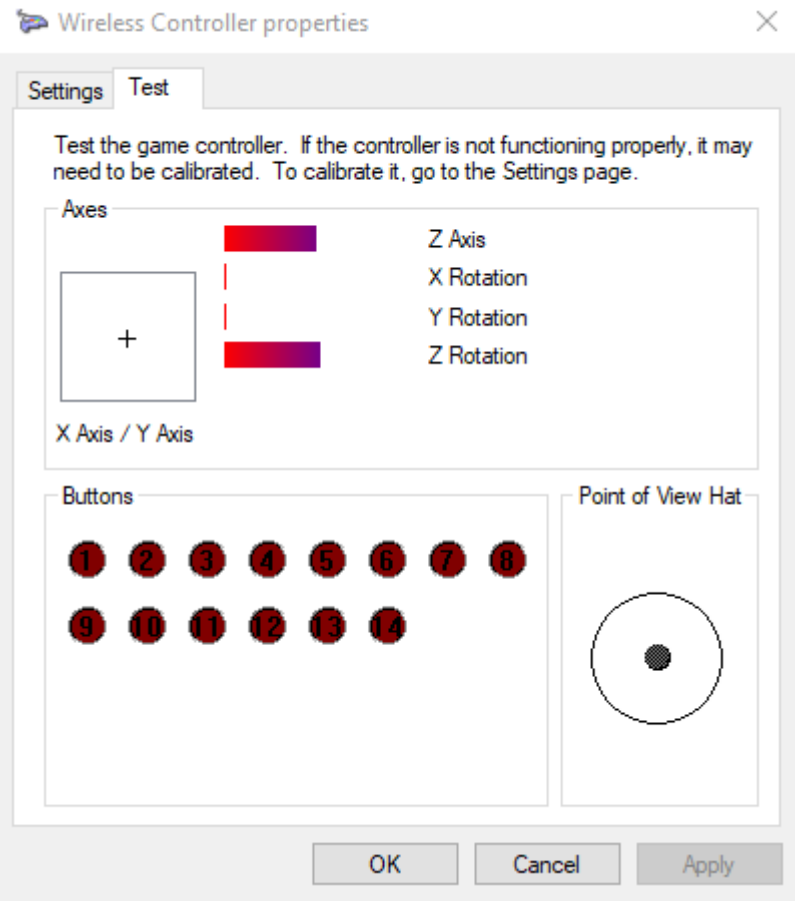


Figure 21, DS4 controller in joy.cpl

Axes

Z-Axis (HALFWAY)	RIGHT STICK X-AXIS
X Rotation	L2
Y Rotation	R2
Z Rotation (HALFWAY)	RIGHT STICK Y-AXIS
X-Axis / Y-Axis	LEFT JOYSTICK

Buttons

1	SQUARE
2	CROSS

3 CIRCLE

4 TRIANGLE

5 L1

6 R1

7 L2

8 R2

9 SHARE

10 OPTIONS

11 L3

12 R3

13 PS HOME

14 TOUCHPAD

D-Pad

The D-Pad was simply represented as the POV-Hat as shown in Figure 21 for which pressing a button represented an arrow in the same direction on the POV-hat.

3.6.4 Resemblance between DS4 and HID controller output

What is interesting is that the buttons: Square, Cross, Circle, Triangle, L1, R1, L2, R2, SHARE(called Select in HID controller), and finally OPTIONS(called Start in HID controller) all share the same button number in the HID controller output and the DS4's joy.cpl analysis. This may indicate that instead of the D-Pad working with degrees, it instead works with X and Y axes like in the HID output in appendix A, section 1. Next to that it also means that if the continuously printed value that is disrupting the HID joystick output is filtered away, the joystick logic could have a chance to be useful and if so, imitated.

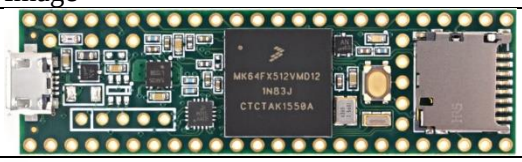
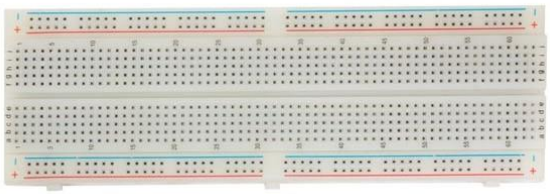
3.7 Prototype setup

3.7.1 Controller implementation

The setup for the controller implementation that will be tested is created for the PC using the XInput library with the Arduino IDE, Teensyduino patch and Teensy cores support for the library as explained

in section 3.6.2. This decision was made so that the concept of the CFG and configuration loading through an SD card could be tested given that analyzing and reproducing the HID protocol would take up too much time for the project.

The hardware that was used has been depicted in Table 2:

Component	Count	Image
T3.5	1x	
Tactile push-button switch	4x	
Male-Male Dupont(jumper) wires	20x	
10K Ohm resistors	4	
Breadboard 64 rows (at least 40 rows)	1x	
PS2 Joystick	2x	

USB A to Micro USB (With data transfer capability)	1x	

Table 2, Hardware used for prototype

The resulting prototype after combining the elements except the cable can be seen in Figure 22.

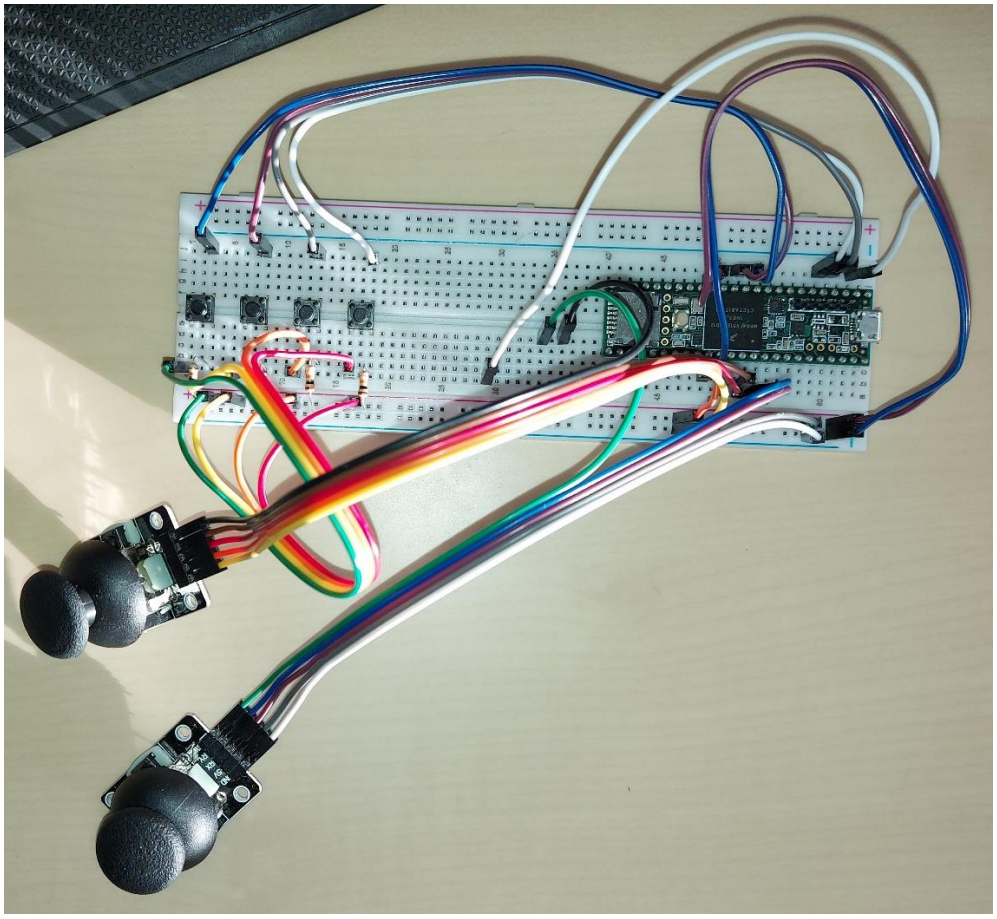


Figure 22, Prototype setup assembled, excluding Micro USB to USB cable

The code used for the prototype includes the basic functionality and a thorough explanation of how to understand the library-specific functions and variables. This can be found in Appendix B.

Checking if the controller is wired correctly, what buttons are being pressed, and how the joysticks perform can be tested using either one of the following methods:

1. Implemented debug mode with Serial monitor as explained in section 3.5.3 point 2.

2. Joy.cpl program in windows 10.
3. Web-based solution of Gamepad Tester [47].

3.7.2 HID Analysis setup

Only T3.6 or T4.x boards work with the USBHost_t36 library, and a header with 5 male pins should be soldered onto the PCB as shown in Figures 26 through 28

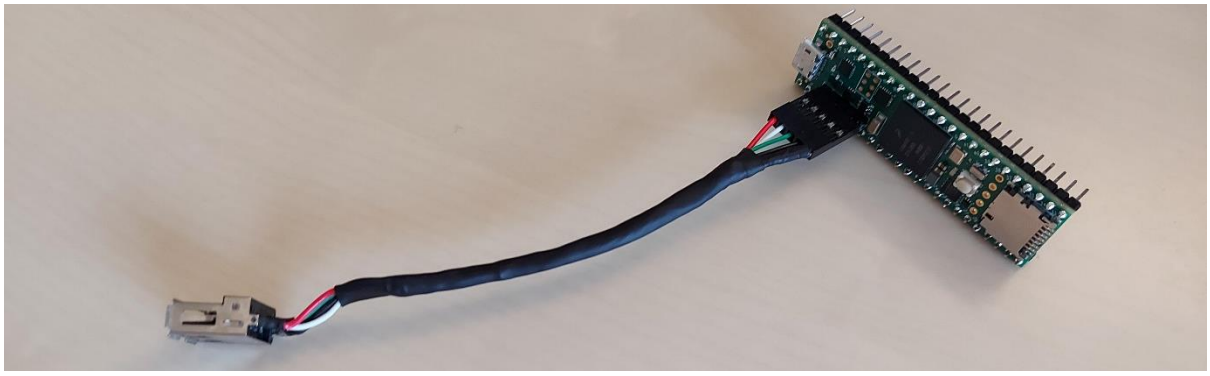


Figure 23, the full image of connecting a USB A female header to T3.5

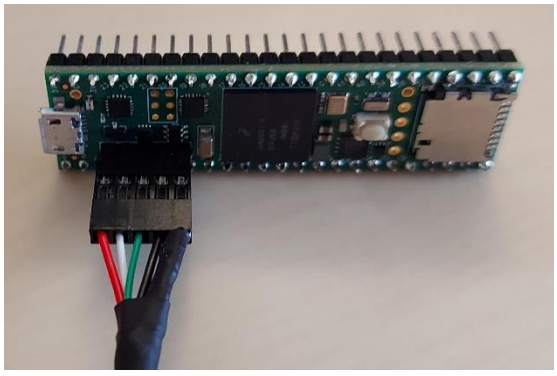


Figure 24, Close up of wiring at board

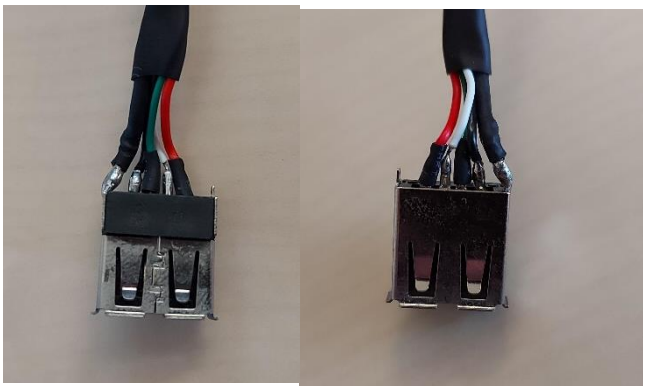


Figure 25, Female header one side and other side

The program that was loaded onto the T3.5 can be found in the Arduino IDE's examples after having used the Teensyduino patcher. To access the file used for analysis with the USBHost_t36 library, go to: *File > Examples > USBHost_t36 > HIDDeviceInfo*.

4. Evaluation method / Results

Validation of the solution is to be acquired through quantitative tests of latency and evaluating the feasibility of the menu-based controller with a use case analysis. Permission has been granted by the ethics committee to test the controller and its feasibility with an expert but there was not enough time to complete this said test.

4.1 Pipeline latency

The latency of the implementation will be tested with a camera setup, where the frames of the camera output are counted from when the button is pressed up to the point where there is output in-game in the form of a muzzle flash, or character movement. The framerate of the camera is static, shown in Figure 30, and the framerate in-game is kept variable to be as high as possible as shown in Figure 26, given that a higher update rate will reduce input lag by default. A low update rate would mean that the controller sends, or computer receives fewer instructions which decreases the accuracy of the movement and shows a delayed response to the input. The camera is a GoPro Hero 7 Black and recorded in 120 FPS at 2.7k resolution (2704x1520). The game Insurgency: Sandstorm was running at an average of 175 FPS.

Figure 27, In-game FPS counter

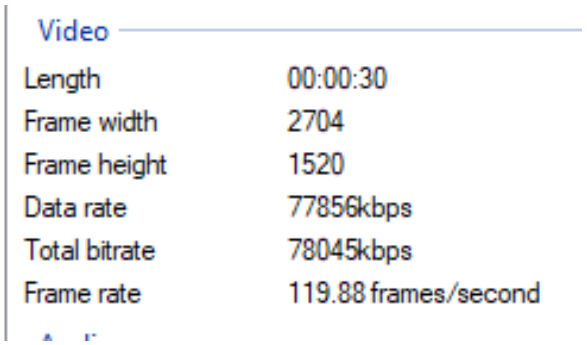


Figure 26, GoPro Hero 7 recording information

The limitation in this setup is the software used for the preparation of the analysis. The free version of Davinci Resolve was unable to output more than 60 frames per second. The resulting video with the camera footage and in-game footage combined shown in Figure 28 was analyzed frame by frame in 'VLC Media Player' using E to skip through the frames and count how many frames passed between a button press and the expected in-game action. The expected action in this case was the in-game character changing his stance from a standing position to a crouching position. The comparison between an unpressed button and a pressed button can be observed between Figure 29 and Figure 30 respectively. The frame time of the transfer between button to in-game movement took 1 or maybe 2 frames but it is not clear enough with the current setup. Given that the output is 60 FPS, 1 frame is

16.67ms. This means that the input latency is at the most 33.34ms but could also be shorter than 16.67ms.

The frames were counted by eye and the recording setup was created at home with non-ideal lighting and electronic equipment. This resulted in poor accuracy of the test but it at least proves that the input latency is responsive enough to not be bothered by a signal delay.

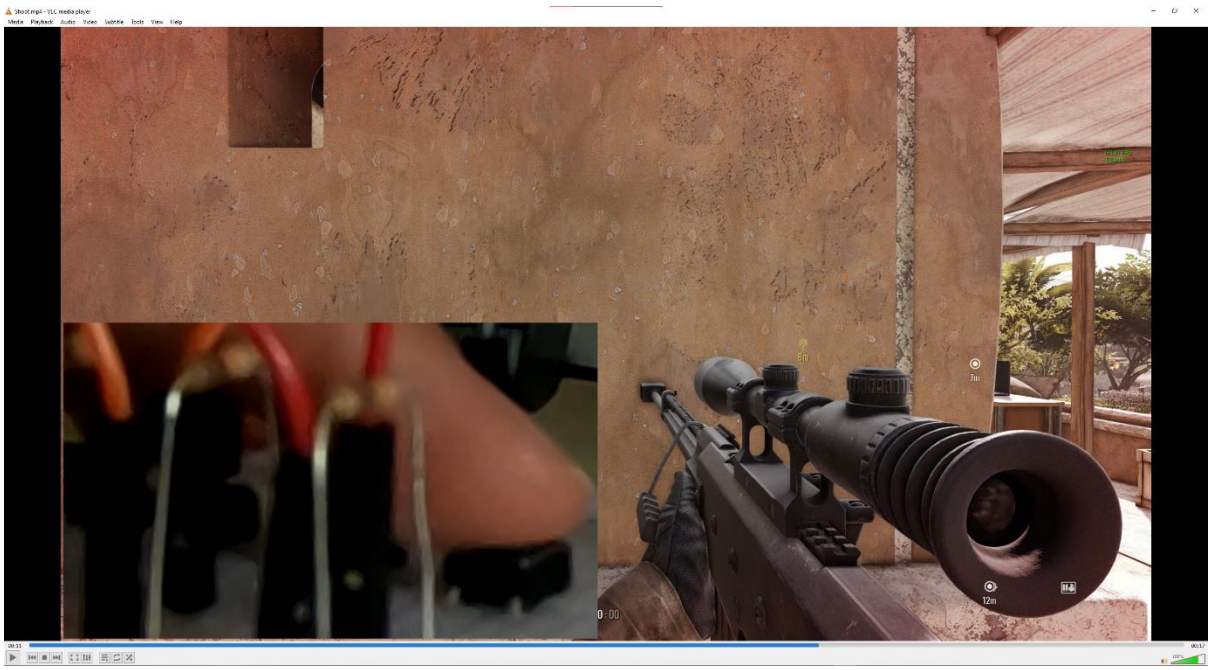


Figure 29, Button press footage overlayed on gameplay footage

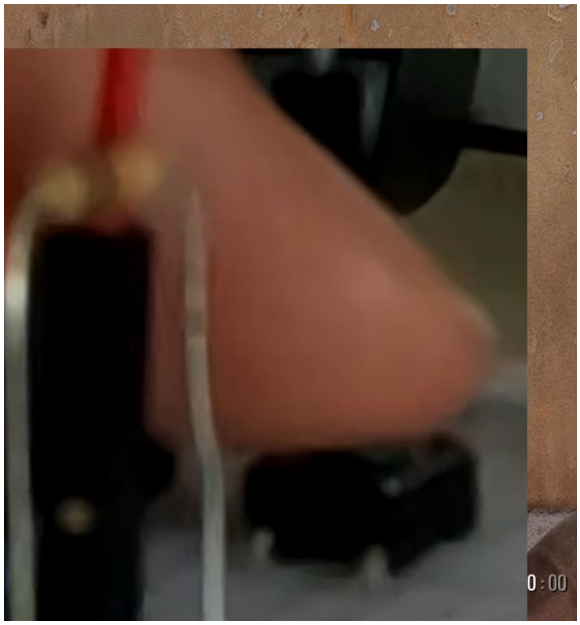


Figure 30, button not pressed

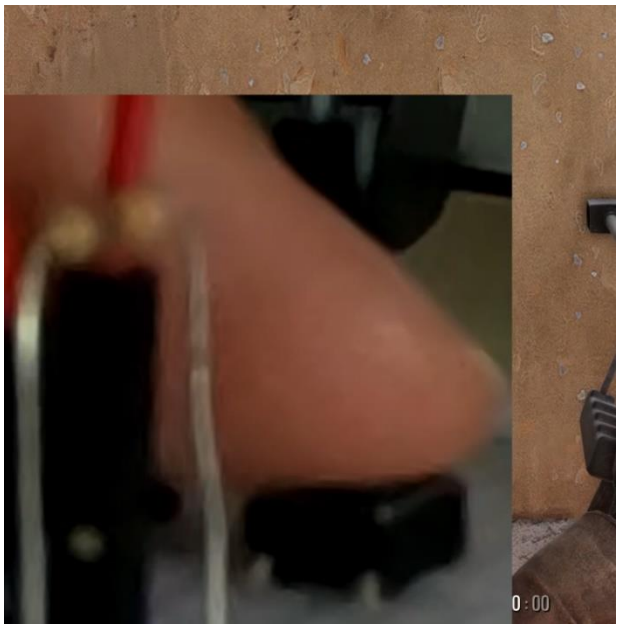


Figure 28, button pressed

4.2 Usefulness of concept

From the opinion of a gamer, with multiple thousands of recorded hours on both PS4 and PC, it is safe to say that the entire concept is very promising and should be explored further and development is created for each console. The reasons for this opinion are supported by several arguments, being:

1. The CFG help gamers, who are unable to game with a standard DS4 controller due to the placement of the buttons and weight of the device, play the same games, but with a lighter controller where the buttons are all within reach. The same functionality in a tighter package also allows multiple designs to be explored for various grip-specific casings and different numbers of available buttons.
2. The button remapping is a lacking subject across all systems, where there are only a handful of studios that use remapping for Sony PS games. Luckily the number of studios that are incorporating remapping into their game on the platform is increasing. However, the remapping for the controller can still be improved and this concept makes use of that. The ability to create a remap profile per game allows for seamless transitioning between games without having to worry about whether it is supported or not.
3. In addition, the ability to remap and configure profiles that make use of combination buttons on a single press is not a known thing to do either. The only system known to make use of this is the PC where this technique is referred to as ‘macros’. With this system implemented in the game controller, it can make pressing combinations of buttons that would either be hard to reach or difficult to press much easier, keeping the game interesting and fun. However, this may be viewed upon as ‘gaining an advantage’ because of there being less movement needed to press the same buttons compared to an abled player. Whether this is considered cheating or unjust would have to be explored further.
4. The additional visual feedback through the LED rings and LCD allows the user to keep track of the position in the menu and game profile that is loaded up.
5. Creating the CFGs is easy enough through a plug-and-play system with an SD card that could potentially take the menu setup on a CSV file and let the Arduino interpret the configurations. This system allows for quick menu editing and allows changes without needing to code.

4.3 Validation Interview

Due to there not being a finished product present that is able to be tested, the general concept, its functionality, and future development of the controller have both been investigated through unstructured interviews with representable parties in order to ensure validation for the future

development path of the project. A qualitative, unstructured interview structure was chosen because when the interview proceeds based on the answers of the interviewee, in a non-direct and open-ended nature, new areas of interest or topics may arise that were not initially thought of.

The expert interview will be conducted with an electrical engineer who has a broad background in the Human Interface Device (HID) USB protocols, which is what the controller will have to be developed with in the future. The interview will be held over Google Meet, with them being informed about the procedure and their verbal consent given orally due to the interviewee residing in the USA. The goal of the interview is to gain better insight on the currently determined future development path of the special controller by mimicking the HID protocols from an existing controller.

The user interview will be held with a gamer with Duchenne, who is based in the Netherlands but is at the time of the interview and course of this project in France, meaning that the interview will also be held over Google Meet, with them being informed about the procedure and their verbal consent given orally. The interview has been transcribed in Dutch, in Appendix F, so that the text has its original message and has not been interpreted through translation. The gamer in question has been on television in the Dutch television show “We Gaan Het Maken” where he and a team of experts worked on the controller implementation for the first time. He has also been interviewed on separate occasions for television and is able to think and answer with his own free will as he is still very capable mentally and is only limited mobility wise. The goal of the interview is to understand to what extent Marc is for- or against certain aspects of the menu and resulting pipeline of the solution.

4.3.1 Expert Interview

The transcript of the unstructured interview in the form of conversational writing can be found in Appendix E. The interviewee is Dr. Theodore Omtzigt, based in the USA and Chief technology officer at Lemurian Labs and has over 30 years of experience in computer- and system architecture and robotics, next to having worked at Intel, NVIDIA, 3DFx, and being an inventor holding 10 patents revolving around high performance computing and evaluation engines. Erwin did not have any prior knowledge of the project. The interview has been transcribed using Google Docs’ using the ‘voice typing’ feature in the extras tab. The recording of the interview was played back over speakers that stood next to the microphone that Google Docs used for its voice input. The output of the resulting text has been reformatted in a better presentable format without altering the answers. The answers of the interviewee have been presented in cursive to distinguish between the voice lines of the interviewer and interviewee. The interview was conducted on the 11th of July, 2022, starting at 14:39 and concluded at 15:52 CEST.

From the expert interview there were two key take-aways. The first being that the HID USB protocol is a multi-byte protocol, which in this case appeared to be 64 bytes, and has a very large state space. The question is about how you explore that state space. It could indeed be an option to replay an existing controller's behavior by reproducing the HID bytes that are being sent in packets, and observe whether replaying the same packets of bytes yields the expected results. There is also a descriptor linked to the device that may contain a state that could be required in order for the controller to function, so this should be explored also. Once the understanding of the protocol is adequate you should be able to reproduce an entire controller, but the amount of time this may take is difficult to determine.

It may even be so that a product specification (Product Spec) would need to be acquired from Sony's DualShock 4 system to properly implement such a solution. Sony will not give you access to the Spec because you asked nicely, you will likely need to pay and sign some form of a contract. It is a billion dollar market place and leading companies would not like to lose control, hence why they control their product specs so neatly. It is advised to try and reach out to Sony in order to get in contact for development of the special controller. Reverse engineering is entirely feasible but does not solve the fundamental problem which is product development. If you were to commercialize the product you would need to look at contacting the appropriate owners of the intellectual property to get the right agreements in place to build the product. Whether that is worth it is debatable because the product cost will not be the problem. What is much more problematic is the supply chain, which will likely costs several thousands of euros.

4.3.2 User Interview

The transcript of the unstructured interview in the form of conversational writing can be found in Appendix F. The interviewee is Marc Jacobs, a gamer with DMD who has some experience with and knowledge of the concept. The interview has been transcribed using Google Docs' using the 'voice typing' feature in the extras tab. The recording of the interview was played back over speakers that stood next to the microphone that Google Docs used for its voice input. The output of the resulting text has been reformatted in a better presentable format without altering the answers. The answers of the interviewee have been presented in cursive to distinguish between the voice lines of the interviewer and interviewee. The interview was conducted on the 12th of July, 2022, starting at 11:23 and concluded at 11:47 CEST.

The current implementation with the stripped down DS4 controller uses the special button menus but only with one configuration and configured for one game, Fall Guys. The implementation worked well for the game, but it was not possible to play other games with the controller. Only if you would dive into the C++ code could you change the button assignments but that would be really

inconvenient when switching between games. It was also observed that there could have been additional buttons on the controller to offer more possibilities, because currently the index finger was resting freely and could also use a button. The desired position would be on the same location as the L1, R1 and L2 and R2 buttons. The gamer with Duchenne has access to a controller converter and Xbox Adaptive Controller, meaning that an implementation could be made like Variation 4, Figure 10 in section 3.2.3 as a first trial, after which the concept could be developed instead of staying stuck on the HID implementation. If possible it would be great to split those tasks between teams to divide the work load and get something to work and test.

It does not matter for the client whether the pipeline requires a lot of components or comes in a small package, as long as the controller is lighter and easier to hold and the solution works as presented in the concept. An impactful quote from the interview was: “It matters to me that I can keep on playing video games, for as long as possible. In what way does not matter much to me.” This was translated from the original quote in Dutch, which stated: “Mij gaat het er in ieder geval om dat ik kan blijven gamen, zo lang mogelijk. En op welke manier maakt dan niet zoveel uit.” Especially with some friends moving towards the PS5, it would be great if the client could move towards that system but the new controller is heavier and wider than the DS4 controller, making it even more difficult to play video games.

In final words, the concept is great with the only remark being that the client hoped to have additional buttons for the index fingers. He quickly got used to the menu switching and was able to remember the control function groups well with the associated color from the LED rings on the display box.

5. Discussion

5.1 Limitations

The PS system does not accept an unofficial controller without a CC in the process. They are extremely hard to come by to this day due to numerous factors like the chip shortages, but are a requirement if this solution is to be developed as in Figure 7, variation 1. The scarcity also affected the research as a CC only became available halfway through the project, delaying any initial testing and prototyping for the desired system. An ideal system would be one where there is no CC needed, but that is only feasible for a solution dedicated to the computer or the Xbox using the Xbox Adaptive Controller as shown in Figure 12, variation 6.

Next to that, the pipeline latency test in section 4.1 had a low resolution due to software and hardware limitations. The latency was shown to be not more than 33.34ms, but could also be shorter than 16.67ms. It indicates that the latency is low enough to feel responsive, but in order to properly determine the latency of the system the researcher should invest in better software for analysis and also take note in creating a decent hardware setup with proper lighting and a camera that can shoot high FPS.

In addition, there was no finished prototype to be evaluated because of the constant obstacle discoveries throughout the development path for the PS and no solution with existing libraries working. It would have been wiser to create a working implementation for the computer to test the concept and controllers and test and report on those items rather than staying stuck in the development of the PS implementation and still not knowing whether the concept is solid enough to pursue. The concept validation interviews were a great help in understanding the needs and expectations better, but there is always a difference between an idea of-, and a real product's functionality.

When conducting the interview with Marc, it was difficult to go into detail on the concept for the CFG and the functionality thereof because of all the technical terms. This resulted in the answers not being completely focused on the functionality of the device, but it did show that Marc does not mind how big the solution is or how many wires are being routed to him. It matters most that he can keep on playing video games for as long as possible. There should be more research done on how to translate the technical terms to something that can be understood easier and better evaluated.

5.2 Directions for future research

What became clear through the expert interview in section 4.3.1 is that there are two directions the future research could take. Either with a lot of bought equipment such as the Xbox adaptive controller and a lot of 3.5mm jack wires which is relatively quick to implement, or a solution that takes a lot of time but would function as a smaller package using HID USB protocols to mimic a HID controller with the Teensy 4.1 microcontroller.

If the project is continued by a single researcher it is advisable to first create a prototype with the Xbox Adaptive Controller, like in Figure 12 variation 6, to first review and evaluate the concept of the CFGs and button remapping's per game. Then, focus on improving the handheld controllers for the options of putting more buttons on. Followed by then developing software for handling the creation of button remapping's per game so that the user has less trouble trying to conform to the data structure required for special formatting like JSON or CSV. Lastly, the goal of the product's package is to fit it all in a compact manner so that it can be used on wheelchairs and does not hinder the user when gaming or the resulting solution would feel heavy when being held.

If this primary prototype for testing the concept has finished, or there is a team available where the workload can be split up and minds can be combined, it is advised to try and create a system that is able to replay the HID USB packets of a HID controller to see if a said string of bytes yields the expected results. If this is completed for an entire controller, the T4.1 should be able to mimic a full controller without the need of an Xbox Adaptive Controller and 3.5mm wire package, resulting in a compact prototype that is even easier to fully assemble thanks to the smaller pipeline and cheaper overall. This would be the ideal solution but is more time-costly.

6. Conclusion

To conclude, the investigation regarding the development of a modular gaming handheld is a lengthy process, especially when exploring the different possibilities of implementations of the final system. It was shown how none of the existing Arduino-based libraries offer a possible solution for the PS4 but were able to reproduce a working controller for the PC. The solution that will end up working for the PS4 will either include a lengthy development plan for configuring the T4.1 as a HID controller by replaying packets that are known to work on an existing controller, or an implementation with the Xbox Adaptive Controller with a lot of wiring which doesn't solve the initial question of improving the existing pipeline.

Next, answers will be provided for the main research question: *"What are the characteristics of an effective and practical profile menu and controller, aimed at users with muscular dystrophy?"* And sub-research question: *"What is the state of the art in adaptive game controllers?"*

RQ1: *What are the characteristics of an effective and practical profile menu and controller, aimed at users with muscular dystrophy?*
In order for the solution to be effective and practical the CFGs and button remapping process must be recognized as being an improvement over existing technologies and implementations. Configuring the CFGs should therefore be configurable through a piece of software where the user has a simple task of configuring buttons by, for example, a drag and drop system. The point is that the user should have to do as little as possible and especially not have to code themselves, given that not all parents of children with DMD are technically inclined but do want to provide a good experience for their child. The general points of improvement that the controller holds over existing technologies was touched upon in section 4.2.

The controller itself should also be modular in such a way that the user is able to either add or remove buttons as they please to a certain extent. The client would like additional buttons for his index fingers for example, but someone else may not. Thus, a 3D model of a controller should be explored where the ability to switch between these two cases is a possibility, especially when it comes to the practical aspect of the controller.

SubRQ1: *What is the state of the art in adaptive game controllers?*
As presented in section 2.1 with the literature review, Microsoft is currently the only company that has put out an assistive technology for its gaming community. It is uncertain why Sony has not followed the same footsteps and dedicated a special controller or aid to their gamers with special needs. Luckily those who would like to enjoy the benefits of the Xbox Adaptive Controller in the PS ecosystem can do so through a CC.

Appendix A

Example HID output of controller without joysticks

HIDDeviceInfo.ino sketch,

Running on Teensy 4.1, controller observed through USB Header. Connected to pc through micro usb.

Send “c” to only show new values.

Connecting:

```
USBDeviceInfo claim this=2000CAC8

**************************************
** Device Level **
  vid=46D
  pid=C21A
  bDeviceClass = 0
  bDeviceSubClass = 0
  bDeviceProtocol = 0
09 04 00 00 01 03 00 00 00 09 21 10 01 00 01 22 3D 00 07 05 81 03 08 00 0A
*** Device HID1 46d: c21a - connected ***
  manufacturer: Logitech
  product: Logitech(R) Precision(TM) Gamepad
HIDDumpController Claim: 46d:c21a usage: 10004 - Yes
*** HID Device hdc1 46d: c21a - connected ***
  manufacturer: Logitech
  product: Logitech(R) Precision(TM) Gamepad
```

Pressing left arrow:

```
HID(10004): 01 80 00 04 89 08 00 38
usage=10030, value=1(X)
usage=FF000100, value=1
```

Releasing left arrow:

```
HID(10004): 80 80 00 00 89 08 00 3C
usage=10030, value=128(X)
usage=FF000100, value=0
```

Pressing down arrow

```
HID(10004): 80 FF 00 20 89 08 00 38
usage=10031, value=255(Y)
usage=FF000103, value=1
```

Releasing down arrow
HID(10004): 80 80 00 00 89 08 00 3C
usage=10031, value=128(Y)
usage=FF000103, value=0

Pressing right arrow
HID(10004): FF 80 00 08 89 08 00 38
usage=10030, value=255(X)
usage=FF000101, value=1

Releasing right arrow
HID(10004): 80 80 00 00 89 08 00 3C
usage=10030, value=128(X)
usage=FF000101, value=0

Pressing up arrow
HID(10004): 80 01 00 10 89 08 00 38
usage=10031, value=1(Y)
usage=FF000102, value=1

Releasing up arrow
HID(10004): 80 80 00 00 89 08 00 3C
usage=10031, value=128(Y)
usage=FF000102, value=0

Pressing select
HID(10004): 80 80 00 01 89 08 00 38
usage=90009, value=1 (BUTTON 9)

Releasing select
HID(10004): 80 80 00 00 89 08 00 3C
usage=90009, value=0 (BUTTON 9)

Pressing start
HID(10004): 80 80 00 02 89 08 00 38
usage=9000A, value=1 (BUTTON 10)

Releasing start
HID(10004): 80 80 00 00 89 08 00 3C
usage=9000A, value=0 (BUTTON 10)

Pressing square
HID(10004): 80 80 01 00 89 08 00 38
usage=90001, value=1 (BUTTON 1)

Releasing square
HID(10004): 80 80 00 00 89 08 00 3C

usage=90001, value=0 (BUTTON 1)

Pressing cross
HID(10004): 80 80 02 00 89 08 00 38
usage=90002, value=1 (BUTTON 2)

Releasing cross
HID(10004): 80 80 00 00 89 08 00 3C
usage=90002, value=0 (BUTTON 2)

Pressing circle
HID(10004): 80 80 04 00 89 08 00 38
usage=90003, value=1 (BUTTON 3)

Releasing circle
HID(10004): 80 80 00 00 89 08 00 3C
usage=90003, value=0 (BUTTON 3)

Pressing triangle
HID(10004): 80 80 08 00 89 08 00 38
usage=90004, value=1 (BUTTON 4)

Releasing triangle
HID(10004): 80 80 00 00 89 08 00 3C
usage=90004, value=0 (BUTTON 4)

Pressing L1
HID(10004): 80 80 10 00 89 08 00 38
usage=90005, value=1 (BUTTON 5)

Releasing L1
HID(10004): 80 80 00 00 89 08 00 3C
usage=90005, value=0 (BUTTON 5)

Pressing R1
HID(10004): 80 80 20 00 89 08 00 38
usage=90006, value=1 (BUTTON 6)

Releasing R1
HID(10004): 80 80 00 00 89 08 00 3C
usage=90006, value=0 (BUTTON 6)

Pressing L2 (button style)
HID(10004): 80 80 40 00 89 08 00 38
usage=90007, value=1 (BUTTON 7)

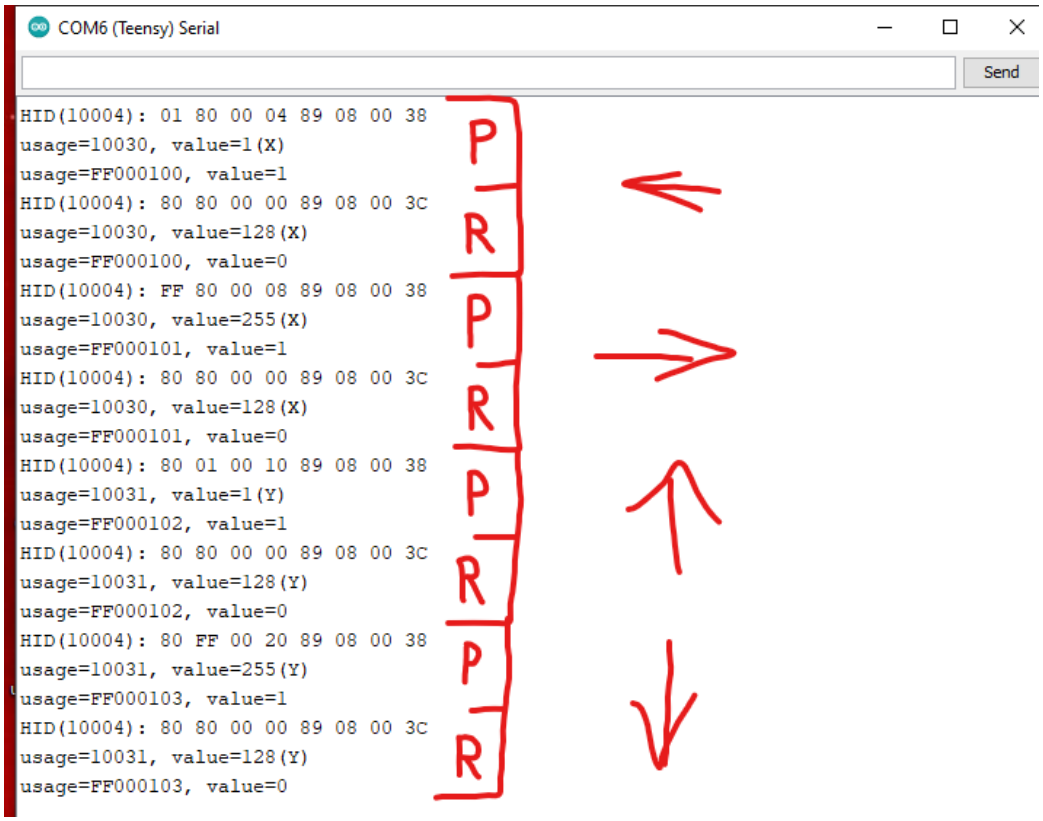
Releasing L2 (button style)
HID(10004): 80 80 00 00 89 08 00 3C

usage=90007, value=0 (BUTTON 7)

Pressing R2 (button style)
HID(10004): 80 80 80 00 89 08 00 38
usage=90008, value=1 (BUTTON 8)

Releasing R2 (button style)
HID(10004): 80 80 00 00 89 08 00 3C
usage=90008, value=0 (BUTTON 8)

D-Pad logic within Serial monitor:



P is for press and R is for release

Values are coded on an X and Y axis, with minimum values of 1 and maximum values of 255, where the activated state is coded as 1. When released the last pressed axis returns to its resting position of 128 and holds a value of 0 as opposed to the active state of 1.

Appendix B

Code for controller used as prototype

```
/*
``````````````````````````````````````````````````
Buttons use press(), release(), and setButton()
XInput.press(BUTTON_A);
XInput.release(BUTTON_B);

boolean yState = digitalRead(ButtonPin)
XInput.setButton(BUTTON_X, yState);
``````````````````````````````````````````````````
Button List:
BUTTON_LOGO
BUTTON_A
BUTTON_B
BUTTON_X
BUTTON_Y
BUTTON_LB
BUTTON_RB
BUTTON_BACK
BUTTON_START
BUTTON_L3
BUTTON_R3
``````````````````````````````````````````````````
Directional pad (D-PAD) (Point of View Hat)
DPAD_UP
DPAD_DOWN
DPAD_LEFT
DPAD_RIGHT
``````````````````````````````````````````````````
OPTIONALLY SET CONSECUTIVE D-PAD STATES
boolean upPad = true;
boolean downPad = false;
boolean leftPad = false;
boolean rightPad = true;

XInput.setDpad(upPad, downPad, leftPad, rightPad);
``````````````````````````````````````````````````
JOYSTICKS
JOY_LEFT
JOY_RIGHT

both have two axis, x and y, and use 16-bit signed values (-32768, 32767) with 0 as center.
north-east corner is positive for both axis.

in the setJoystick() function, the order of arguments is:
```
```

```
joystick ID, x pos, y pos.
````````````````````````````````````````````````

//center the left joystick.
XInput.setJoystick(JOY_LEFT, 0, 0);

//ALL THE WAY RIGHT, SLIGHTLY DOWN
XInput.setJoystick(JOY_RIGHT, 32767, -2000);

//set the range for the input
XInput.setRange(JOY_LEFT, 0, 1023);
````````````````````````````````````````````````

if using micro switches instead of potmeters, d-pad like functionality is supported
boolean stickUp = true;
boolean stickDown = false;
boolean stickLeft = false;
boolean stickRight = true;

XInput.setJoystick(JOY_LEFT, stickUp, stickDown, stickLeft, stickRight);
````````````````````````````````````````````````

*/
// using tut from https://www.partsnotincluded.com/how-to-emulate-an-xbox-controller-with-arduino-xinput/

#include <XInput.h>
//const boolean UseLeftJoystick   = true; // set to true to enable left joystick
const boolean invertLY = true; // set to true to use inverted left joy Y

//const boolean UseRightJoystick = false; // set to true to enable right joystick
const boolean invertRY = true; // set to true to use inverted right joy Y

// Safety measure
const int pin_Safety = 0; // ground this pin to prevent inputs

// Input pins
const uint8_t pin_JoyLX = A0;
const uint8_t pin_JoyLY = A1;
const uint8_t pin_JoyRX = A2;
const uint8_t pin_JoyRY = A3;

const uint8_t pin_ButtonA = 2;
const uint8_t pin_ButtonB = 3;

const uint8_t pin_DpadL = 10;
const uint8_t pin_DpadR = 11;

// output pins
const uint8_t pin_LED = LED_BUILTIN;

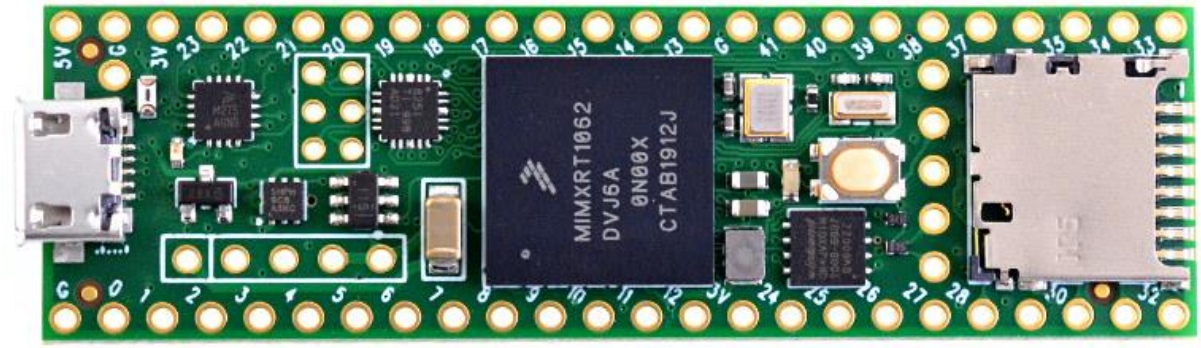
// Analog Input Range
const int AnalogRead_Min = 0; //10-bit ADC
const int AnalogRead_Max = 1023; //10-bit ADC
```
```

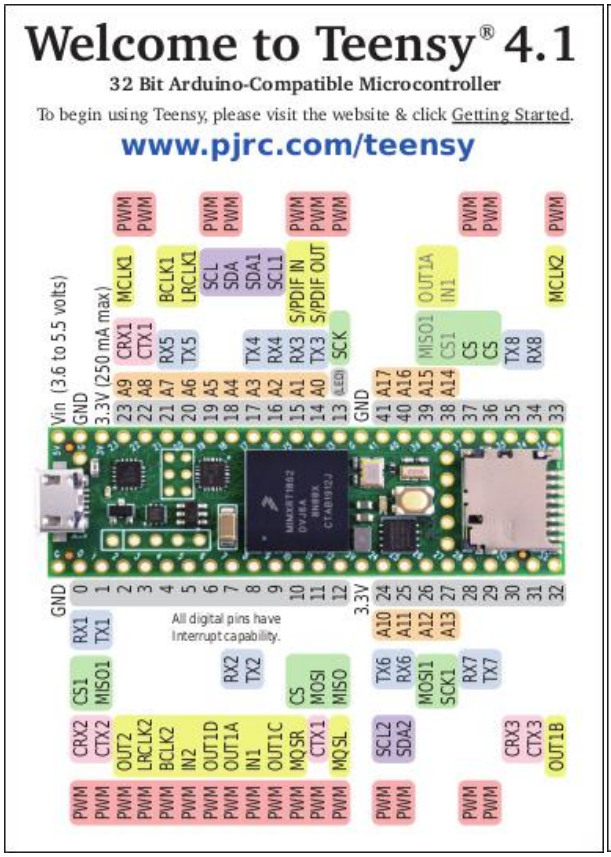
```
void setup() {
//  Serial.begin(9600);
  // Set safety pin
  pinMode(pin_Safety, INPUT_PULLUP);

  // Set input pin modes
  pinMode(pin_ButtonA, INPUT_PULLUP);
  pinMode(pin_ButtonB, INPUT_PULLUP);

  pinMode(pin_DpadL, INPUT_PULLUP);
  pinMode(pin_DpadR, INPUT_PULLUP);

  // Set output pin modes
  pinMode(pin_LED, OUTPUT);
  digitalWrite(pin_LED, LOW); // turn 'off'

  // Define ranges
  XInput.setRange(JOY_LEFT, AnalogRead_Min, AnalogRead_Max);
  XInput.setRange(JOY_RIGHT, AnalogRead_Min, AnalogRead_Max);

  // Setup library
  XInput.begin();
}

void loop() {
// using INPUT_PULLUP > LOW is pressed, HIGH is released. therefore using NOT operator
// treating trigger like a button with setButton instead of setTrigger.

// Safety measure
  if (digitalRead(pin_Safety) == LOW){
    return;
  }

// Read pin states
  boolean pressA = digitalRead(pin_ButtonA);
  boolean pressB = digitalRead(pin_ButtonB);

  boolean pressLeft = digitalRead(pin_DpadL);
  boolean pressRight = digitalRead(pin_DpadR);

//  example on how to set the triggers as simple buttons
//  boolean pressTriggerL = digitalRead(pin_TriggerL); // treat trigger as button
//  boolean pressTriggerR = digitalRead(pin_TriggerR); // treat trigger as button

  int joystickValueLX = analogRead(pin_JoyLX);
  int joystickValueLY = analogRead(pin_JoyLY);
  joystickValueLY = AnalogRead_Max - joystickValueLY;

  int joystickValueRX = analogRead(pin_JoyRX);
  int joystickValueRY = analogRead(pin_JoyRY);
```

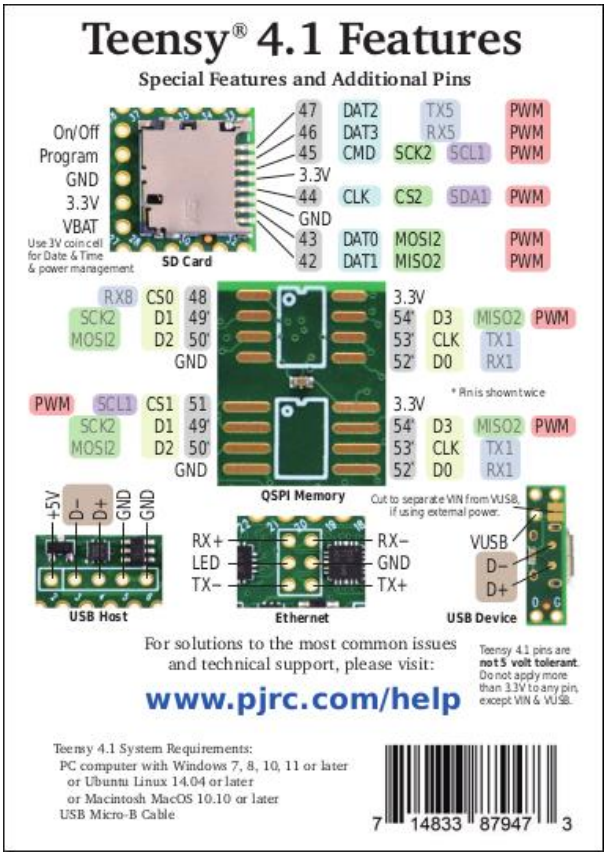
```
  joystickValueRY = AnalogRead_Max - joystickValueRY;

  // Set button, trigger and joystick states
  XInput.setButton(BUTTON_A, pressA);
  XInput.setButton(BUTTON_B, pressB);

  XInput.setButton(DPAD_LEFT, pressLeft);
  XInput.setButton(DPAD_RIGHT, pressRight);
// or XInput.setDpad(dpadUp, dpadDown, dpadLeft, dpadRight);

  XInput.setJoystickX(JOY_LEFT, joystickValueLX);
  XInput.setJoystickY(JOY_LEFT, joystickValueLY, invertLY);
  XInput.setJoystickX(JOY_RIGHT, joystickValueRX);
  XInput.setJoystickY(JOY_RIGHT, joystickValueRY, invertRY);
}
```

Appendix C

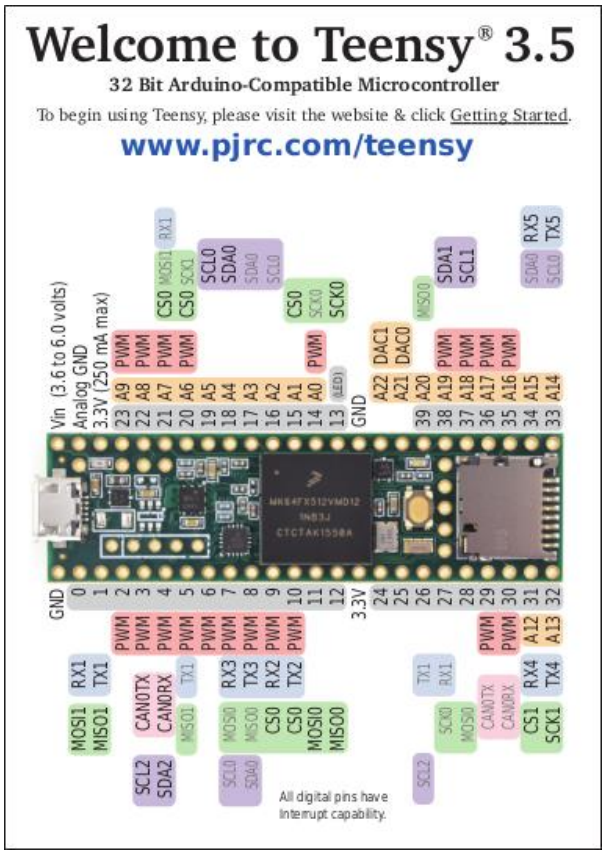
Teensy 4.1 pin-out and special features [29]

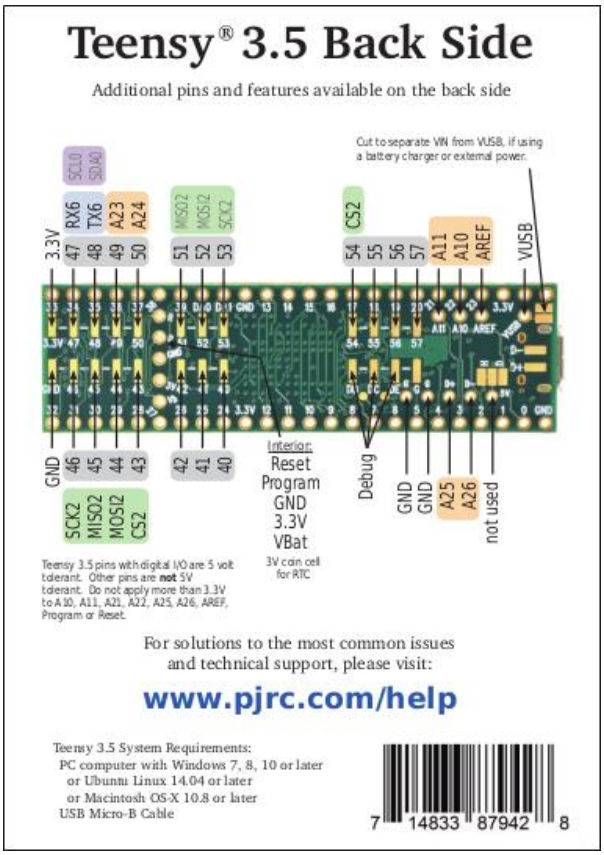






Teensy 3.5 pin-out and special features [48]**.**





Appendix D

HID analysis output in the serial monitor.

A single event, which is a button press:

```
HID(10005): 04 00 0F 80 80 80 80 00 00 00 00 00 00 FF 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=90003, value=1 (BUTTON 3)
usage=FF000026, value=255
```

Button press followed by a button release, two events:

```
HID(10005): 04 00 0F 80 80 80 80 00 00 00 00 00 00 FF 00 00 00 00 00 01 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
usage=90003, value=1 (BUTTON 3)
usage=FF000026, value=255
usage=FF00002C, value=513
HID(10005): 04 00 0F 80 80 80 80 00 00 00 00 00 00 FF 00 00 00 00 00 00 02 90 01 00 02 00 02 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Full structured output of the HID controller using the USBHost_t36 library with the HIDDeviceInfo example sketch:

```
HID(10005)
Begin topusage:10000 type:2 min:0 max:1
  usage=90001, value=0 (BUTTON 1)
  usage=90002, value=0 (BUTTON 2)
  usage=90003, value=0 (BUTTON 3)
  usage=90004, value=0 (BUTTON 4)
  usage=90005, value=0 (BUTTON 5)
  usage=90006, value=0 (BUTTON 6)
  usage=90007, value=0 (BUTTON 7)
  usage=90008, value=0 (BUTTON 8)
  usage=90009, value=0 (BUTTON 9)
  usage=9000A, value=0 (BUTTON 10)
  usage=9000B, value=0 (BUTTON 11)
  usage=9000C, value=0 (BUTTON 12)
  usage=9000D, value=0 (BUTTON 13)
  Begin topusage:10000 type:42 min:0 max:7
    usage=10039, value=15(Hat)
    Begin topusage:10000 type:2 min:0 max:255
    usage=10030, value=128(X)
    usage=10031, value=128(Y)
    usage=10032, value=128(Z)
    usage=10035, value=128(Rz)
    Begin topusage:10000 type:2 min:0 max:255
    usage=FF000020, value=0
    usage=FF000021, value=0
    usage=FF000022, value=0
    usage=FF000023, value=0
    usage=FF000024, value=0
    usage=FF000025, value=0
    usage=FF000026, value=0
    usage=FF000027, value=0
    usage=FF000028, value=0
    usage=FF000029, value=0
    usage=FF00002A, value=0
    usage=FF00002B, value=0
    Begin topusage:10000 type:2 min:0 max:1023
    usage=FF00002C, value=513
    usage=FF00002D, value=400
    usage=FF00002E, value=512
    usage=FF00002F, value=512
  END:
```

Appendix E

“I will be asking questions related to the development of a modular gaming controller.”

Alright,

“for gamers with muscular dystrophy.

Alright,

“the interview revolves around your assessment of the concept and will have a special focus on the future of the project and any ideas and tips that you may have.”

Alright, one second. So we got a game developer. That's the. No, no, gaming.

Yeah, gaming controller and the development thereof will be what I will be asking questions about,

right. So the product is a game controller. The audience is a gamer with muscular dystrophy.

The official term for this is Duchenne Muscular Dystrophy, and that is a disease where the muscles progressively deteriorate over time. also, it affects the joints, so what you get at some point is that the hands start to grow sort of inward, and they get stuck in, ‘this’ position where also the normal controllers become too heavy to hold, the buttons become too hard to press, combination buttons can be too hard to press. So that’s where this controller would come in and solve those issues where they would still be able to play the same games but with a controller that is specifically made for them, to still let them enjoy the fun of gaming.

Alright, how many people are affected by this?

Umm. Well, the idea is that the Duchenne muscular dystrophy group is rather small, but this would also open up a window for a larger accessibility scene in the Netherlands. Where not only people with muscular dystrophy could use this controller, but also maybe as a visual aid or people with missing limbs.

What is the visual aid piece?

The yeah. Wait, I still have to finish the consent. We'll get to that in a second, alright. Yeah. So As for the procedure and how your data will be processed, this is an online interview where the audio of the conversation will be recorded and partially transcribed to be presented in the thesis. And you may leave at any time during this interview, but please don't.

Alright,

so do you understand the procedure and how your data will be processed?

Yes, I do.

And do you consent to freely participate in this interview of this research project?

Yes, I do to my free will. My own free will.

Great. All right. Then I will be sharing my screen. Umm. To show you the pipeline of the solution. Let's see, that is 3.3. Let's see. Is that clearly visible?

Yeah, let me see that. OK. So we got, blah blah blah. Visualization test controller DS4. … into a microcontroller. Design requirements.

Yeah. So the requirements are the controller where the big black circles are, the joysticks and the square circle, triangle and F button are all separate buttons, of which the most right button is a button that is able to switch through function groups and the function groups include different sets of buttons so that you, so that you can emulate a controller with 25 buttons. With this controller that now only has, let's say, three active buttons.

Yep, understood.

The idea is then to also make it so that the joystick is able to create button input as well, so that if you would move the joystick up, it would also be a certain button press, and that's all done through code. Um and the microcontroller that handles that controller input is a Teensy 4.1. Are you familiar with that?

No.

Alright, I'll quickly show it. Let's see 3.2.2.So this is the microcontroller

oh OK

And it. Was chosen because it has an integrated SD card slot where the button remapping profiles would be saved onto an SD card coded in or formatted in either CSV or JSON, so that the. C++ code would be able to read those profiles and configure them for the solution.

Right.

Umm. Now. Let's see. Back to …

Sorry, what did you call this processor?

It's Teensy4.1.

But the. Uh, but the key is that it's an, an ARM. Widget?

Yeah, it was also like because of the. Processing capabilities it would be Much faster than the standard Arduino to be able to handle all of the things it will have to do eventually.

Is that a lot?

Well, compared to standard microcontrollers, it is, surprisingly.

Interesting. Yeah. surprisingly.

Yeah. Yeah.

600 megahertz to emulate a couple of buttons. Okay. Educate me. Alright. OK.

Yeah, let's see. So. Umm. So that controller will handle all of the inputs and outputs and also the visualization of what menu you are in through. Two LED rings. And depending on their color, you will know what menu you are in which is.

Are you meaning an an actual finger ring?

Oh no. Like. LED's that have been formed into a circle.

OK.

And they are placed on the table top of the wheelchairs of the people with the disease.

Yeah.

And when they are gaming, they will be able to see the color in their peripheral vision instead of having to focus on a lettered LCD screen where they would have to actively. Reading and looking at the display.

Interesting are they color LED's?

Yeah, yeah, they're full RGB yeah.

Why would you use it in a peripheral? I guess your peripheral doesn't have color vision,

I mean just so that you can observe it in the corner of your eye. Without having to fully focus on it.

OK, interesting. So that's the. So that's the visualization of the groups.

Yeah. Yeah. So for example, Group One would be Red, Group two would be blue and then group three would be green. And those are the groups you would be cycling between. And the children with Duchenne muscular dystrophy are familiar with such menus because that's also sort of embedded in their wheelchair control.

Oh, interesting.

Yeah. So they are sort of used to the process more than we are. So they will pick it up rather quickly. I presume, yeah, but yeah, I haven't been able to test.

How did they come to be? How did you guys learn about that?

Well, I picked this project up from a person who made the start of it. Them. And um, he worked closely with a person with Duchenne muscular dystrophy, and together they figured out that this concept works rather well because, well, A they play their games a lot because that's sort of their socializing. They can't go out.

Oh, good point

so, yeah, so their gaming isn't just a. Yeah, leisure activity for them, it's their socializing. It's their communicating with friends, keeping in touch. Umm. So they will be spending a lot of time with this system and given that this sort of resembles their wheelchair controls in a way they will be able to pick it up rather quickly as the idea.

makes sense yup perfect. All right, got it.

Now the tricky part is. The sending the signal to the PS4 because of its proprietary nature. The PlayStation only accepts official controllers. Umm. And in order to send a signal from a non official controller you have to sort of trick the system into thinking that it is an official controller. And this is done through existing devices called controller converters. Let's see. Did I mess up the order? Umm. So these are controller converters or controller emulators. Where your controller. Would. Be plugged into the input. See can you see my cursor?

Yep.

Yeah, again. So the input would take your controller and the end of the USB would be plugged into your PS4.And inside here the. Controller converter would take your signals and translate that to um. Sort of. The dual shock for communication protocols so that the PS4 understands. Uh. That it has to do something.

Alright.

Is that sort of clear?

Yeah, that's trivial.

Yeah, right. So now the problem is that the existing libraries and human interface device protocols that have been written to emulate controllers all send really weird. Formatting or they output the

analog signals and the directional signals in a weird format that the controller converters do not understand. Whereas normal controllers like let's see. This one right here.

Oh, I'm sorry. I was blocking my notes. OK. Yep, got it.

This is a regular controller that works on your PC, and when you plug it in with the USB into one of those controller converters, it works flawlessly like a normal controller. So because of none of the existing libraries functioning like they should. The idea is to analyze the output of these controllers and mimic that. Umm. By sending the yeah, same exact. Byte sequences for a certain button press.

Alright, there's got to be a spec for that somewhere, right?

Yeah. Yeah. So The thing is that there is existing PlayStation 4 communication protocols out there by someone who analyzed it. It's just that I am not sure if this is uh. Yeah, how I should be approaching this?

OK, well, let's. We'll we'll get to that. Uh, because we'll have to do some little background check.

Yeah.

If you find a spec. Um, and then the the next question is, is like, why would the libraries not? Why would? Why is there a disconnect between the libraries that you have? And what their functionality is?

Yeah. So all the libraries. Umm. They function to make the microcontroller into a controller that the PC understands, right? It is not made for. Let's see. It's not made to turn your microcontroller into a human interface device. They are specifically designed to turn your microcontroller into an Xbox controller for the PC using XINPUT and Directinput APIs instead of the human interface device. Protocols. Like in the regular controllers. So that that has a OK to say that one more time. So the libraries they turned the controllers or they turned the libraries, turning the microcontrollers into a controller that works with Microsoft Xinput and Directinput APIs so that they have full functionality for a PC, and not necessarily. For emulating a human interface device controller. And that's where that goes wrong.

Hmm Human interface.Controller. Why? Why is that a different spec?

I'm not sure I I don't know where it goes wrong. I think the there is a difference in the way that the bites are being communicated. Because there's this thing. In. Windows. Let's see. do i have a picture of it. Yeah. So this is a program in Windows called Joy.Cpl and just made specifically for testing controllers.

All right,

now if you would plug any controller into your PC and open up this program, you would see this menu where the buttons are represented. With simple numbers. And the joysticks are represented with X&Y axis and. Reset axis and set rotation. For the right joystick. So. When you. Code A microcontroller with one of those libraries that already exist. The PC is able to understand it perfectly. But for some reason the controller converters. They don't know what to do with the that same input. So all that I've been able to get through to the PS4 using the existing libraries are simple button presses.

OK.

And none of the joystick inputs, which is actually essential. Because without joysticks, you can't play any games.

Right. OK.

Yeah. So the idea now is to analyze an existing controller like this one, see what output it gives. Because and because of this one working. If you would emit emit or imitate that output. Umm. And sort of yeah, reproduce the byte sequences that it creates. You would be able to create the same controller. In. For my logic. Yeah, and. And then, without having any of those libraries doing the work for you, you would have to create this from the ground up.

Yeah.

And. Yeah, but I don't see any other way.

Well, find the spec. This is uh, this shouldn't be a reverse engineering problem. That's that's. Uh. So there clearly is a spec to the PC controller. There is clearly a spec for the emulators. There's clearly a spec for the PS4.So.Getting access to the specs. Uh,

what do you mean with specs?

Specification for that controller. They they don't get created out of thin air.

Yeah, yeah,

there's some spec that they need to adhere to adhere to.

Yeah. There's some sort of security handshake happening between the controller and the PS4.And if you plug in any random controller, that controller sort of gets denied because it is not. Umm. Conforming with that security handshake that the digital security handshake to say like, hey, I'm a PS4 controller and Oh yeah, I'm a PS4 system welcome.

Now these emulators have that apparently.

Yeah. Yeah, these emulators there. Let's see this size. And um. Yeah, all that magics happening in here and I do not have the USB analyzing tools to. Um, know what's going on inside of these.

Umm. But the. The. Project manager of this project, yeah. They should provide you these things. They they don't. They shouldn't give you. Go reverse engineer it.

Yeah, that was sort of the job I've been given. They were because my job was to simplify the existing pipeline, which was to simply take apart a PS4 controller and then rewire the connections to those special controllers. Which is quite a hell of a job because it's a flimsy PCB inside those controllers and not a regular PCB. Umm. So yeah, my job was to simplify the pipeline and all the simplification. Is starting to really go the other way.

OK. Umm. So I'm I'm let's let's uh I need to get an understanding of. When you say simplify the pipeline.

Yeah,

that means that there was a pipeline that was already working.

Yeah, yeah.

And So what was that pipeline?

So that one was the you had the controllers that have not changed, they connected with Ethernet cables to a microcontroller and that microcontroller sent digital signals. To a PS4 controller with wired connections that had been taken apart. so the PS4 controller was taken apart, wires were put from the micro controller to the PS4 controller on its PCB. And then the PS4 controller was simply able to connect to the PS4, but instead of pressing the button. Um it that button press was now handled by the microcontroller. But in order to yeah, create that solution for people, you would need like. Uh. You would need experts that are able to do that sort of taking apart and soldering job. Bless you. And that's not feasible for to make for everyone. So if there was a solution where you would just have the the plug and play ability. That. That would have been better, and that's what I was set out to investigate.

OK. So in that wired configuration, what specification documents did they use in order to? Make these connections.

Um, no. They they simply sold their to the existing connections.

Yeah, soldering a wire doesn't do anything right. You have to have still protocol compliance. You need to have. You need to have some spec that tells you what wire is what.

Oh, like that? Like, oh, what the. Umm.

All I'm asking. OK. Why is the wired Connection. Circumventing this security. Handshake

the the controllers that existed. Wired to a microcontroller which was then wired to an existing dual Shock 4 controller. So I an existing PlayStation 4 controller. That was taken apart and used so the security was there in the system, but you had to take the controller apart for it and solder your connections to it, which is not feasible for. Yeah, everyone.

I I'm OK explain to me why you why the security protocol is still there.

Yeah.

So why wouldn't you be able to emulate that then?

Well, in my solution,

yeah.

Yeah, the idea is that my solution. I'm sort of circumvents the whole taking apart and soldering job and would make the special controllers a plug and play solution.

OK. And and so that's not working because you're not getting past the security handshake.

Well, yeah, these security handshake is now handled by those controller converters that I'm using. They they are able to do the handshake and then take any sort of controller as USB input and make sure that that controller works. It's just that I'm not able to. Emulate a controller that they those controller converters understand.

Understand. OK, I got it. Umm. So these emulators work under some specification, yeah, so. Yep, let's get that specification.

Yeah. And how would you? Go about that.

Uh. Well, the there's got to be a. Uh, yeah, this should be the role of your advisor. For God's sake, let's see.

Because there is a PlayStation 4 developer Wiki. With. Yeah, certain information about what information goes between the PlayStation 4 controller and the PlayStation system itself. Umm. But there are only I think. Two companies and one single person who have been able to develop such a device.

Let's see PS4 emulator controller. Uh 'whole Tia weekly P4'.Let's see official PlayStation 4 emulator. Uh blah blah blah.

So the names of such devices are Titan, one and Cronus Max PLUS.

Uh Titan one, Cronus Max plus.

I'm not sure if that's. Limited to Europe, maybe in the US they would have a different. Sort of thing, but those are the two general systems that are used. In the solution such as these.

Alright. Uh, Cronus Max Plus hardware guide. Blah blah blah. Uh. Yeah. So. Umm. DualShock wireless controller controller menus. OK so. One way. Well, OK. The question you had is how would you go about this? Fundamentally, you go about this by by finding the spec Now 999 out of 100 times. That requires exchange of money. Because this articulates a product and a product, people spend money on developing products for and they like protection to not have Willy nilly come into these spaces and therefore there tends to be some special interest group. That guards the specs and you need to become a member of that special Interest group to get access to the specs. And. Normally you if if you're in this space, would pay the money to get the spec so that you can build products that are adherent to a spec, right?

Yeah.

So that any of this, it's not compatible. Craft doesn't enter your. Your your vocabulary so. So the question that I have now is. Why did they give you a reverse engineering task that is? Aggravating, yeah. Um, you're here to freaking learn how to become a EE. And this is high school maker crap. Um, so is that the intent of the program? That's the. The question then it cannot be the intent of the program. You're a freaking engineer.

Yeah. The intent of the program was to simply create a controller and test that product and then. Yeah.

So the input subject such a project is A is a specification of what your product needs to adhere to.

Yeah.

OK. So if you don't have that. Umm. OK, this is aggravating. Because I, you know, I don't know what what lives in the constraints of. Your advisor and what he or she has access to or is willing to pay money for. And maybe maybe that persons idea of getting you exposure to engineering problems was OK, why don't you reverse engineer a controller? Which? Which I would find. Questionable. Um. OK, so I'm I'm a little bit at a loss of how to proceed.

Yeah, because I think the the project took a pretty significant turn throughout the process because of certain things failing, which led to me having to reverse engineer such a controller in order for anything to work.

Right.

And it took way too much time. In order to get at that point itself because of all the research and development. That went into finding possible solutions. Because the idea of such a. Project for us as

students is to. See the need for a solution somewhere using technology. Develop a certain product that would solve. That problem and then test that with the user and evaluate. How well it works. And that was also the idea for me. But then yeah, this. Sort of got out of hand with the solution not being feasible for just a single person in this time span.

Yeah. see reverse engineering is.99 out of 100 times illegal.

Yeah, this is... Yeah. But if if I were to reverse engineer like the the PS4 security handshake and the PS4 communication protocol and. Make a product out of that and sell that. That would be illegal. But if I would simply mimic. The communication protocol of such a controller. Umm. That is able to send its input to the controller converter which then understands this input and. Creates. Yeah, a usable controller that is able to give the same experience. That would be OK.

Uh, alright, so we're looking basically at PC controller, game controller, game controller. Controller. USB, no, game controller spec. Blah blah blah. Yeah, this is crappy. Umm. So you say this is a USB. The game controller USB. Let's see Google.

Yeah, I can show you these sort of outputs that it gives. Umm. Let's see. So am I still sharing the Word document?

Yes.

Alright, so the image that I have selected here. That is the. Yeah, sort of string of bytes that the. Controller would send to the microcontroller when a button when a simple button is pressed. So it's a I believe hexadecimal format. Where a simple set of values is changed whether button is pressed or not and it is then now my task to sort of make that a similar. Or will actually make that same? Communication protocol. Because the controller converter understands this certain specific controller. So if I would simply copy that and use that, that would be fine,

yeah. Well, the way. I would do it, so here's the here's the problem that. You know. This type of approach. Would have is that. Um. Without knowing the spec. There is 7341 different ways this can go sideways right. There could be special bytes that you have no awareness of that you know cycle through modes and whatever, right. So it's a. The biggest problem of such task is that. How do you explore the state space? Because you don't know the state space and that state space may have a flip bit somewhere that moves you to a completely different segment of the state space, right? So it's a, it's an electronic system. Electronic systems have a particular set of states they can be in. That's called the state space and your reverse engineering task is this mapping of all the possibilities in this state space right, now. Given the fact that you're talking about a multibyte protocol, this is a very large state space. Right. OK. So the way I would do it. Is without knowing the state space. You cannot

approach it from first principles in the sense that you cannot say. I hypothesize that this is the protocol, or this is the machinery that creates these commands. What you need to do is you need to get to a point where you can replay. That controllers behavior right? So you don't know first principles, but you will be able to test whether or not experiments with these bytes yield the same result, right? So the way I would do it is I would build a an interposer and a replay mechanism. Where you basically used that game controller. And every time you have a command, you are recording that that output. OK.

Yeah. So that's what you see. The Word document that is the recorded output when I pressed what was it circle for example?

Lovely. So now the hypothesis is that that means it's a circle button press in a certain control mode.

Yeah.

OK. Now. Can you replay that? Can you stick that same USB protocol packet into that emulator and actually get a? Round button press response.

Yeah. And what would you call such a string of hexadecimals

packet. A packet.

All right.

Yeah. I mean, there's so that you know, you're basically having a state. You. Create an event or an event is created that. Trigger some other event. Right. And your goal is to try to figure out of all the events that you can generate. What are the responses? To that USB or what comes out of that USB. Yeah, yeah. Output, yeah. Umm. So. As I said right the. So how did you get those packets?

Um, that same teensy microcontroller um is has a. USB. Yeah, it is. It is able to take USB input. Umm. So I connected the USB connection of the controller to that microcontroller.

Yeah,

and there is an existing library that is able to read human interface device info.

Yeah.

And then you simply would open up the serial monitor. Using that microcontroller plugged the game controller in. And then it I would press a button and or you you would see the descriptor first. I'm saying like hello. I am such a controller. I expect this kind of input and I give this sort of output and then if you would press a button it would show such a packet. And when you release the button, it would also show such a packet with numerous other packets in between that I was unable to define. But yeah, this was a quick analysis, basically of the. Human interface device communication.

Yeah. OK. So these intermediate bites. ummm. Who is responsible for those intermediate bites? Your USB controller Probably. Umm hmm. OK, back to the question at hand.

Yeah.

Uh. You probably have a couple of those packets characterized that resume.

Mm-hmm. Yeah, yeah.

And do you have a mechanism to? Let's see in that microcontroller you should be able to replace some of these packets onto the emulator, correct?

Yeah, I'm just not sure. Like should I just be replaying the entire packet like it's presented here?

I would,

yeah. OK. So.

I would, but the next question is, is what are these other USB packets represent? They they may be slot allocations, they may be. Umm

yeah.

And there may be state management things that you may need. Um. But yeah, so that interposer is the teensy. If you can. Uh. Use that teensy to replay these packets. And then you would have a little software library that allows you to say okay round button press triangle button press, X button press F button press.

Yeah,

right. And then you should be able to emulate a control sequence. That you can test. If. So from a software engineering point of view. You would use the Teensy To Replay these packets, right? So the characterization job is identify the packet that is associated with an event on the human interface controller.

Yeah.

Then. Store those away. And then build a software interface that allows you to. Replay a particular human interface control sequence. Right. Yeah. So now. You would have the ability to test whether or not your. Understanding of the protocol is correct. Yeah, right. And also it allows you a really nice reproducibility so that you can actually guarantee or get confidence that the software you're producing. Is doing the right thing. Right. And it it it the nice thing about it is that. All of your engineering now is reduced to software on the Teensy.

Yeah. Yeah,

right. So you don't need any of the stinking controller crap. All you can do everything with the Teensy,

yeah.

That's. Umm. So that way. You get confidence that. Uh, you can actually drive commands. Into the emulator that the PS4.Understands.Without having to worry about the rest.

Yeah, yeah.

Umm. So that Teensy now is your interposer, right? The Teensy can be used to receive the packets and record the packets.

Yeah.

Then if you have these packets associations with events, then your software could say hey, I would like to create this event. And the Teensy would then say or your software would say, hey, for this event I need this sequence of packets. I have the packet. Content. So I replay this sequence of packets. And that should have known. Response on the PS4.

Yeah, yeah.

Uh, OK. So. That gives you at least the ability to make some progress, and what is the expectation now of this whole thing? Of your product.

Well, because I was not able to produce a functioning end product that I was able to user-test. The only type of validation that I can get for my concept is going to be through expert interviews. So to finish off this project, I will be conducting this interview and then saying like, OK, you work a lot with human interface device protocols and you gave this advice for the tech side of it. And that sort of functions as validation but and that's sort of concludes the bachelor thesis project, but I want to continue this still. In the summer break. Where I would be produ- Yeah. Reproducing these packets with the button presses.

Yeah, there is. No. There is no replacement for actually having the system wiggle.

Yeah, so that it works the controller. Um works for the PC. But the. Sort of the the idea of the project is to make it work for the PlayStation 4 and PlayStation 5.And if this ends up being too difficult, I it it might need to be concluded that it simply is not feasible. To go further in development for this.

Yeah. You know, when we develop product and you have a market. That is. Bifurcated or fragmented right within? A controller from. Vendor A and a controller from vendor B.

Yeah.

That are incompatible, like PC versus Apple for example, right?

Yeah.

Um. They tend to be separate projects because the technology stacks for those are different,

yeah.

Umm. And typically, what is more important is the supply chain for these products is different. Right. So you're the components you got to buy the sulfur pieces that you need to develop? The the specs you need to buy are all different. And. Apple is a great example. Apple will will. And I think Sony is the same way. Apple will not. Give you access to their gear unless you sign the contract that you are an apple supply chain partner, right? So. So back to that question at hand. It's not so much that it's feasible or infeasible. It's entirely feasible, but normally. Your program manager. Needs to. Interface with all the right people in the industry to get special specs to get. Testing equipment to get. The prerequisites in place so that a product developer can actually develop a product. Yeah, yeah. Umm. You you you do not reverse engineer. And for the simple reason that. It's it's a stupid exercise. That's the first thing. And secondly, and nine-, as I said, 99 of a hundred times is illegal. Right. I mean if you get a software license from Apple or Microsoft, the first thing in the clause says you will not reverse engineer this stuff.

Yeah. Yeah,

right. So. Two for a college. Um project? To ask oh, why don't you go reverse engineer in my mind is inappropriate and. Umm yeah, it's just. It's inappropriate. Certainly when you talking about interfaces to systems like PS4.Which are very. Litigious. Um organizations and The funny thing is, is that. They're not unreasonable. They're litigious so that this reverse engineering stuff is not affecting the product.

Yeah, yeah, yeah.

But you simply call Sony and say, hey, I am working on a controller for muscular dystrophy, would you be willing to participate. Of course, they say yes. Right. Yeah. Oh, it's a really strange situation. You find yourself in. Umm. So you know, I, I I find it very odd and inappropriate. Let's put it that way.

So your advice would then be to maybe first get in contact with Sony, try and pitch this accessibility controller instead of having to try and reverse engineer and possibly enter illegal territory.

Yeah, absolutely. Always play these things above board because one reverse engineering is a inappropriate activity. Uh secondly. It doesn't solve the fundamental problem, which is product development,

yeah.

And product development requires that you become part of the supply chain of a product like PS4.So don't start on the wrong footing with your supply chain, particularly Sony.

Yeah. OK.

So yes, my suggestion is is that if you want this as a product. Then talk to the appropriate. Um. Owners of the intellectual property. And get the right. Agreements in place that gets you access to a specification so that you can build a product.

Yeah. I mean-

think about it right,

it’s the. Logical solution, yeah. Yeah. But the sort of the idea of this project, was sort of just shoved my way like, oh, this this doesn't work. Just analyze a human interface device controller and mimic its output. So. And if I didn't do that, I wouldn't have had anything to do in the report or to report on so to say. So. To me, it was all very weird to even start on a solution for the PlayStation 4 when I could have been focusing on creating a solution that did work on the PC like I have the PC implementation ready to go and develop further. But I had to focus specifically on the PlayStation. Which yeah, because my advice to those kids would be like, listen, I'm going to be taking years for this. You might be. Not alive anymore. When this is finished. umm it's easier-

That’s a little morbid

yeah yeah no it's true because they don't live very long and they would be sad to see them progressively get worse conditions when i could have been making this solution on the PC ready in maybe a few months instead of years.

So really the complexity from an engineering point of view is equal, right?

Yeah.

If you can do this in a month for. The PC with the right tools, you can do it in a month for the PlayStation.

Yeah, yeah, yeah. And then again, like with the right tools.

Exactly.

Yeah. If if I give you a screwdriver then. It ain't going to happen. Hmm. Umm, so I'm still confused about what the goal was for this project, right? It's a. It's a bachelor's thesis project. Yeah. Yeah. So the goal was to create a controller and user test that and sort of develop. A basis with background research that this solution is needed. But I think they should have focused on creating the solution for

the PC where development is actually possible without having to reverse engineer and enter illegal territory.

Yep, I that would be my assessment as well, yeah. Um. And. Yeah, it it the having a. So the one thing we don't haven't discussed yet is is it worth it? Right. So the audience for this product? Or the product road map. Is something that. Um. If the muscular dystrophy. Product or the product for mollusk-.Malo-.Muscular dystrophy. Disability. Yields insight in. Weight and force requirement for controllers. Then. That could conceivably be expanded. To other disabilities. Yeah, right. So muscular dystrophy is the, the, the prototype environment in which we can demonstrate certain new requirements for controllers. Right. And the product attributes you described which is. Umm. You have these function groups and you have a visualization mechanism that is congruous with wheelchairs, wheelchair products that these kids are already using.

Yeah. Yeah.

That's a nice product requirement, discovery and association. But what is it? What is it? Um. Actually, if you sort of look at the the product cost, right, development of a product like this is not the. It really should, should be a, you know, ten thousand, $15,000 kind of problem, right? What is much more complicated? Is the supply chain to get this particular product marketed and delivered to these particular organizations? That is a much more expensive that is measured in the hundreds of thousands of dollars, right? So product development and product requirements analysis is a infinitesimal amount of dollars. For delivering this product to the customer. Right,

yeah.

So. The and this is another reason and another argument that says focusing all this or wasting all this energy on. Delivering something for a a particular segment of the market when you have the ability to do it for another market and a much lower cost. Is an argument that you should never have been asked to do that.

Yeah, no, I agree.

It's a failure of project management. That didn't recognize the complexities in this in this problem. So.

Yeah, yeah,

the, you know, the argument of, like, how you would now make this a product. Right. Is. As I indicated right, the way we do products for different segments is. And Microsoft segment or an Apple segment or a Sony segment are fundamentally different segments. Why? Because the supply chain is different.

Yeah, yeah,

I I need to get a different set of partners if I want to produce a product from Microsoft versus Apple, right? You do not work together. So as I indicated, the engineering is typically only 5% of the cost of a of a product. Right. Yeah, so.95% is allocated to a particular vendor like Sony or Microsoft. Or I should say the 95% of the costs associated with delivering product to a customer. Is spent on a particular vertical. And you cannot reuse that in another vertical, right? So. The validation of whether or not this is a good idea. Is it needs to. One be commensurate to the cost, right? So say that say that it takes $15,000 to build this physical product. With this idea that 20 percent 20x multiplier comes in from the actual delivery of a product to a marketplace, we would have $300,000 in order to build this product, right? So what's the what's the total market for? Muscular dystrophy kids in in Holland. How many are there?

Oh, I'm not sure.

Is it measured in the thousands or is it measured in the 10s of thousands? I think thousands, thousands. OK, so say it's $1000.Uh, so if it takes you 300,000 to deliver? And you want a 50% margin, you would need this to sell at. Uh 300 + 50 so $600.Right. So the controller would be $600.00 well $600.00 when your Xbox is how much does an Xbox

like 400.

Yeah, exactly. So $600.00 for the controller.

Yeah.

And idea OK so. This is not. Commensurate with the market opportunity, so this means that it really would be either a prototyping exercise so you don't have to spend the $3000 because you're not actually building a market, right, you're just building a prototype. Now you're measuring it, OK. We need to cover the cost for time and material, but then we're going to get information about how the market. Validates or or tests or agrees with the requirements analysis and the delivery of that in the form of a particular prototype, right? Umm. So this is sort of a simple calculation to sort of say, OK, well, what is this engineering really about? Is it prototyping? Is it requirements discovery? Is it validation of a particular design hypothesis, right?

Yeah. Yeah.

This is the question that. Umm. Really should be sort of part of the. Uh, what is the? What is the gate for success? Right. What's the gate that says OK, we're done.

Yeah, I think that would be prototyping,

OK, because-

makes perfect sense.

Yeah, the design hypothesis has already been concluded, so to say that with the controller itself. Like the the design for the controller is already oh optimal up to this point. Yeah. Requirement confirmation is. Still. A bit of a Gray area. Where the requirements have simply been settled upon and it is not thought about whether something else should be added. And yeah, I think this whole product or whole project in itself was based around prototyping, but took a really dirty turn with the reverse engineering requirements.

Yeah.

So in in that regard, I also think that, yeah, in terms of validation. Yeah, it's not feasible. For the PlayStation development. Because of the potential illegal territory and human interface device protocols that you would need to be mimicking or creating your own with a lot of time and effort into that,

yeah. Yeah, it it's exactly that's how I would summarize it too,

yeah.

And the university should never have allowed you to. Reverse engineer a an industrial product. That that is, uh, questionable- Questionable behavior,

yeah. Yeah. No, not now that you, I, I didn't even think about that up until this point. I thought, you know, like fun building stuff. Innocent me. But actually if I would have continued with this, it could have gone very wrong I think.

Yeah, it's it's just not appropriate. That's the.

Yeah, yeah.

Um, so when I direct engineers, right? We. We are very. Cognizant of. The cost of the time of that engineer. Right. So if I, if I look at? Delegating a task,

yeah.

Then there is 15 different ways to do that. And I'm looking at because it always is the time of the engineer that is the most expensive piece. Right.

Hmm.

So. The evaluation is OK. Do I burn the time of an engineer? Or do I buy a product or a spec? Yeah. That is an you know, before you delegate work. Umm. That's that's the the the the question you ask as a program manager. Um. There's two things that. Or the the reasons for reverse engineering. Is to educate right? How did this work? But you tend not to reverse engineer specifications because that is. Busy work.

Yeah,

right. Because if you simply buy the spec, you get a much better product. Because you actually are part of the supply chain, so reverse engineering is really. How in the world did they do this right kind of thing and can be done in software that can be done in hardware that can be done with and as I mentioned these interposers right logic analyzers. I mean if you really want to understand. What you don't know? Then you reverse engineer right? For educational purposes, yes, you. You're asking the question, how did they do this? And then you, you know, you you you try to. Devise an experiment that allows you to build that understanding.

Yeah, yeah,

right. And as you sort of look at. Uh. Product ecosystems around Microsoft, around Apple, around Sony, then in the EULA's you sign right there. It typically has a lot of language about thou shall not reverse engineer this stuff.

Yeah,

right. And the reason for that is obvious is they they just spend a billion dollars building this marketplace. They don't want. They don't want to lose control over who interfaces with these products. Right.

Yeah.

So for example, Tesla. It's a great example, Tesla. Um, if you and I would start a garage that services Teslas, right? You're not allowed to. Because Tesla basically says no third party will access our our our product.

Wow. OK, well, you know, I'm not surprised. Yeah,

yeah. And and and the fundamental reason is typically. Uh. Quality control. Right, Sony sells.50 million of these units. And if anybody and his uncle would be able to plug stuff into that and something go wrong, yeah, we're gonna call. They're gonna call Sony.

Yeah. Yeah.

And Sony will then get a bad Rep. The Sony brand would get a bad Rep because somebody else. Had poor quality componentry,

right, because, and that's why they keep full control of their ecosystem to circumvent such liability concerns.

Yeah, exactly. So. Umm. So that's why reverse engineering is a, you know, kind of it's not appropriate, certainly not for university as an educational system. Yeah. Yeah. I mean, if you're an particle physicist or an experimental physicist, by all means. But you're not. You're a product developer.

Yeah. Yeah.

*So it is odd. So anyways, we have ******* enough about that problem. Umm. So back to. Prototyping into something that you learn about with respect to the product, requirements for disability, human interface devices for disabilities.*

Yeah,

right. And. That indeed it does require an an A full circle. Right. You you built the product or you prototype, the product you let a a sample of the customer target customer base.

Yeah.

Use it and give you feedback. Yeah. So I do think you wanna. You wanna close that loop from an A pedagogical point of view. Right. You wanna build this system and have somebody play with it and

yeah.

I would argue it is irrelevant if that's a PlayStation 4.Versus an Xbox or a PC? So I would say, well, if you wanna get to the end to the finish line. Go and find the shortest path to that finish line. If the PC version is working, then present that to. Half a dozen of your muscular dystrophy buddies or not buddies, but your your customers and let them play with it and say I like this. I hate this. Because I just from a. Just from a closing the loop point of view, right and now I'm really talking about what is the educational opportunity here is you want to go through the whole cycle?

Yeah, yeah.

You cannot walk away midstream. Um. Because you'll run out of time. You know that.

Yeah, yeah, yeah. So because I I still want to develop this and we've also set-up a little group with the with other kinds of projects, so together we form a sort of assistive technology labs. That guy we would, for example tackle such solutions in a group of students together. So for example, there's also this one person with locked in syndrome where he can only move his eyes. Um, so we're trying to see if we can develop a robot that he can drive simply with an eye tracker to explore the world, which is. Yeah. Something he can't do. So project like that is what we would like to try and develop. But I think this is a great learning school about. You know, picking your battles.

Yeah, true.

But yeah, in terms of validation and yeah, closing the loop, really trying to see what you would like to prototype I. Which you will agree that I should be dropping the PlayStation implementation and try and see if I can get it to work on the PC and see how that works out with the features.

I agree with that statement for the simple reason that. You should- unless they give you the spec. For the PlayStation 4.I think it's a fool's errand. To try to. Um. Yeah, to try to build a system that that is PlayStation 4 compatible without without the right inputs. So. And then secondly, right, what is the? What's the ROI of your time with respect to the learning that still needs to take place?

Yeah, yeah.

That is not reverse engineering PlayStation 4 specs. It is product development presenting a prototype recording, customer feedback, and articulating the next step for improvement, right. That is the learning that's been taking place yet,

yeah. Yeah. No, I agree. That's. Uhhh, Yeah, it happened the way it happened so.

Yeah. And you've learned something from it.

No, definitely, definitely.

Appendix F

Vind je het goed dat dit wordt opgenomen zodat ik het in mijn rapport kan zetten?

oh ja dat is goed hoor.

ja. Mooi.

Hopelijk kun je dan nog slagen.

Ja, nee, dat komt wel goed. Even kijken, dus de stand van zaken is dat ik het heb geprobeerd te maken voor de PlayStation 4.En de implementatie die jij van Edwin had gekregen. Dan gebruikt je twee van die losse controllertjes die je zo naar een soort van uit elkaar gehaalde PS4 controller gingen.

Ja, dat klopt

ja en dat werkte dus Alleen voor *Fall Guys* was dat?

Ja, maar bedoel je nu dat dat gene wat ik jou heb Laten zien ben die twee nunchucks?

Ja ja.

Ja. Dat werkt Alleen voor Fall Guys. Ja,

en dat werkte dus dat je een knopje op die controllers had om tussen menutjes te switchen, toch?

Ja en een andere knap op bijvoorbeeld kunnen doen.

Ja, en hoe werkte dat voor jou? Voor *Fall Guys*

ja, ik kwam daar op zich wel. Met Fall Guys kon ik dat wel, Maar ik kom wel een keer in andere spellen meer doen. En Ik had het idee dat er nog een knap. Wij moesten op meer mogelijkheden te hebben. En nu game, Ik ben een gewone controller dat ik met spellen kan afwisselen op ja, dat lukt mij nog wel,

want de joysticks konden die ook een normaal knopje nadoen, bijvoorbeeld dat je Als je de rechter joystick omhoog deed, dat het dan kruisje zou zijn.

Ja

oké. En wat voor knopjes zou je dan nog meer willen erop?

Ja, Het was gewoon dat ik dan als die vast heb, dat ik aan de bovenkant ook nog wat wijsvinger nog iets zou kunnen inklikken

waar je normaal L1 en R1 hebt zitten

Ja of R2 en L2

ja oke. Ja, dan heb je toch iets meer functionaliteit inderdaad?

Want in Nijmegen ben ik nu ook nog met met iets bezig. En dan kan ik met mijn hoofd opzij bewegen om ook een knop. Te activeren of met mn voeten, dan krijgen we ook meer mogelijkheden,

Is dat met Jordi toevallig?

Bij Jordy van Heeswijk? Ja

ja. Ja, ik werk ook deels met haar om te kijken hoe en wat voor oplossingen Ik kan maken. Ik heb nu ook zo een Xbox adaptive controller.

Ja, die heb ik ook besteld, Maar ik had eerst dat met een Combinatie met zo een Titan ken je die.

Ik heb hier een andere soort liggen. Dit is een Cronus Max. Dat is zo een zelfde soort-

maar wat doet dat eigenlijk? Die Titan dat vroeg ik me af

nou, die Titan

die combinatie-

dus zeg maar, Dit is gewoon zo een USBtje en dit stop je zo in je PlayStation 4 waar je normaal je controller zegmaar zou inpluggen en dan hier aan het einde, Hier stop jij dan jouw eigen controller in?

Want Dat is dan eigenlijk gewoon om die adaptive controller te koppelen met de PlayStation controller.

Hier zit dus een soort van. Ja, voor mij is het ook echt magie in wat ervoor zorgt dat een willekeurige controller begrepen kan worden door de PlayStation.

Ja,

en Dat is. Ik heb dus deze dingetjes ook proberen te gebruiken om mijn eigen implementatie werkend te krijgen.

Ja,

Het is Alleen dat het zelf komt een controller maken ongelooflijk lastig is, Omdat je het echt heel. Ja laag level moet coderen met heel veel detail. En daarvoor ben ik dus gaan, want bijvoorbeeld deze controller. Dit is gewoon een soort van standaard. Computer controller ja die kan je dus zo in die controller converter zo een Cronus Max van Titan Two, dan plug je die er zo in en dan stop je dit zo in je PS4 en dan zou je die kunnen gebruiken. Dus wat Ik heb geprobeerd is… Ja?

is die cronus Max is dat goedkoper of weet jij dat niet?

Oh, dat weet ik niet. Ik heb deze van een van NMA gaming geleend.

Oh ja, die die zijn er ook bij betrokken, ja, maar die moet ik zo meteen weer teruggeven. Ik kan dus wel even zoeken hoe duur die is. Ja, Ik heb een Titan Two wel besteld en die hebben we al, dus dat maakt toch niet meer uit.

Ja nee, want deze is ook zo te zien iets van € 90. ja tussen de 90 en de 100.

Maar die moet ik hebben, dus die heb ik besteld.

Ja. Nee,

want die waren bijna niet op voorraad

nee klopt die zijn echt heel lastig te vinden inderdaad.

Ik heb die op AliExpress besteld,

oh oké? Top ja dan duurt het ook eventjes voordat die Er is,

Maar ik heb die net voor de vakantie gekregen. Alleen het jammere is dat Jordy heeft In de schoolvakanties. Alles vrij.

Oké,

dus waarschijnlijk kunnen we dat vanaf september installeren,

ja. Oké. Ja, dus, dan zou je nog eventjes van je vakantie moeten genieten.

Ja, eigenlijk veel langer

ja. Kijken, maar ja, dus, want zeg maar met zo een Xbox adaptive controller. Ik kan ervoor zorgen dat de controller van Edwin dus die jij zo ja redelijk fijn vindt om mee te spelen.

Ja,

Ik kan er dus voor zorgen dat die zo een soort zelfde soort implementatie via de Xbox Adaptive controller werkt. En dan naar de PS4 gaat, waardoor je die controllertjes dus kan gebruiken voor. Allerlei soorten spellen, Dat is het idee.

Ja,

Het is Alleen dat met het project probeerde ik het zeg maar zo compact mogelijk te houden, zodat je niet allerlei soorten dingen moest kopen. Dus probeerde ik het zeg maar zelf. Ik probeerde, dan zeg maar zelf deze controller kijken wat voor signalen die stuurt en die signalen proberen na te bootsen. Maar Als ik dat uiteindelijk zou doen, dan zou dat een heel klein beetje illegaal zijn.

Ja,

Omdat je dan zeg maar gewoon een product letterlijk gekopieerd. Ja, dat klopt dus de nu dat je een Xbox adaptive controller hebt zou ik kunnen kijken of ik, want Ik heb zelf ook een Xbox adaptive controller van Jordy mogen lenen. Om te kijken of mijn implementatie daarmee zou werken.

Maar die moet ik nog na de vakantie bestellen, want die hebben ze wel op voorraad.

Ja mooi, want Ik ben daar nu dus mee bezig om te kijken van oké, Als ik vanuit mijn kleine. Kleine apparaatje, Als ik daaruit elk. Want weet je hoe die werkt de Xbox Adaptive controller?

Ja ongeveer wel, ja,

ja, zeg maar elk knopje en elke joystick die heeft zijn eigen draadje. Wat je met zo'n 3.5 mm jack Combineert en, Het is het idee om dan. Via zo, een micro controller dus zo een klein computer boardje dan zet je elk draadje van zo een knopje die van zo een Xbox adaptive controller komt in die controller.

Ja

die dan eigenlijk het denkwerk doet. En die neemt ook de input van jouw hand controllertjes. Maar het is Alleen dat daar heel veel echt heel iets van 25 draadjes. bij nodig zijn, Om dat Goed te krijgen, dus Het is nog even kijken of dat handig is.

En, hoe bedoel je het dan dat dat die die controllers van Edwin dat ik die dan gebruik in combinatie met mijn PlayStation controller of wat is eigenlijk de bedoeling?

Dan eigenlijk, Dan heb jij jouw hand controllertjes en die gaan dan naar een soort van computer boardje. En dat computer boardje, dat gaat dan naar de Xbox Adaptive controller. De Xbox adaptive controller gaat dan naar de Titan en die stuurt het signaal naar je PlayStation.

Ja,

dat zou in theorie moeten werken. Het is Alleen één heel groot pakket wat je dan krijgt Omdat je dan echt 25 best wel dikke kabels hebt. Dus dan zit je denk ik echt met zo een gigantische kabel. Maar het is nog even dus kijken hoe dat het handigst is om dat bij jou uiteindelijk te kunnen leggen Zonder dat het heel zwaar word.

Ja, want dan heb ik dadelijk dus eigenlijk twee opties op te kunnen gamen, dat waar ik met Jordy mee bezig ben met jullie.

Ja, want met Jordy ben je dan met een hoofd controller bezig?

Nee, daar heb ik gewoon de PlayStation controller op mijn hand, maar dan kan ik wat ik net zei, met mijn hoofd en voeten extra knoppen toevoegen. Dat ik nu niet zo goed bij kan,

oké Ja, Het is gewoon even kijken, denk ik, wat het ja niet per se wat het beste werkt, maar gewoon wat jou mogelijkheden zou bieden. Ja en dan Als het voor jou werkt, dan werkt het Misschien voor anderen ook en die krijgen dan ook de keuze daartussen, dus Het is inderdaad, ja

ja, dat is ook goed voor mijzelf ook op de kijken naar de toekomst, hoe ik het langst kan gamen.

Daarom ja. Ja, want het ik, Ik vind het wel jammer dat het project uiteindelijk niet is gelukt Omdat de ontwikkeling om direct op de PlayStation te coderen. Dat is. Dat is dus heel moeilijk en vrijwel niet te doen. Dus je hebt al die systemen daartussen nodig, helaas met zo een Titan two en zo een Xbox adaptive controller om iets werkend te krijgen. Maar ik neem aan dat jij dat zeg maar prima vindt Als we de controllertjes die jij in je handen krijgt zo licht mogelijk houden en dan maakt het niet uit hoeveel draadjes daarachter liggen.

nee dat, Maar ik kan niet uit. Nee,

oke, nee, want dan dan denk ik dat dat de oplossing wordt, dus. Ja, hoe zou jij dat vinden?

Ja ik ik, ik zou het fijn vinden Als ik, mij gaat het er eigenlijk vooral omdat ik in ieder geval kan blijven gamen zolang mogelijk. En op welke manier maakt dat niet zoveel uit Als het maar lukt?

Ja en. Dan kunnen we ook nog even kijken of we nog extra knopjes op die van Edwin kunnen designen.

Ja met de 3d printer neem ik aan.

ja ja dan dan passen we het model een beetje aan en dan zetten we er denk ik kijken hoe we die knopjes het beste daarop kunnen zetten. Maar ja, dat, dat zal dan nog wel even duren, want Ik weet niet zeker hoe Edwin zit In de zomervakantie, want We gaan hier nog wel mee verder. We gaan hier wel mee verder voor jou.

En ja voor anderen is dat ook goed.

Ja. Want met Edwin hebben we dus een soort Assistive Technology Lab opgezet voor allerlei soorten projecten zoals deze voor ja, een soort van speciale oplossingen. En dan gaan we in een team dat soort oplossingen aanpakken. Dus Misschien dat we dit controller ding dan best snel in dat team aanpakken of dat ik het zelf moet verder developen. Het is Alleen dat ik al deze spullen zeg maar moet lenen en ook weer terug moet brengen,

ja dat maakt het wat lastiger.

ja, inderdaad, dus ik moet.

Ik heb graag ook de spullen zelf de Titan en de Xbox adaptive controller

ja dus bijvoorbeeld Als ik, want Ik kan dan gewoon iets maken en dan zou ik. Ik ben ja, dan zou ik bijvoorbeeld een keer bij jou kunnen langskomen om dat te testen of zo.

Ja, dat wel, maar, het werkt Alleen lastig dat ik zo niet naar je toe zou kunnen brengen. Of wat dan ook,

nee dan zou ik gewoon langskomen. Maar ik ga in ieder geval kijken of ik zelf ook gewoon aan die spullen kan komen. Zodat ik het zelf heb om te testen, want dat zou het een stuk makkelijker maken.

Ja, dat klopt.

Want. Het is ja voor de PlayStation is het heel moeilijk, maar voor de computer werkt het prima, zeg maar. Want in mijn computer heb je geen rare protocollen nodig waar je omheen moet coderen. Dus heb jij enige mogelijkheid om op een computer te gamen?

Ja momenteel heb ik geen pc meer

oké,

maar ja ook vooral Omdat ik het altijd met de PlayStation doe. Maar zou je bijvoorbeeld dan ook zo PlayStation controller aan de computer te kunnen koppelen?

Zo, een speciale controller.

Nee gewoon. gewoon een standaard PlayStation controller.

Ja, de gewone standaard PS4 controller, die kan je inderdaad ook op een computer koppelen. Ja.

En Ik ben er dan ook aan het kijken, zeg maar of we ook uiteindelijk een oplossing kunnen vinden dat ik ook op de PS5 zou kunnen gamen.

Ja, Als het goed is, zou dat met zo een Titan One en zo een Xbox adaptive controller moeten werken.

Ja, Ik heb mijn Titan Two zelf maar.

Oh Titan Two, wacht, haal ik ze door elkaar. Ja, sorry, nee, ik bedoelde de Titan Two. Ja. Ja, dus dat zou dan de de oplossing zijn waardoor jij ook gewoon op de PS 5 kan blijven gamen.

En ja, want nu is het probleem. De PlayStation 5 controller. Die is iets anders dan die van de 4 die is wat zwaarder en breder. Daar heb ik toch meer moeite mee.

Oh juist, Dat is die witte controller, toch?

Ja, die heb je volgens mij ook in het zwart

in ieder geval iets puntiger is die,

ja.

Ja, Misschien dat je zo een PS4 controller dan ook in zo een Titan Two kan steken en dat hij dan op de PS5 zou werken.

Ja, dat zou kunnen, dat weet ik niet.

Ja, Maar dat is te proberen mijn ieder geval. Als we dan die speciale controllertjes op een Xbox adaptive controller te werken krijgen, dan zou dat ook op de PlayStation 5 moeten werken. Dus ik zal even kijken met. Jordy en Edwin. Hoe we dat dan gaan aanpakken? Ja, ja, zeg Maar we proberen het zo simpel mogelijk te houden Zonder dat je extra dingen moest kopen, maar. Ik denk dat er geen andere mogelijkheid is dan dat je die dingen moet kopen om dat te maken.

En daarom hebben we die ook aangeschaft.

Ja, en Ik ben blij dat je dan die dingen hebt aangeschaft, want dan betekent het dat we ze jou zo meteen. Als het goed is gewoon aan die controller kunnen zetten. Dus ja, als het dan. Uiteindelijk ook met de PlayStation 5 lukt dan dan wil ik die ook wel kopen.

Ik heb Alleen zelf en de Universiteit ook niet geen PlayStation 5 om op te testen.

Ja ik ook niet.

Heeft Jordy, die toevallig, weet je dat?

Nee volgens mij ook niet

zijn die nog steeds zo moeilijk-

Dat is moeilijk aan te komen, ja.

Nog steeds joh.

Ja nog altijd,

Wauw. Ja, Dat is ook een probleem, want hoe zit het met jouw vrienden? Zitten die dan allemaal op de PlayStation 5? Al of

Ik heb Ja, Ik heb wel een paar vrienden die de 5 hebben, maar niet allemaal. Ja, Maar dat is dan wel

uiteindelijk is het dan wel het idee dat jullie allemaal naar de PlayStation 5 gaan dan.

Ja dat hopen we wel, ja.

Ja. Nee, dat snap ik want. Ja, Als het goed is dan die controller of nou die Titan two. Die zou zo dat signaal moeten omvormen dat die PlayStation 5 dat ook gewoon begrijpt. Dus Als het goed is zou je daar geen problemen mee moeten hebben. Maar ja, Ik kan dat niet garanderen. Maar ja, Het is in ieder geval dat je weet dat we dus nog steeds mee bezig zijn.

Ja, Dat is al goed.

Ja, en? Want. Ik heb me dit nu, zeg maar helemaal in verdiept en Ik weet nu wat wel en niet kan. En mijn kennis zal zo meteen dus naar een nieuwe student gaan die hier volgend schooljaar ook. Verder mee gaat of verder mee probeert te gaan.

Maar jij zou daar ook mee verder willen gaan?

Ja ja, nee Ik ga daar sowieso mee verder.

Oh, je bedoelt meer dat jij dat je daar de kracht een beetje kunt bundelen?

Ja, inderdaad

met meer handen kan werken

precies. Ja dat ik niet alles alles zelf moet doen, Maar dat dat we ja Misschien taken kunnen verdelen of. Dat soort dingen dat het Misschien iets sneller zou gaan. Hopelijk. Even kijken.is er zo verder nog iets wat jij zou willen weten.

Nou, Ik heb eigenlijk mijn vraag al gesteld.

Ja nee, want Ik heb jou geupdate over de situatie. Vertelt wat We gaan doen? wat de plannen zijn.

Want, want ik dacht dat dit meer een een interview was, maar toch niet zo.

Zo nee, nee, Het was meer eventjes jou een update geven over de situatie. En wat jij van het concept zeg maar vond. Ja en nou jij vindt dat concept fijn. Je hoopte Alleen op meer knopjes om dat nog beter te maken en dat je het op. Andere spellen kan toepassen.

Ja, daar kan ik dan altijd nog kijken wat ik zelf de fijnste manier af wil.

Ja ja, want dat dat menu switchen, hè? Is dat makkelijk voor jou eigenlijk om die knopjes te onthouden? Welke er In het menu zitten?

Ja tot nu toe wel. Maar ik speel dat nu eigenlijk weinig nog Omdat ik meestal game ik nu met de gewone controle, Omdat ik welke kant wisselen tussen de spellen. Dan lukt dat wel, dus ik gebruik hem eigenlijk momenteel niet

en toen je hem nog gebruikte. Hoeveel menutjes had je toen?

toen ik het nog gebruikte. Ik had 3 verschillende menus, maar Fall Guys gebruikte ik er twee.

Oké. En want. Is dat gewoon makkelijk voor jou onthouden Omdat je daar aan gewend bent? Of zo heb jij een soort van vergelijkbaar systeem?

Nou ik, mij viel wel op Edwin heeft dat toen. Ook geprobeerd. Maar ik heb dat veel sneller onder de knie. Nee ik ja ik. Ook door het te doen, maar op zich had ik het wel snel door.

Ja. Even kijken hoor.

Maar is er nog iets wat jij wilt weten?

Het beeld loopt een klein beetje vast. Ja. Even kijken ja, Ik ben er weer terug. Sorry wat. Wat zei je?

Of jij nog vragen had?

Nee ik, Ik heb alles gesteld wat ik wou stellen en. Ja, jij weet nu dus de situatie. We gaan kijken hoe dat zit met een Xbox adaptive controller en zo een Titan. En. Dan ga ik dus bezig met dat soort van. Ja, dat Fall Guys systeem eigenlijk zodanig maken dat dat werkt met andere spellen ook.

Ja Dat is, Dat is duidelijk

en het idee is dan ook om met een SD kaartje per spel menutjes te kunnen maken.

Ja dat ieder spel zo de knoppen heeft hoe ik ze zou willen hebben.

Ja ja. Het is Alleen nog wel even de vraag. Ja, want dat gaat best technisch moeten zijn met het formaat waarin dat op dat SD kaartje komt te staan.

Ja

dus Misschien dat dat eerst nog heel ruw in normale tekst getypt moet worden, Maar dat daar later Misschien een soort software voor zou komen. Dat dat iets logischer gaat?

Ja.

Maar in ieder geval dat jij dus per spel die menutjes hebt en dat dan via een Xbox adaptive controller kan aansturen. En.

Ja

is wel vet dat je dan ook een hoofd controller hebt.

Ja en met de voeten dat ik zo een beetje optil hebben we ook met twee extra knoppen.

Oké. En, Dat is wel handig voor iets van ja, Ik weet niet een FIFA ofzo, want FIFA heeft best veel combinatie knoppen.

Ja ook bij Fortnite.

Ah Oke. Met dat bouwen.

Dat heb ik daar al getest

Ja. Nou, Dat is wel cool, ja? En en dat werkt dan ook met de PlayStation.

Ja, daar moest ik die Titan Two. Die adapter controller dan kon ik dat ook.

Oké, want werkte dat ook met zo een Xbox adaptive controller of dat nog niet?

Ja

oké ja.

Die moest ik allebei hebben, ja?

Nee, fijn dat fijn dat je die hebt aangeschaft, want dat geeft mij ook heel veel meer mogelijkheid.

Ja, dan kun je dan kunnen we wat meer?

Ja, precies. Oké, ik zou zeggen, Ik heb jou lang genoeg opgehouden en. Jij moet lekker van Frankrijk gaan genieten. Dan duik ik hier mijn rapport weer in.

Ja, dat zal wel lukken, maar dan wens ik jou veel success met het project.

Dankjewel, dat komt helemaal goed en ja, hopelijk tot gauw weer door je ouders te groeten.

zal ik doen, en hou me op de hoogte

zal ik doen, oké, doei Marc.

Bedankt daag.

Works Cited

[1] E.-M. Strehle and V. Straub, "Recent advances in the management of Duchenne Muscular Dystrophy," *Archives of Disease in Childhood,* vol. II, no. 100, pp. 1173-1177, 2015.

[2] J. Broomfield, M. Hill, M. Guglieri, M. Crowther and K. Abrams, "Life Expectancy in Duchenne Muscular Dystrophy: Reproduced Individual Patient Data Meta-analysis," *Neurology,* vol. 97, no. 23, pp. e2304-e2314, 2021.

[3] Y. Chunzhi, M. Ningling, Y. Chifu, G. Hao, H. Jiahong, G. Hefu, L. Xiang and J. Feng, "A novel automatic diagnosis based physical therapy for duchenne muscular dystrophy children," in *3rd IEEE International Conference on Data Science in Cyberspace*, Guangzhou, 2018.

[4] M. M. H. P. Janssen, J. Horstik, P. Klap and I. J. M. de Groot, "Feasibility and effectiveness of a novel dynamic arm suport in persons with spinal muscular atrophy and duchenne muscular dystrophy," *Journal of NeuroEngineering and Rehabilitation,* vol. 18, no. 1, p. 84, 2021.

[5] K. Nizamis, W. Schutte, J. J. Grutters, J. Goseling, N. H. M. Rijken and B. F. J. M. Koopman, "Evaluation of the cognitive-motor performance of adults with Duchenne Muscular Dystrophy in a hand-related task," *PLoS ONE,* vol. 15, no. 1, 2020.

[6] "DUALSHOCK 4 wireless controller," Sony, [Online]. Available: https://manuals.playstation.net/document/en/ps4/basic/pn_controller.html. [Accessed 26 March 2022].

[7] J. Beeston, C. Power, P. Cairns and M. Barlet, "Accessible Player Experiences (APX): The Players," *Springer, Cham,* vol. 10896, pp. 245-253, 2018.

[8] D. Maggiorini, M. Granato, L. A. Ripamonti, M. Marras and D. Gadia, "Evolution of Game Controllers: Toward the Support of Gamers with Physical Disabilities.," *Communications in Computer and Information Science,* vol. 654, pp. 66-89, 2017.

[9] "USB Component, HID: Human Interface Class," keil, [Online]. Available: https://www.keil.com/pack/doc/mw/USB/html/group__usbd__hid_functions.html#details. [Accessed 28 April 2022].

[10] F. Iacopetti, L. Fanucci, R. Roncella, D. Giusti and A. Scebba, "Game Console Controller Interface for People with Disability," in *International Conference on Complex, Intelligent and Software Intensive Systems*, Pisa, 2008.

[11] M. Hassan, Y. Shimizu, Y. Hada and K. Suzuki, "Joy-Pros: A Gaming Prosthesis to Enable Para-Esports for Persons With Upper Limb Deficiencies," *IEEE Access,* vol. 10, pp. 18933-18943, 2022.

[12] Y. Garrido, A. Marco, J. Segura, T. Blanco and R. Casas, "Accessible Gaming through Mainstreaming Kinetic Controller," in *Springer*, Berlin, 2009.

[13] N. Villar, K. M. Gilleade, D. Ramdunyellis and H. Gellersen, "The VoodooIO gaming kit: a real-time adaptable gaming controller," *Computers in Entertainment,* vol. 5, no. 3, p. 7, 2007.

[14] Microsoft, "Xbox Adaptive Controller - Game your way.," Microsoft, 4 September 2018. [Online]. Available: https://www.xbox.com/en-US/accessories/controllers/xbox-adaptive-controller. [Accessed 28 April 2022].

[15] L. Heutink, M. Janser, Y. van Den Elzen, D. van Der Pijl and I. J. de Groot, "Virtual Reality Computer Gaming with Dynamic Arm Support in Boys with Duchenne Muscular Dystrophy," *Journal of Neuromuscular Diseases,* vol. 5, no. 3, pp. 359-372, 2018.

[16] L. P. Lowes, L. N. Alfano, R. Crawfis, K. Berry, H. Yin, I. Dvorchik, K. M. Flanigan and J. R. Mendell, "Reliability and validity of ACTIVE-seated: an outcome in dystrophinopathy," *Muscle & Nerve,* vol. 52, no. 3, pp. 356-362, 2015.

[17] B. Schabron, J. Desai and Y. Yihun, "Wheelchair-mounted upper limb robotic exoskeleton with adaptive controller for activities of daily living," *Sensors,* vol. 21, no. 17, p. 5738, 2021.

[18] R. Ossmann, D. Thaller, G. Nussbaum, C. Veigl and C. Weiß, "Making the PlayStation 3 accessible with AsTeRICS," in *Computers Helping People with Special Needs*, Linz, 2012.

[19] "Titan 2, Basic Usage Overview," ConsoleTuner, [Online]. Available: https://www.consoletuner.com/products/titan-two/. [Accessed 28 April 2022].

[20] "Titan One, It's time to unleach the power!," ConsoleTuner, [Online]. Available: https://www.consoletuner.com/products/titan-one/. [Accessed 28 April 2022].

[21] lonnwolf031, "Splitcontroller," 23 September 2021. [Online]. Available: https://github.com/lonnwolf031/splitcontroller. [Accessed 5 July 2022].

[22] edwindertien, "splitcontroller," 8 May 2021. [Online]. Available: https://github.com/edwindertien/splitcontroller. [Accessed 5 July 2022].

[23] psdevwiki, "DualShock 3," 27 June 2022. [Online]. Available: https://www.psdevwiki.com/ps3/DualShock_3. [Accessed 5 July 2022].

[24] D. Geuer, "Controller Icons," 2020. [Online]. Available: https://www.artstation.com/artwork/0nXOWw. [Accessed 5 July 2022].

[25] PS4 Developer Wiki, "DS4-USB," PS4 Developer Wiki, 5 September 2021. [Online]. Available: https://www.psdevwiki.com/ps4/DS4-USB. [Accessed 30 June 2022].

[26] Electronic Arts, "Accessibility Settings FIFA 21 for PS4," EA, [Online]. Available: https://www.ea.com/able/resources/fifa/fifa-21/ps4/accessibility-settings. [Accessed 5 July 2022].

[27] R. Murphy, "FIFA 21 Controls: Attacking, Defending & Goalkeeping on PlayStation, Xbox & PC," Goal, 24 December 2020. [Online]. Available: https://www.goal.com/en/news/fifa-21-controls-attacking-defending-goalkeeping-playstation/1dgx1kuk9i6sn1bpumurztbvhg. [Accessed 5 July 2022].

[28] dogtopus, "Licensed PS4 Controller "Security Chip" Secure Element," 11 February 2022. [Online]. Available: https://gist.github.com/dogtopus/dae307c7773e792150990a06e79583d0. [Accessed 29 June 2022].

[29] PJRC, "Teensy 4.1 Development Board," PJRC, [Online]. Available: https://www.pjrc.com/store/teensy41.html. [Accessed 18 July 2022].

[30] P. Stoffregen, "CoreMark - CPU Performance Benchmark," 21 Aug 2019. [Online]. Available: https://github.com/PaulStoffregen/CoreMark. [Accessed 5 July 2022].

[31] Microsoft, "Comparison of XInput and DirectInput features," Microsoft, 8 February 2022. [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/xinput/xinput-and-directinput. [Accessed 30 June 2022].

[32] Arduino, "Arduino IDE 1 Portable Installation," Arduino, 26 June 2022. [Online]. Available: https://docs.arduino.cc/software/ide-v1/tutorials/PortableIDE. [Accessed 30 June 2022].

[33] Arduino, "Software," Arduino, [Online]. Available: https://www.arduino.cc/en/software. [Accessed 30 June 2022].

[34] PJRC, "Teensyduino," PJRC, [Online]. Available: https://www.pjrc.com/teensy/teensyduino.html. [Accessed 30 June 2022].

[35] PJRC, "Download Teensyduino, Version 1.56," PJRC, [Online]. Available: https://www.pjrc.com/teensy/td_download.html. [Accessed 30 June 2022].

[36] PJRC, "Using USB Joystick," PJRC, [Online]. Available: https://www.pjrc.com/teensy/td_joystick.html. [Accessed 30 June 2022].

[37] D. Madison, "Arduino XInput Library," 30 December 2021. [Online]. Available: https://github.com/dmadison/ArduinoXInput. [Accessed 30 June 2022].

[38] D. Madison, "Teensy XInput USB Mode," 30 December 2021. [Online]. Available: https://github.com/dmadison/ArduinoXInput_Teensy. [Accessed 30 June 2022].

[39] K. Hamaluik, "Making A Custom Teensy3 HID Joystick," 26 October 2013. [Online]. Available: https://blog.hamaluik.ca/posts/making-a-custom-teensy3-hid-joystick/. [Accessed 30 June 2022].

[40] noycebru, "How to create your own Xbox 360 USB PC controller with Aduino XInput Library," 4 May 2020. [Online]. Available: https://youtu.be/61U-S7_XTDw. [Accessed 30 June 2022].

[41] noycebru, "3D Printed PC Controller," 17 December 2020. [Online]. Available: https://youtu.be/2vhEw8xGN00. [Accessed 30 June 2022].

[42] D. Madison, "XInput.h," 10 January 2021. [Online]. Available: https://github.com/dmadison/ArduinoXInput/blob/master/src/XInput.h. [Accessed 30 June 2022].

[43] D. Madison, "How to Emulate an Xbox Controller with Arduino (XInput)," 19 March 2019. [Online]. Available: https://www.partsnotincluded.com/how-to-emulate-an-xbox-controller-with-arduino-xinput/. [Accessed 30 June 2022].

[44] "Internet Archive WayBackMachine," Internet Archive, [Online]. Available: https://archive.org/web/. [Accessed 30 June 2022].

[45] Eleccelerator, "DualShock 4," Eleccelerator, 15 January 2015. [Online]. Available: https://web.archive.org/web/20210521113252/http://eleccelerator.com/wiki/index.php?title=DualShock_4. [Accessed 30 June 2022].

[46] johndrinkwater, "dualshock-research," 29 November 2013. [Online]. Available: https://gist.github.com/johndrinkwater/7708901. [Accessed 30 June 2022].

[47] Gamepad Tester, "Gamepad Tester and Debugger," Humanbenchmark, [Online]. Available: https://gamepad-tester.com/. [Accessed 30 June 2022].

[48] PJRC, "Teensy 3.5 Development Board," [Online]. Available: https://www.pjrc.com/store/teensy35.html. [Accessed 5 July 2022].

[49] Console Tuner, "Titan Two," Console Tuner, [Online]. Available: https://www.consoletuner.com/products/titan-two/. [Accessed 28 06 2022].

[50] CronusMAX, "Getting to know your CronusMAX PLUS," CronusMAX PLUS, [Online]. Available: https://www.cronusmax.com/manual/getting_to_know_your_cronusmax_plus.htm. [Accessed 29 6 2022].

[51] edwindertien, "Splitcontroller," 8 May 2021. [Online]. Available: https://github.com/edwindertien/splitcontroller. [Accessed 5 July 2022].