Handheld Emulation Station

Design Document

Group: sdmay19-25

Client/Faculty Advisor: Dr. Julie Rursch

Team Members:

Nick Lang

Sean Hinchee

Matthew Kirpes

Nic Losby

Jacob Nachman

Team Email: sdmay19-25@iastate.edu

Team Website: https://sdmay19-25.sd.ece.iastate.edu

Version 1.0

Updated: 10/12/2018

Table of Contents

1. Introduction

1.1 Acknowledgement

No acknowledgements at this time.

1.2 Problem and Project Statement

The problem to be addressed is that people do not have a dedicated platform from where they can play old school games in their pockets. We ourselves love to play old school games, but do not like wasting our phones battery life to do it. So we want to design a handheld emulation station that can fit in our pocket, with ideal battery life.

Our plan is to get an LCD, battery, some buttons, and a raspberry pi, and hook them all up to a PCB board with a stylish design case. We need to make sure the battery life can last all day and can play any retro game the user desires. We want our emulation station to be a good enjoyable experience to use.

1.1 Operational Environment

Overall, we need to make sure that our emulation station will be able handle the cold and the heat. Users will have it in their pocket so we want to make sure it can handle all basic weather conditions. Water resistance is ideal so the user does not have to worry about their device getting ruined if they are out with it in the rain as well.

1.4 Intended Users and Uses

The intended users are people who want a handheld and portable platform in order to play classic games.

<u>Uses</u>

- Handheld and portable gaming
- Wireless when not charging
- Docking port for charging and to share display to another screen via HDMI

1.5 Assumptions and Limitations

Assumptions

- The games being emulated must be able to run on a raspberry pi zero
- One user at a time
- The games emulated must be owed due to copyright laws

Limitations

- The end product will not weigh more than 12 ounces
- The cost of the end product will not be more than one hundred fifty dollars
- An end product must be created before May 2019
- Battery Life(~2 hours)

1.6 Expected End Product and Deliverables

The end product will be a handheld and and portable gaming platform for classic games that can be run on a Raspberry Pi Zero. This will have a screen, battery, and tactile buttons. There will also be a docking port for charging and the ability to share display to another screen via HDMI. There will also be a user's manual along with the emulation system to describe its uses. This will all be delivered before May 2019.

2. Specifications and Analysis

2.1 Proposed Design

The project has three major components; PCB design, creating a kernel module, and finally case design. The first component is being worked on by researching PCB design software and getting to know how to use the software to layout a PCB. This PCB will be the interface between the battery and the raspberry pi. This PCB will be the central part of our design and will require us to use oscilloscopes to make sure that our design will have the most efficient power transfer.

The kernel module is also a key part of the project and this will involve creating interrupts to make the emulator more responsive. This part of the project is important to getting the best performance out of the emulation and making the controls as responsive as possible. This part of the project is being researched to figure out the best way to implement this kernel module to the required specifications that will most improve the input lag.

The case design is also important to protect the internals of the emulator without hindering the use of the buttons. For this piece of the project we have been researching different 3d design softwares to develop a 3d model that we can 3d print. The most important part of this part of the project will be creating a sleek design that houses all of our components and may take multiple attempts to get just right.

2.2 Design Analysis

To date we have tried laying out cardboard pieces trying to get a feel for where the parts should layout. It went well in order to get a feel for the size of the completed device. We may need to modify the shape of the PCB. Currently it is planned to have a large cutout which could help increase the size of the battery but it would be marginal and increase engineering thought into the traces in the PCB significantly.

One issue that still needs to be solved is battery density. It may be worth it to reuse battery cells meant for phones and chain them together in the same style at the first gen iPad. This would heavily complicate our power circuit and it may not be worth it in terms of payoff.

The strengths of all of our analysis so far is pulling knowledge from iFixit teardowns to see how others in industry are currently designing their power circuits.

3. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

The necessary types of tests are:

- Hardware testing for the power circuit to ensure reliability and prevention of short circuits
- Hardware testing of the joint between the RPi and PCB to ensure it is electrically connected
- Hardware testing for the screen to ensure the displayed image is accurate
- Software testing with the buttons to ensure the input is registered quickly and accurately
- Hardware testing of the entire system to ensure power draw is at its minimum
- Software testing for the kernel module to ensure the kernel can be upgraded in the future without breaking the module
- Software testing for the emulator to ensure accurate and efficient emulation.

The procedure for testing the power circuit will be using a bench power supply to slowly lower input voltage (from the battery side) to check if the boost converter can handle the out of spec voltage in order to ensure system stability if the battery degrades over time. A challenge will be the simulating the power circuit since it has been a few years since anyone on the team has touched circuit simulators.

The procedure for testing the joint between the RPi and PCB as well as the screen will consist of an oscilloscope to check the signals being sent and if they match the expected waveform. The screen will also go through a visual check for the displayed colors and the LCD profile will be adjusted in order to ensure color accuracy.

The procedure for checking power efficiency will be using a multimeter in the amperage setting to check each components power draw and any software configuration will be tweaked to lower power draw.

The procedure for checking the kernel modules' extendability will be installing an updated kernel and adding the proper initd or initramfs hooks to ensure our kernel module is rebuilt and reloaded with every new kernel.

The procedure for checking emulation accuracy will be based of existing emulator tests and well as checking the leaked reference manuals for the proper system with reverse engineering tools such as IDA and Binary Ninja to ensure the correct opcodes are being read and executed from the ROM.

3.1 Interface Specifications

We will be using the GPIO pins on the Raspberry Pi for button input and potential for driving the display as well. The PCB will give access to all of the 40 pins and the display pins are so small it would be very difficult to solder a ribbon cable across all of them to run to the display. If the GPIO pins are unable to be used for display output we could use an external FPGA to run the display it would increase our power consumption but reduce engineering difficulty.

The kernel module will interface with the USB bus in order to enable interrupt-like performance for input lag reduction.

3.2 Hardware and software

We will be using a multimeter, bench power supply, oscilloscope, and a logic analyzer. The multimeter will be great for testing the battery voltage and current draw. The power supply will be used to test if circuits work first before introducing the varying voltage from the battery discharging. The oscilloscope and logic analyzer will help check the input from the buttons for accuracy as well as checking and debugging output to the display.

We will also use Eagle and numerous circuit simulation software in order to test our circuits before building them physically as well as to check the correctness of circuits built by the PCB supplier's robots.

3.3 Functional Testing

Unit tests can be easily composed for the kernel module as well as the potential emulator. The upsides to these methods are that we can incrementally develop tests as well as the software itself. In the case of the kernel module, these would take the form of tests to quantitatively test whether we are getting rapid enough input for USB. In the case of the potential emulator, the tests would be in regards to accuracy of emulation. Ideally, speed would come naturally to a properly implemented and accurate emulator. Potentially we may be able to test for speed in the emulator, but it is possible that this will not be practical in a unit test.

System testing will most likely have to be done manually with quantitative results being calculated where possible. Ideally, the system's unit tests should take care of accuracy, so system tests should be focused on attempting to measure performance and minimize latency within the system itself.

3.4 Non-Functional Testing

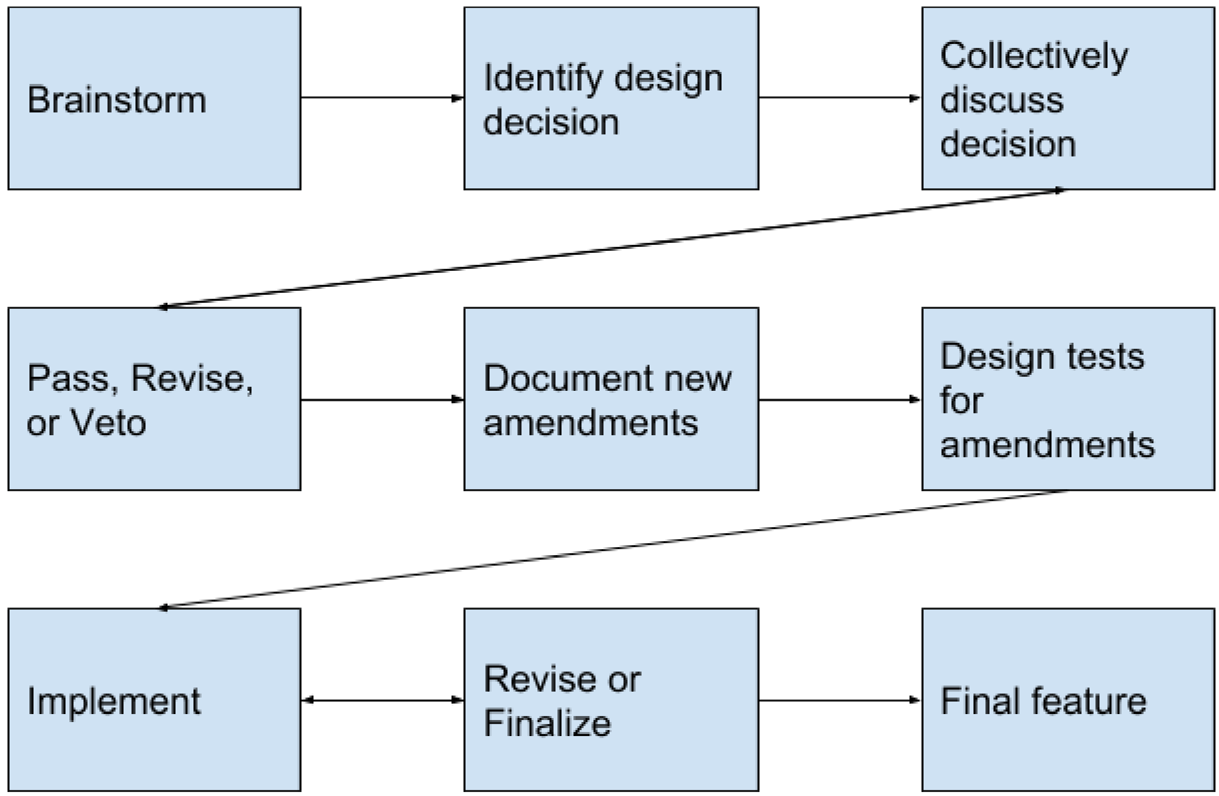
Testing for performance will in major take place in the system testing phase as it is an important part of our system's functionality. That is, real time performance is a goal of the project and is a functional rather than non-functional goal.

Security will be tested by attempting to breach or penetrate the protections of the emulation station and derive from that point what potential strengths or weaknesses are within the architecture of the project. Compatibility with existing systems is tested by the process of loading existing roms and comparing the performance and accuracy experience with existing systems to our system. Usability, similarly, takes root in comparisons to existing systems and the usability present there and comparing our system thereto.

3.5 Process

Methods described in section 3.2 are explained in line with their definitions.

Flow diagram for testing and design development:



3.6 Results

Since we are still in the research and planning phase of the project there is nothing to discuss as far as the testing results and successes or failures related to it. This section will contain the the images of oscilloscope testing of the PCB as well as the unit tests for the kernel module. The testing phase will be very important to this project and therefore this section will contain lots of information about the struggles and successes of the testing phase but since we are still in the planning and researching phase the information is not yet available.

4. Closing Material

4.1 Conclusion

So far our work has been focused on researching and expanding our knowledge of how to go about designing the emulator. We have also looked into different tools and started learning how to use these tools to help us achieve our project goals of creating an emulator that meets the needs of nostalgic gamers who need a compact and portable emulator with great battery life. We will go about achieving these goals by creating a PCB to interface a battery with our raspberry pi that will maximize the battery life. We will also create a kernel module to get the greatest performance from the emulation and reduce lag. These will help to achieve the the goals of our project.

4.2 References

None at current time.

4.3 Appendices

None at current time.