Western University Faculty of Engineering
Studio Section 21 – Yue Zhou
Team Identifier: 21Movie
Team Members: Jahangir Abdullayev, Evan Cowan, Amit Altman, Jagbir Brar, Shajith Shajahan, Talha Kunwar

PINE-ABLE ™: For your movies

Design Description: Providing a plug-and-run custom streaming solution with minimal effort.

Figure 1 Final Product Disconnected

The 1-Page Guide of How to Use The PINE-ABLE ™

Requirements:

- Rename a USB drive to “21MOVIE” *(Capitalization-sensitive)*
- Prepare the USB drive with your movies in a folder named “movies” *(Capitalization-sensitive)*
- In another folder on the USB drive named “thumbnails”, prepare image posters named the same as their corresponding movie in the “movies” folder *(Capitalization-sensitive)*
- Have a monitor with audio ready and powered on
- A comfy environment for the movie bingeing that is to now take place!

First Set-up:

1. Plug your USB drive into the only USB port on the side of the case of the PINE-ABLE ™
2. Plug the PINE-ABLE ™’s HDMI cable into your desired viewing device.
3. Make sure that viewing device is plugged in to power and is turned on.
4. To power up the PINE-ABLE ™, simply plug in its power cable to an outlet or equivalent power source.
5. On your viewing device, you will see an L logo along with repeated texts, console commands and a blue and black screen. **That is perfectly normal.**
6. From there, the PINE-ABLE ™ custom UI will launch and will enter a 50 second loading screen with a countdown and the PINE-ABLE ™ logo. **Please wait patiently.**
7. Once the countdown ends, you will see a movie list present.
8. Your movie experience is now ready to be enjoyed.
9. **Unplug the PINE-ABLE ™ from power when done the session.**

Repeated Session:

1. Power up the PINE-ABLE ™ by plugging in its power cable to an outlet or equivalent power source.
2. On your viewing device, you will see an L logo along with repeated texts, console commands and a blue and black screen. **That is perfectly normal.**
3. From there, the PINE-ABLE ™ custom UI will launch and will enter a 50 second loading screen with a countdown and the PINE-ABLE ™ logo. **Please wait patiently.**
4. Once the countdown ends, you will see a movie list present.
5. Your movie experience is now ready to be enjoyed.
6. **Unplug the PINE-ABLE ™ from power when done the session.**

Controls:

- **While in movie listing / main menu:**
  - **Red button:** Opens and plays movie in the middle of the list (“Currently Selected”).
  - **White Button:** Scrolls to next movie in list (to the Right of the “Currently Selected”).
- **While movie is already being played in full screen:**
  - **Red button:** Goes back to movie listing / main menu.
  - **White button:** Pauses / un-pauses the currently playing movie.

1 Need / Challenge

1.1 Need Statement

Legend: The User The Need The Insight

The impaired user has troubles with mobility, speaking, and understanding that causes an inability to perform basic actions, but requires some independence for selecting and watching movies so that she can minimally assist herself without the need of helpers.

1.2 Our Solution; an Overview

Our final working solution to the needs of the client consists of a 2-part device that plays movies at the press of two simple buttons. The first part is the remote itself, set up in a 3D printed case, and two simple arcade buttons soldered to a ribbon wire. The second part is the single board computer, set up with another 3D printed case, and 3 connections to it: USB power, HDMI signal output, and GPIO pin connection to the remote's ribbon wire. The single board computer is programmed to recognize the GPIO pins (and therefore the remote) as two inputs, which are then converted into simulated keystrokes for the computer. The computer's software system is set up to a regular Raspbian (like Debian) operating system, with Python scripts to launch a web application on a regular launch, including a script to read directories for movie downloads and posters. The web application (after booting and receiving the information of the locations of the movies) presents a screen of three movie posters at a time, and with button interaction, the user can interact with the application to scroll through movie listings and play movies. When the movie player is in session, the button inputs are registered in the software as different actions, with a pause/continue button, and an exit button. On exit, the user is returned to the movie selection menu.

2 Final Design Documentation

2.1 Movie User-Interface

2.1.1.1 Frontend

The user interface is developed to show large bold colours that are easily identifiable from far away. The design features three large movie posters: A bright centered poster, and two darker posters on each side flanking the middle movie. This highlighted center poster is the movie that will be played when selected, with the side posers showing what movie is next in the order, and what movie is before. Refer to 7.3 USER INTERFACE for a screenshot of the visual design.

The interface is created using a static HTML file (*index.html*) with custom JavaScript (*script.js*) and CSS (*style.css*) that opens upon start-up of the central processing unit. The HTML includes three empty images with undefined sources, which are then populated upon reading the *movies.js* JavaScript file. The HTML also includes a hidden video player, which is hidden and revealed upon selecting and deselecting the movie.

The JavaScript code detects the input of the controller by detecting whether the keys "A" or "B" are pressed, with the left button on the controller being mapped to "A", and the right button being "B". The code also detects how fast the user presses the buttons and prevents double clicking by adding a 250ms delay to each button press, as well as preventing the pressing two buttons at the same time, with only the first button pressed functioning within the timeframe.

Upon advancing the selection, the movie index will increase until the maximum number of movies is reached where it will reset to 0. This controls displaying the images, with the source (*src)* of the images being changed

every button press to the new movie index. When exiting a movie, the array is updated with the timestamp from the currently playing movie, so it can resume from the saved timestamp upon reselection. A full breakdown of the user interface controls and responses can be displayed as a flowchart, refer to 7.4 USER INTERFACE FLOWCHART.

The interface will also display a 50 second loading screen upon first launch, saving whether the device has already displayed the loading screen in the browser's local storage. Upon the countdown of the loading screen ending, the page will be refreshed, which allows for reloading of the *movies.js* file, which will have been updated by the backend code.

2.1.1.2 Backend

As previously stated, upon start-up of the central processing unit, the HTML will auto-launch and display the loading screen while the USB device properly mounts, and the backend python script runs. At the same time as the HTML file is opened, the shell file (*launcher.sh*) is run which starts the frontend launch python script (*frontend_launch_script.py)* and the pin-in GPIO read script. (gpio_read.py). The GPIO script detects any input from the controller buttons and binds them to the keys "A" and "B", which is used by the frontend JavaScript.

The launch script first waits 45 seconds for the USB device to properly mount, and then scans the movie directory and thumbnail directory on the USB drive. The script then saves that information to a new JavaScript array, in *movies.js.* This file includes an array of all the movies in the movie directory and their file type (Ex. mp4, m4a, mkv, etc.), as well as the play head location (The timestamp of where the user left off in the movie). The file also includes a Boolean variable of whether each movie includes a thumbnail image from the thumbnail directory.

If connected to the internet, the device will attempt to download posters for any movies without a thumbnail, and save them to the thumbnail folder, so that even if an internet connection is removed, the posters can be used in subsequent sessions. If a movie is present and a poster cannot be found or downloaded, a default image is used. Upon completion of the python script, the HTML will refresh from the loading screen and load the new movie data.

The full code for the frontend and backend can be viewed at:
https://github.com/JanikThePanic/ES1050_21Movie.

2.2 Design Process and Criteria

After our team was given our project description, we quickly developed an understanding of the issue at hand. Our client required a device that acts as a simplified remote, with a limited set of necessary functions. The solution must tackle this and must provide the client at the very least a working remote setup to semi-independently play movies. However, beyond that fact, we knew little to none about the wants and other needs of the client for the solution.

Our initial idea was to build a remote button connected to a single board computer, with our own programmed software implementation. This fixes the client's inability to use the DVD drives and DVD players we were informed about and provided the client the independency to choose a movie to watch.

Our first client meet that occurred a week into the beginning of the semester outlined the needs of the client. The client cleared confusion about the problem being tackled. She explained her wants, including a remote that the user can use with her impairments, which included minimal functions for an easier understanding of the remote's functions, as the user's impairments prevent her from comprehending the regular commercial

universal remotes and all the functions. The remote had to be built with large buttons, as the user has trouble with her coordination. Our team then proceeded to ask her questions about the solution, and any preferences she had. We discovered the client prefers wireless capabilities for portability purposes and wants the device to be accessible. If developing software (like our solution), the UI must be basic and only have a couple options for selecting which movies in a scrolling fashion, as too many choices may overwhelm the user. The device must also be “plug-and-play”, giving the client the ability to plug in the device and begin playing movies immediately without any advanced debugging. And lastly, the client requested the device does not overload the helpers with required troubleshooting for the use of the device, and therefore the solution must be stable in the hardware and software field.

Immediately following the client meeting, our team knew exactly how to tackle the problem at hand. We decided to tackle the problem with our first idea: to create an all-in-one device for the client that provides a movie player compatible for the user and a remote paired with it that is simple for the client to use. The idea was to begin programming the UI software to match a scrolling-like movie player and integrate a video player within it that plays the movies off the ripped DVDs (as in the downloaded movies off the DVDs). For the remote, we decided to approach a wired solution first, as the most important thing is for the device to work, and if we had time and budget left once complete, we could implement wireless connectivity.

The beginning stages of the creation of the solution consisted dominantly with the creation of the software. We came to realize the software will need to be consist of the following: a web application that acts as our movie player, a Python script ran on launch that boots up said web application in full screen, and a Python script that reads the contents of the folder designated to storing the movie files. During these stages, the remote was being researched, and we came to determine that our budget lacked the ability to purchase both an Arduino device and a single board computer. The single board computer was a necessity for the software, so we decided to scrap the Arduino and work on creating a remote with no chip behind it. Luckily, we discovered we could use the GPIO pins of the single board computer to send signals directly from the buttons to the computer.

Nearly halfway through the semester, we began forming a list of parts to purchase. This included the single board computer, two 100mm arcade buttons (big enough for the user), some ribbon wire, and an HDMI cable for the signal. The software around this stage was nearly complete, and we were able to use prototypes of the software for our lab and lecture presentations.

After receiving the purchased items, the software was basically complete (with any modifications needed for the device integration itself), and the building of the remote has commenced. We developed two CAD models for the 3D printing of the casing for both the remote and the single board computer. In the meantime, we also soldered the ribbon wire to the arcade buttons, and completed importing all the created scripts, the web application, and the movie files to the single board computer itself.

About a week and a half before the final showcase, the device was complete. We successfully 3D printed the casing for both parts, and assembled the full device to completion, with the device fully operating as intended with no bugs. The final product was able to pull files from an external USB drive, giving the helpers the ability to add more movies to the solution as they would want, and increase the storage. The device was ultimately wired, as we ran out of time and budget to supply the complexity of both the wireless signal and the remote’s charging/battery power. Same went for streaming services, as we decided to avoid implementing it, as such would require the helpers to dive too deep into troubleshooting, as so many things were discovered to go wrong along that process. Overall, the project was a success, and the design process of our solution proved to be critical to developing a working final solution that best complemented the client.

2.3 Assembly and Creation

Please find all the CAD files here:
https://cad.onshape.com/documents/15996914b5f57752e048cd60/w/de13a332f00e9705805cabc4/e/1fce9818c50afdd6d68f579e?renderMode=0&uiState=642bd8a60f2d6c238fcc130b

2.3.1 Main Components

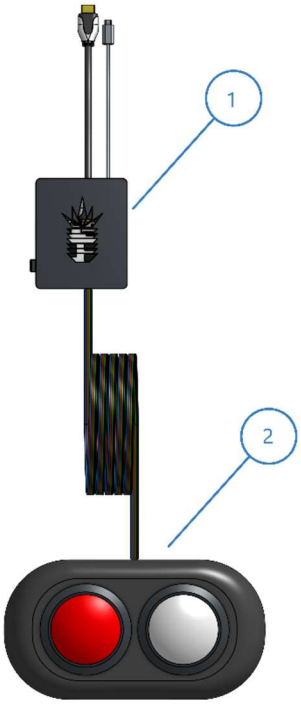


Figure 2 Main Components

The PINE-ABLE ™ is created to mainly revolve around the two main components of system: the central processing unit (Labeled 1 in Figure 2), and the user-handled remote (Labeled 2 in Figure 2).

The central processing unit consists of a Libre Tritium ALL-H3-CC H5 2GB Computer Board, a board very similar to the iconic Raspberry Pi, which is then enclosed in a custom case that both shields the board from the environment while allowing airflow for chip cooling, aided by a heatsink. The board is then attached to the necessary accessories: a USB on which the desired movies are stored, a Micro USB cord for power, and an HDMI cable to connect to a TV or monitor on which the UI and movies will be played on.

The user-handled remote is intended to be placed on the client's tray on their wheelchair. This remote is made with easy to see and click buttons to allow the client to both easily distinguish them and press them with minimal force. The buttons are connected to the Libre's GPIO pins, which read the state of the buttons. It is important to note that pull down resistors were added to each button to prevent a floating state which would lead to false readings of the buttons being pressed.

2.3.2 Central Processing Unit

To take a closer look at the central processing unit, refer to 7.1 EXPLODED COMPUTER UNIT DRAWING. The exploded view drawing clearly outlines the assembly of the casing and the Libre inside. To secure the Libre to the case, four Hex socket head cap screws M2x0.40 x 8 were used, which held onto standoffs in the bottom part of the casing. Past that, the two halves of the casing came together with the help of four more Hex socket head cap screw M2x0.40 x 15 which sat flush with the bottom thanks to some countersinking. Past this, optional Non-Slip Gripping Pads were added to the bottom of the case to cover the screw heads, but primarily to allow the case to not slip on a given surface. That step being optional, the Non-Slip Pads are not shown in the drawing.

2.3.3 User-Handled Remote

To take a closer look at the user-handled remote, refer to 7.2 EXPLODED USER-HANDLED REMOTE. The exploded view drawing shows the assembly of the remote, which is done by positioning the buttons into their designed hole cut-outs, and by placing the top shell onto the case lid/base. The case is held together by having the top shell snapping onto the small protruding offset-inward walls of the bottom lid. As well, some glue was used just in-case and as the remote will not be accessed for any maintenance or firmware updates. Past this, optional

Non-Slip Gripping Pads were added to the bottom of the case to allow the case to not slip on a given surface. That step being optional, the Non-Slip Pads are not shown in the drawing.

2.3.4 Circuit

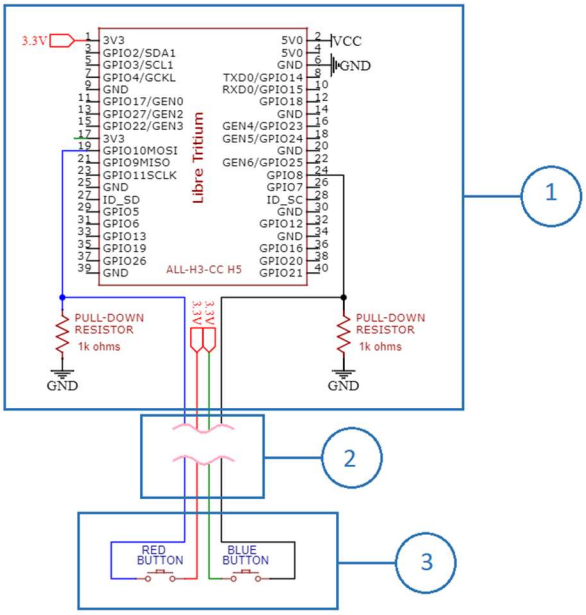


Figure 3 PINE-ABLE ™ Complete Circuit

The complete circuit of the PINE-ABLE ™ is quite simple as shown in Figure 3. In the figure, Label 1 outlines the components within the Central Processing Unit; Label 2 shows the wires connecting the Central Processing Unit and User-Handled Remote; and Label 3 outlines the components within the User-Handled Remote.

Overall, the buttons of the remote are connected to the Libre's 3.3V and then a set GPIO pin, with a pull-down resistor to avoid a floating state and instead be grounded if not pressed. Although it makes no difference which GPIO pins are chosen, the PINE-ABLE ™ connects the buttons to GPIO pins 10 and 8 respectively as they are next to each other. The GPIO pins are coded to constantly read the state of the pins (either HIGH/TRUE/1 if pressed or LOW/FALSE/0 if not) and then simulate an appropriate keystroke which is then picked up by the UI. Within the python code, the GPIO states are read with the following command in the terminal: "gpioget gpiochip1 ###", where ### refers to the corelated GPIO Linux # that can be found in *Libre's Documentation of the Board's Headers*[1]. As such, GPIO pin 10 would have the Linux # 199 while GPIO pin 8 has the Linux # 198.

2.4 Physical Design / CAD

Figure 4 Cable Exits

The PINE-ABLE ™ is designed to be a simple plug-and-run solution, meaning the system simply needs to be plugged into a monitor or similar device, provided power, and it will run on its own. As such, the assembled product is designed to be practically complete, not allowing any of the necessary wires or connections to be broken or unplugged during use, all the technical electronics hidden from the user. This is done through having wire exits that only allow for the neck diameter of the cord and not the actual male plugs, effectively not allowing the wires to be pulled out as seen in Figure 4.

Understanding that there will be times at which a hand impairment or simply shaky hands could lead one to miss a button press, the remote was designed to be as smooth as possible as seen in Figure 5. Avoiding sharp edges, we insured that no matter the force used when attempting to click, a miss will not result in any form of injury.

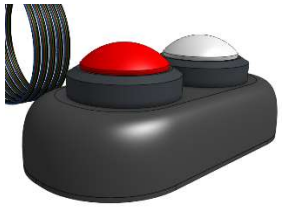


Figure 5 Sharpless Remote

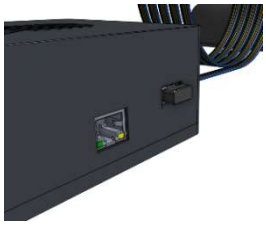


Figure 6 Limited Port Access

To not confuse the clients when first setting up the PINE-ABLE ™, the Central Processing Unit encases the Libre to only allow access to the required USB port for the USB drive with preloaded movies, as well as the optional ethernet port, as shown in Figure 6. The power and HDMI cables are already connected within the case, with the cable necks sticking out, avoiding further confusion. Past that, every other port is blocked off and is only accessible if the case is disassembled for maintenance or troubleshooting.

In order to meet the budget and allow for a lightweight design, all the casings were 3D printed from black PLA. Well allowing for flexible and extravagant designs, the prints needed to be done at a timely manner and avoid defects. As such, ever part was designed with a large flat surface that could be printed on while avoiding the use of supports.

Overall, the complete assembly can be seen at 7.5 ASSEMBLED EXPLODED, which outlines every component at a high-level. The User-Handled Remote came out to be final size of 280 x 50.55 x 150 mm, while the Central Processing Unit was sized at 118.62 x 51.35 x 98.26 mm, as seen in 7.6 REMOTE OVERVIEW and 7.7 CETRAL PROCESSING UNIT OVERVIEW respectively.

All in all, the final Bill of Materials can be found at 7.8 BILL OF MATERIALS.

3 Testing and Validation

3.1 Compliance Matrix: Constraints

#	Constraint	Compliance Assessment	Rationale/Evidence
1	Budget (Keep final costs under current market value stream box ~$120)	Met	Due to resourcing from the wood shop as well as the electrical labs, costs were significantly cut down from $117 to final project expenses amounting to a total of $87
2	Size (Fits on client's tray)	Met	Final dimensions of the controller meant for client's 450x300x20 mm tray were 280x150x50.55 mm. Allowing for convenient placement while not taking too much space away from her.
3	Simplicity of use	Exceeded	Client's care givers were impressed and convinced that the product would be beneficial and easy to use for the client since it only utilizes 2 buttons for all necessary functions
4	Easy visibility	Partially met	Buttons were large and very visible but could have also been differentiated using sensory queues such as bumps or protruding shapes. Also, status of client's visual interpretation of colour was unclear. If known, colour-blind friendlier colours could have been used other than red and white.

5	Minimal Required maintenance	Partially verified	It is difficult to say how much maintenance would be required for practical use. During testing, no maintenance was required to operate the device. Testing by the client and caregivers would be necessary to verify.
6	Minimal strength required to operate	Met	The buttons were designed to be pressed easily as to not let the client motor impairments to be a factor in operations. Requiring a minimum 11.12N of force (2.5lbf). Due to the ease of pressing the button, a time delay of 250ms was added when repressing the button in case of accidental double press.
7	Stability (To not be easily knocked down or dropped from Katlyn's tray)	Met	The product is equipped with rubber grippers to the underside of the remote casing to resist against slipping when placed of a surface. A minimum of 20N or 4.5lb*f are required to slide the remote.
8	Ease of Daily use	Exceeded	The product is designed for "plug-and-play", meaning that as soon as it plugged in, the user-interface is booted up and is ready to function for the client.
9	Allow for 1-5 movies	Exceeded	The product can hold greater than 5 movies, requiring only an external storage device such a USB flash drive. Depending on the storage capacity of the external storage device and the file size of the movie, there is potential for dozens of movies. For testing purposes and ease for client use, we have decided to initially introduce with 8 movies ready to be watched.
10	Bluetooth	Partially Met	The device does have the possibility to allow for a Bluetooth connectivity but due to simplicity of production as well as reliability of the product we decided not to implement that iteration.
11	Internet	Partially Met	The device does have the possibility to allow for internet connectivity through ethernet but because of the clients need for often relocation of viewing we had opted not to take that iteration to trials. However, we still added a feature to auto download the movie posters if connected to ethernet once.

3.2 Objectives Evaluation

#	Objective	Evaluation	Evidence
1	Aesthetically pleasing	Partially Met	The Case was designed and taken out to the public to ask for public opinion regarding the design and quality of the product.

#	Objective	Evaluation	Evidence
2	Buttons should be smooth	Met	The press of the button should be smooth and contain tactile, haptics feedback. To test this out the product was presented to other team members and teams.
3	The case could have more grip	Partially Met	The case was tested to not slide around over various surfaces by team members and fellow teams.
4	The computer housing could have higher heat sync capabilities	Not Met	Three people in the team involved in optimizing the processing speeds of the computer, so the heat transferring capabilities were tested.
5	Could have been mass produced to fit the needs of a certain market.	Partially Verified	The team discussed and characterized the future of the product to potentially further develop the product to adapt the needs for mass production. This included switching casing productions to be injection modeled and for software to be pre-installed on the fresh OS SD card.
6	Could be wireless	Not Met	The product was tested and discussed to potentially be used from other places and be usable from longer distances.
7	Streaming service	Not Verified	The UI was tested and planned to connect a streaming service such as Netflix/Prime.
8	User friendly	Met	The product was given to people to try and figure out the functions of the product on their own. And was confirmed that it was user friendly as everyone after a few clicks understood each function.
9	Reliability	Met	The product was also put through some test of harsh usage to make sure the client does not damage the product itself.

4 Comparison

Practicality Comparison	The current solution employed by the client uses a standard DVD player and television setup, where the staff must change the DVD manually. This setup has a set number of DVDs where the staff may not realize what order the movies have been played, therefore there is a possibility of repeating movies or movies that are played much more than others. Not only that, but it makes sense for the client to be able to select the movies themselves, allowing for choice and control. A new method was needed to be able to reduce staff assistance, while giving accessible choice.

Comparison of Strengths	In both solutions, the client can play movies from their own collection, by either directly playing off the DVD, or using the interface to start playing the ripped DVD movie.
Comparison of Weaknesses	Compared to our solution, the status quo has many disadvantages. The first disadvantage being how DVDs are time consuming to load, and change. With our design, an unlimited number of movies can be loaded within a second, and immediately start playing. Secondly, our design opens the possibility of personal choice to the client, being able to stop a movie and change to a different selection with only a few button presses. Since communication is limited between the client and the staff, this would allow for much more enjoyment of the content, as you can watch exactly what you want to watch, by the second. Finally, our solution also provides the ease of use for the staff. The caretakers will no longer need to load DVDs or ask whether a movie has been played before, you can just set it without the need of external help.
Other Comparisons	The device also allows for downloaded content to be played on the television, which couldn't be played before as it was limited to DVDs. Downloaded YouTube videos, home videos, or extra features can be loaded the same as any other movie, which gives more access to the client.

Current solution vs Our Solution

5 Potential Improvements

Looking back at the feedback and reviews from both the judges and clients, it is fair to assume that our group excelled at completing this project with high efficiency and quality. Our effective yet easy to use system was well liked by judges, but we did receive some feedback. To differentiate between the 2 buttons of our design, we colour coded them, the one that controlled the movie selection was red and the pause/play button was coated with a white colour. What we did not consider was if we were to expand into production, we needed to account for colour blind clients, and we could have included brail on the buttons to help colour blind clients. Another

weak point of our design was the ribbon cable connecting the user interface and the microcontroller. We did include a string release system to solidify the endpoints of the cable however, we could have used a braided cable to cover the ribbon wire to increase the longevity of the cable. Another alternative route we could have taken was the wireless option. With using a wireless connection between the remote and the microcontroller, the client could have gotten more versatility with the remote and not be worried about tangling the connecting wire. Another feature that we could have improved was the movie title display switching. The movie that was being chosen should have been outlined by a distinct colour instead of just being illuminated, to make it more clear which option was being chosen.

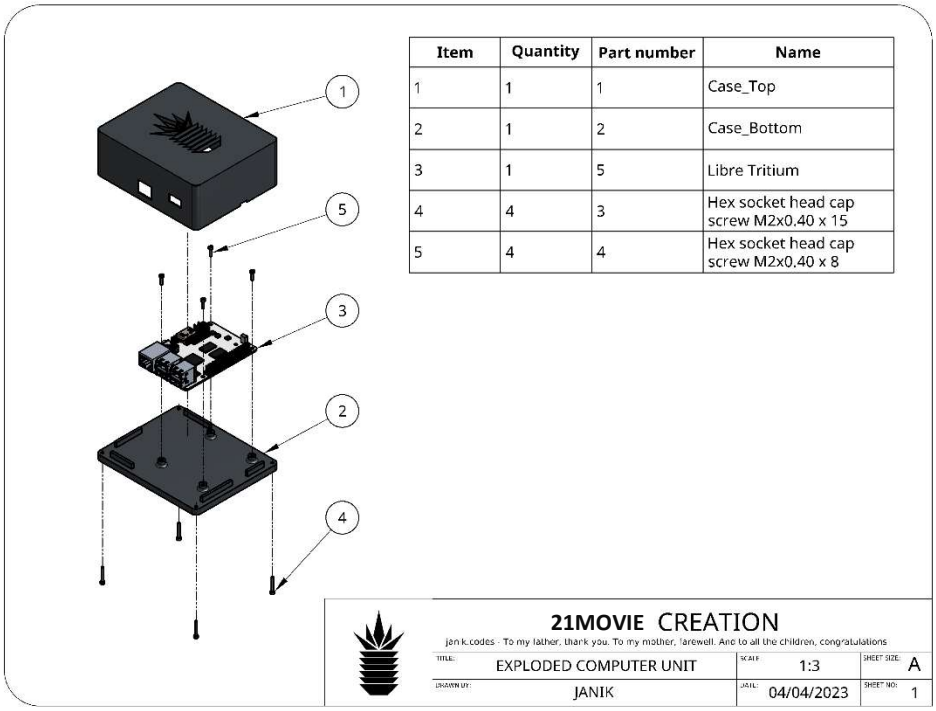
6 References

[1] Libre's Documentation of the Board's Headers

"ALL-H3-CC-V1.0A Headers" libre.computer, 2023 [Online] Available: https://docs.google.com/spreadsheets/d/1ruDmSzQexpXX2iCCqma7Vwu6t_ZlEt8robvI0qSdbwk/edit#gid=0. [Accessed 04 04 2023]

7 Appendix A

7.1 EXPLODED COMPUTER UNIT DRAWING



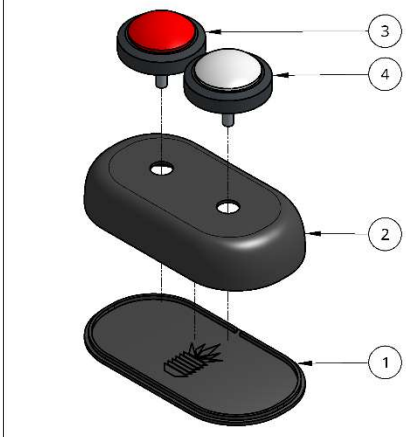
Item	Quantity	Part number	Name
1	1	1	Case_Top
2	1	2	Case_Bottom
3	1	5	Libre Tritium
4	4	3	Hex socket head cap screw M2x0.40 x 15
5	4	4	Hex socket head cap screw M2x0.40 x 8

21MOVIE CREATION

jan k.codes - To my father, thank you. To my mother, farewell. And to all the children, congratulations

TITLE: EXPLODED COMPUTER UNIT	SCALE: 1:3	SHEET SIZE: A
DRAWN BY: JANIK	DATE: 04/04/2023	SHEET NO: 1

7.2 EXPLODED USER-HANDLED REMOTE



Item	Quantity	Part number	Name
1	1	6	Body_Base_Lid
2	1	7	Body_Shell
3	1	8	Button
4	1	8	Button (Painted)

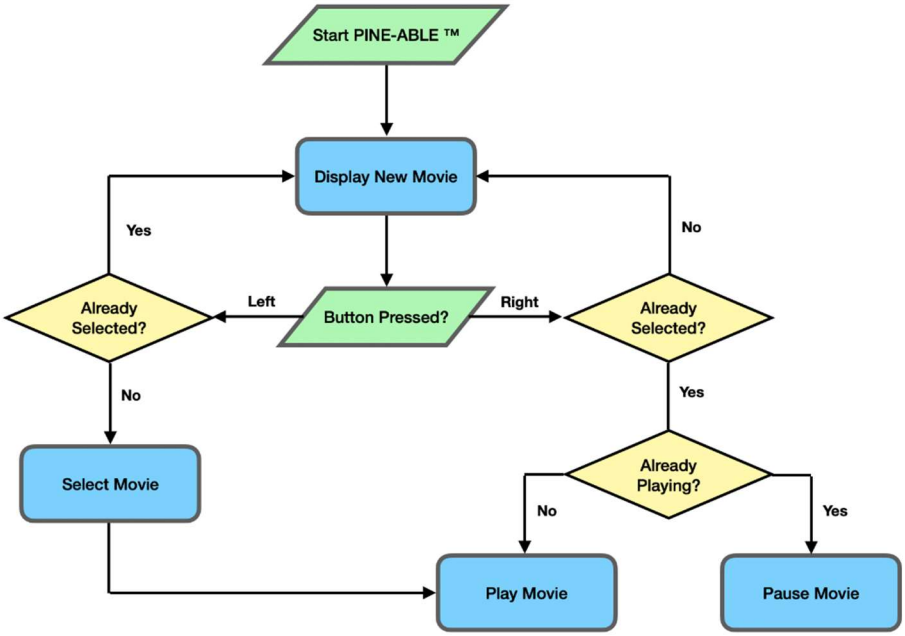
21MOVIE CREATION

jan k.codes - To my father, thank you. To my mother, farewell. And to all the children, congratulations

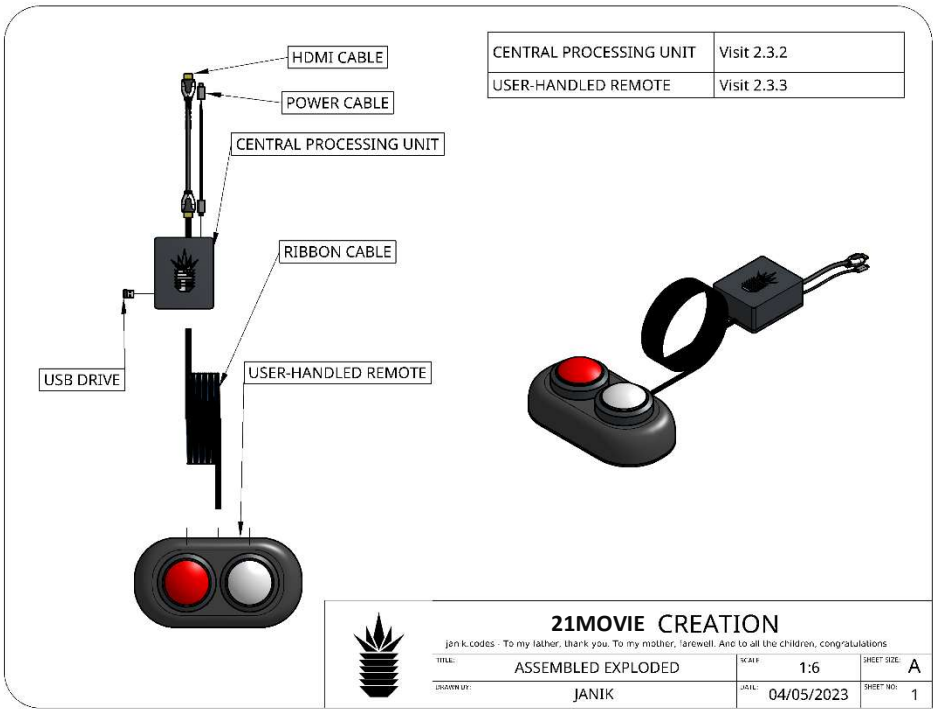
TITLE: EXPLODED USER-HANDLED REMOTE	SCALE: 1:4	SHEET SIZE: A
DRAWN BY: JANIK	DATE: 04/04/2023	SHEET NO: 1

7.3 USER INTERFACE

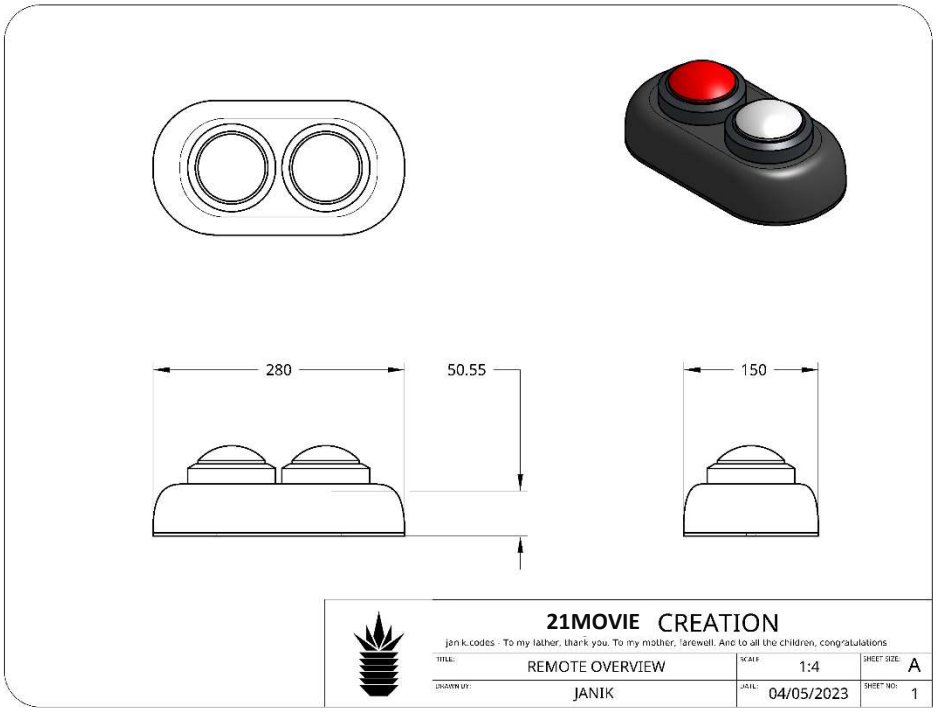
7.4 USER INTERFACE FLOWCHART



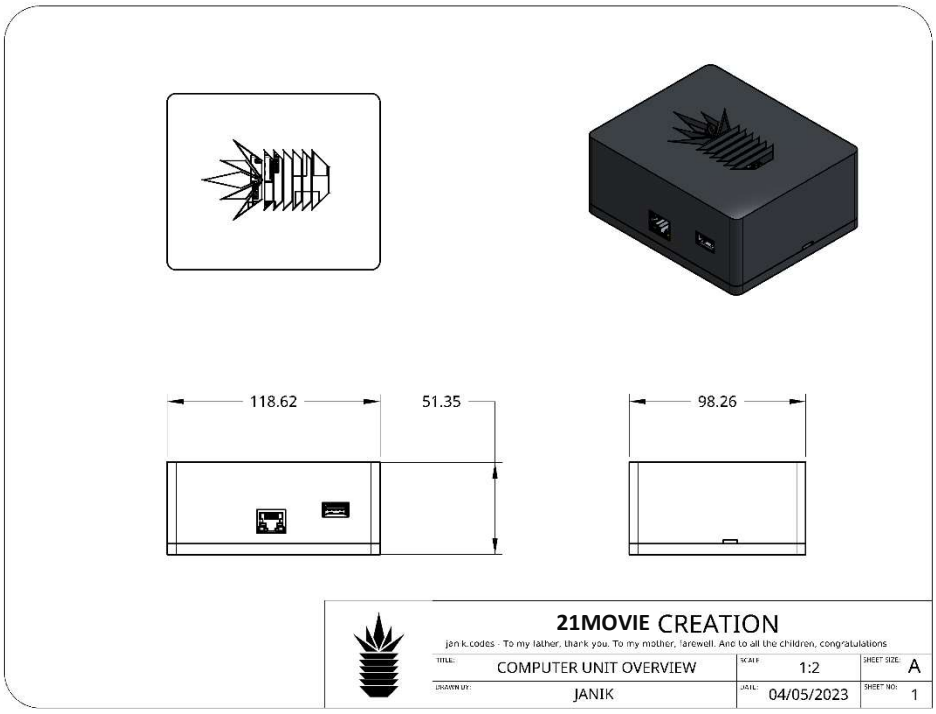
7.5 ASSEMBLED EXPLODED



7.6 REMOTE OVERVIEW



7.7 CENTRAL PROCESSING UNIT OVERVIEW



7.8 BILL OF MATERIALS

Component	Quantity	Part Number	Name	Source	Cost Per Unit ($CAD)
CENTRAL PROCESSING UNIT	1	1	Case_Top	3D Printed	$ 1.30
CENTRAL PROCESSING UNIT	1	2	Case_Bottom	3D Printed	$ 0.53
CENTRAL PROCESSING UNIT	1	3	Libre Tritium	https://a.co/d/4kpDXlW	$ 57.85
CENTRAL PROCESSING UNIT	4	4	Hex socket head cap screw M2x0.40 x 15	https://a.co/d/evWCABT	$ 0.23
CENTRAL PROCESSING UNIT	4	5	Hex socket head cap screw M2x0.40 x 8	https://a.co/d/evWCABT	$ 0.23
CENTRAL PROCESSING UNIT	1	11	USB Drive	https://a.co/d/9FSTHZJ	$ 8.85
CENTRAL PROCESSING UNIT	1	12	HDMI Cable	https://a.co/d/8X1Yx9b	$ 8.99
CENTRAL PROCESSING UNIT	1	13	Micro USB Power Cable	https://a.co/d/fs3XMaU	$ 8.09
CENTRAL PROCESSING UNIT	1	14	SD Card	https://a.co/d/228pbft	$ 6.88
MISC	1	10	Ribon Cable (10')	https://a.co/d/g2aUb0V	$ 0.73
USER-HANDLED REMOTE	1	6	Body_Base_Lid	3D Printed	$ 0.21
USER-HANDLED REMOTE	1	7	Body_Shell	3D Printed	$ 0.43
USER-HANDLED REMOTE	1	8	Button	https://a.co/d/iqDrSps	$ 11.59
USER-HANDLED REMOTE	1	9	Button (Painted)	https://a.co/d/iqDrSps	$ 11.59
				Total Cost:	$ 117.50