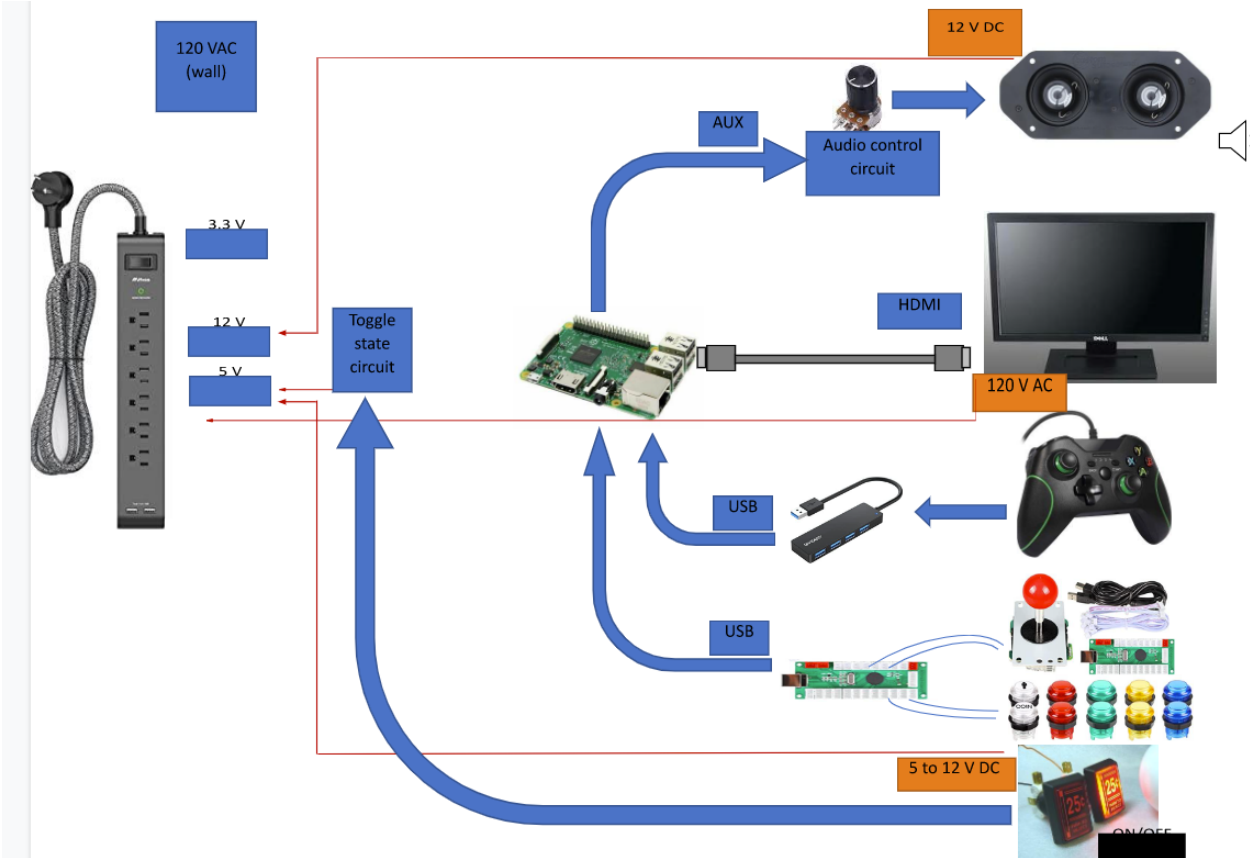
4.3 Proposed Design

4.3.1 Overview

Provide a high-level description of your current design. This description should be understandable to non-engineers (i.e., the general public). Describe key components or sub-systems and how they contribute to the overall design. You may wish to include a basic block diagram, infographic, or other visual to help communicate the overall design.

Our project is creating an arcade style cabinet. This cabinet needs to have audio, visual outputs and have a couple inputs for controlling these. It also needs to have the controls as inputs, with several different controllers as options.

For our design, we decided to have a Raspberry Pi 3 control many subsystems to display the game. The first sub system is a display, which will be shown using a HDMI monitor. Next is audio, which will be connected through a set of speakers. The user will have a potentiometer to control the volume of the game for this speaker. Next is controls, which will be mapped and connected through the USB ports for the Pi. Lastly, an on/off switch is needed, so we have a plan to use an authentic coin drop button to turn the machine on and off.



4.3.2 Detailed Design and Visual(s)

Provide a detailed, technical description of your design, aided by visualizations. This description should be understandable to peer engineers. In other words, it should be clearly written and sufficiently detailed such that another senior design team can look through it and implement it.

The description should include a high-level overview written for peer engineers. This should list all sub-systems or components, their role in the whole system, and how they will be integrated or interconnected. A visual should accompany this description. Typically, a detailed block diagram will suffice, but other visual forms can be acceptable.

The description should also include more specific descriptions of sub-systems and components (e.g., their internal operations). Once again, a good rule of thumb is: could another engineer with similar expertise build the component/sub-system based on your description? Use visualizations to support your descriptions. Different visual types may be relevant to different types of projects, components, or subsystems. You may include, but are not limited to: block diagrams, circuit diagrams, sketches/pictures of physical components and their operation, wireframes, etc.

For Electrical-

Three circuits are needed. A power circuit, on/off circuit, and an audio control circuit. In the block diagram above, these circuits are shown with a blue box.

Power circuit-

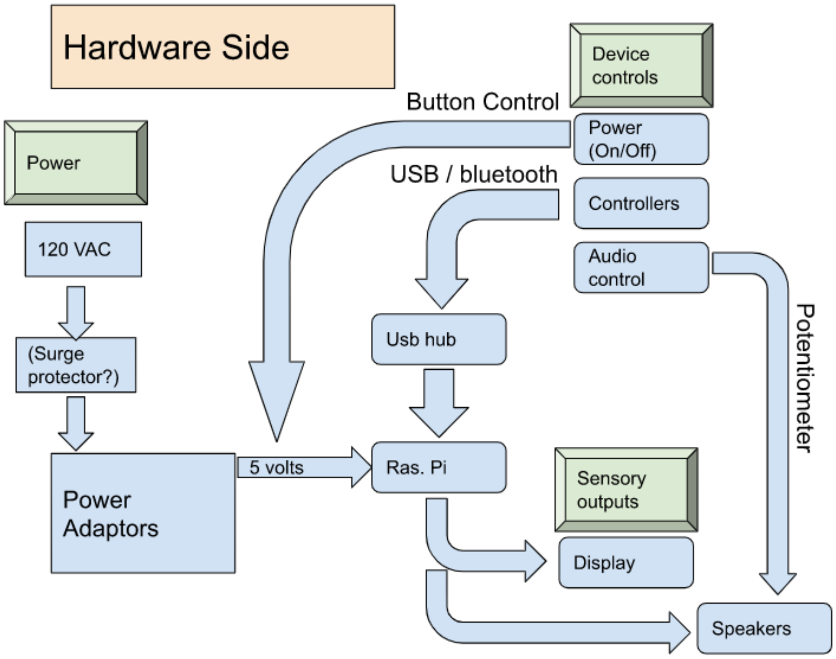
- For many of the devices, we need a select DC power. 12V, 5V, and 3.3V are the most common needs. So a circuit that can supply this is needed.

On/Off circuit

- A simple switch would work for this, but our client wanted this to look much like an old arcade. So the coin drop button would increase the authenticity of the device. When pressing the button, we need a circuit to hold a high state to the rest of the subsystems. When it is pressed for a second time, we need the circuit to stop all power to the other devices.

Audio control

- With the speakers being encased in the device, controlling audio would be easiest with a potentiometer. This means a circuit will need to interrupt the connection from the Pi to the speakers. Clean audio is expected, but also with the games being relatively old the quality of the audio will already be a much lower expected quality.



Software side

-A general user interface will be needed for this design that has at least 6 different screens. A basic start menu that will give the user a few options like game list and settings. Then we will have two storage directories that will store games and user settings.

-Our app will launch on start up, with an easy feature that is provided by the PI.

-The first screen will be a basic start menu for the user to see that will have limited options like play menu or settings

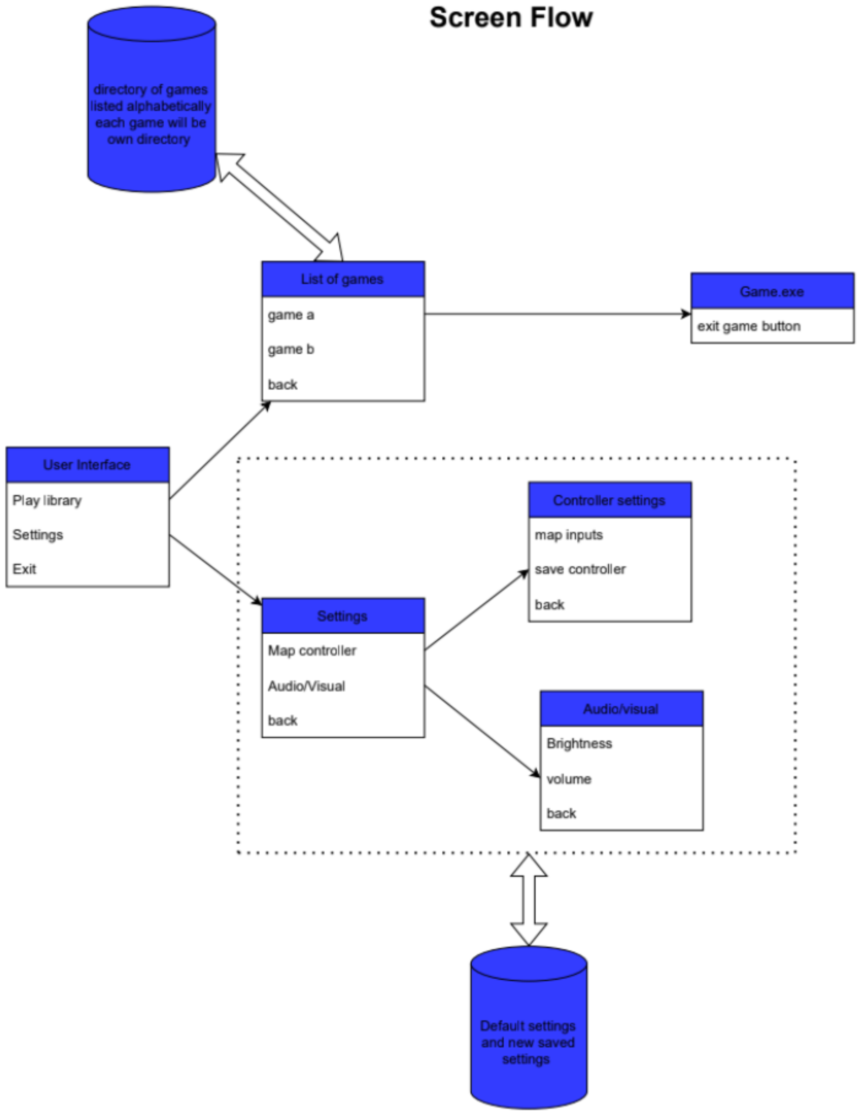
- if the user chooses to play a game the next screen will display a list of games in alphabetical order from the local game directory.

- a user will be able to choose a game from the list and once selected the file will launch the game and if a user wants to exit they will be sent to the start menu

-the settings screen will allow users to map controls or update the audio/visual settings these settings will be saved in a different settings directory.

The application will be written in java and plan to have CI/CD implemented on git so if we want to add games, we can do it online and it will be auto deployed to the pi.

As of right now we are debating between downloading emulators that are able to run original games and custom exe games. If possible we would like to use both. The limitations of running both could be the amount of storage space and usability of exes.



4.3.3 Functionality

Describe how your design is intended to operate in its user and/or real-world context. What would a user do? How would the device/system/etc. respond? This description can be supplemented by a visual, such as a timeline, storyboard, or sketch.

The user first needs to plug in the device. This will be made easy by having all the subsystems be powered by just one extension cord. This will be long enough to reach down under a table that the device would sit on. Once plugged in, the user would need to press the coin drop button to start the device up.

Once started up, The Pi will have a startup phase and then display a simple UI. This will allow the user to select a game. Once selected, the Pi will open the needed emulator and begin the game.

For controls, the user has a couple. First is the power button. Second is an audio control potentiometer. Next is a selection of controls. Built in is a set of arcade buttons and joysticks. These work as a base set of controls. Using a UBS extension, a user can select a different controller.

4.3.4 Areas of Concern and Development

How well does/will the current design satisfy requirements and meet user needs?

It keeps the aesthetics of an arcade cabinet. It keeps the look, the controls, and the display.

Based on your current design, what are your primary concerns for delivering a product/system that addresses requirements and meets user and client needs?

With different games, we need to use a different emulator. This may get difficult to swap between emulators while still using the basic arcade controls. We may have to just turn the device off and on to select a new game.

What are your immediate plans for developing the solution to address those concerns? What questions do you have for clients, TAs, and faculty advisers?

For now, we may have to just turn the device off and on to select a new game. We are also planning to add a reset button for this purpose.

4.4 Technology Considerations

Describe the distinct technologies you are using in your design. Highlight the strengths, weaknesses, and trade-offs made in technology available. Discuss possible solutions and design alternatives.

The Raspberry Pi has many pros and cons which will affect our project.

Pros- cheap, small, and very adaptable

Cons- not a powerful, runs only linex, may have problems running .exe files

4.5 Design Analysis

Discuss what you have done so far, i.e., what have you built, implemented, or tested? Did your proposed design from 4.3 work? Why or why not? Based on what has worked or not worked (e.g., what you have or haven't been able to build, what functioned as expected

or not), what plans do you have for future design and implementation work? For example, are there implications for the overall feasibility of your design or have you just experienced build issues.

We have collected each part except for the monitor. We have tested each work and functions as expected. We have not started any code or the circuits, but will be looking to start that on our 10/26/2022 meeting.