Battery-Free Handheld Game

DESIGN DOCUMENT

Team Number: 18
Client: Henry Duwe
Adviser: Henry Duwe
John Brose, Chief Engineer of Power Systems
Jake Larimore, Chief Engineer of Hardware Integration
Franklin Bates, Chief Engineer of Microcontrollers
Shivam Vashi, Chief Engineer of Software
Daniel Lamar, Test Engineer
Team Email: jbrose@iastate.edu
Team Website: https://sddec21-18.sd.ece.iastate.edu/

Revised: 4-25-2021 / v3

Executive Summary

The goal of our project is to develop a handheld battery-free gaming device that provides users with fun, enjoyable experience while utilizing energy harvesting to supply the energy required for operation. The device will provide users with a unique experience of interacting with the game in a manner that supplies energy in both a single and multi device environment.

Development Standards & Practices Used

- Agile software development
- IEEE 12207-1996 - ISO/IEC International Standard - Information Technology - Software Life Cycle Processes
- IEEE C63.5-2019 - American National Standard for Electromagnetic Compatibility--Radiated Emission Measurements in Electromagnetic Interference (EMI) Control - Calibration and Qualification of Antennas (9 kHz to 40 GHz) - Corrigendum 1

Summary of Requirements

- The device shall run with power being supplied solely from energy harvesting.
- The energy harvesting shall be a part of the gameplay.
- The device shall support some form of multiplayer functionality.
- The device shall be as portable as a standard laptop.
- The game shall be reasonably intuitive to pick up and play by anyone who has played a game before.
- The game shall run through at least one room challenge state after energy harvesting.
- The device shall show a low power screen when it is running out of battery, telling the user to energy harvest.

Applicable Courses from Iowa State University Curriculum

EE 201, EE 230, EE 330, CPRE 288, CPRE 488, CPRE 388, COMS 309, COMS 319

New Skills/Knowledge acquired that was not taught in courses

- Schematic design
- PCB design in KiCad
- C++ best practices
- Interdisciplinary Collaboration

Table of Contents

List of figures

1 Introduction

1.1 Acknowledgement

Professor Henry Duwe was our technical advisor for this project, providing us with technical knowledge and access to labs.

Vishak Narayanan, a graduate student at Iowa State University, provided us with guidance and advice on energy harvesting systems.

1.2 Problem and Project Statement

Design a gaming device that does not have a battery component. The device will use power harvested from the user. Other features include incorporating energy consumption into gameplay and utilizing multiplayer functionality in certain elements of the game.

Batteries are an easy solution to powering devices that cannot have a steady supply of generated power at hand. Although, as new technology is emerging, the downsides to batteries is becoming more and more significant. Batteries can only hold a limited amount of energy, and will either need to be recharged or disposed of. They also take up significant space, have a limited lifespan, and can be expensive depending on the application. Battery-free devices are becoming increasingly common in the industry, and our gaming device is taking another step in this direction. This is a proof of concept that will help further the knowledge in this field and help show if a battery-free device is a viable alternative to a battery powered device.

1.3 Operational Environment

The handheld game will be used in an indoor setting with a regulated temperature and calm weather conditions. It will be able to withstand reasonable wear and tear while being played by the user.

1.4 Requirements

The device shall run with power being supplied solely from energy harvesting.
The energy harvesting shall be a part of the gameplay.
The device shall support some form of multiplayer functionality.
The device shall be as portable as a standard laptop.
The game shall be reasonably intuitive to pick up and play by anyone who has played a game before.
The game shall run through at least one room challenge state after energy harvesting.
The device shall show a low power screen when it is running out of battery, telling the user to energy harvest.

1.5 Intended Users and Uses

The intended use of the project is to have a device that can provide the user a brief, entertaining multiplayer experience. The intended audience for this device is primarily college faculty, staff, and students.

1.6 Assumptions and Limitations

Assumptions:
The users will be reasonably careful with the device.
The users will understand the basic operations of a dungeon crawler game.
The users will be physically able to operate the energy harvesting devices.

Limitations:
The game must have interaction between at least two devices.
The device must have energy harvesting as the main source of power.
The game must feature energy harvesting as a part of the gameplay.
The device must be able to be hand held.

1.7 Expected End Product and Deliverables

The expected end product will be a device that includes an E-ink screen, energy harvesting push buttons, dip switches, and a hand crank all into a small handheld unit. The energy harvesting push buttons along with the hand crank will provide the entirety of the power needed to run the game, and will be incorporated into the gameplay. The push buttons, dip switches, and hand crank will form the core tactile interaction with the user while the E-ink screen will provide visual cues and video game information.

This device will allow you to play a dungeon crawler game that features energy harvesting elements and multiplayer. The main gameplay loop will be switching between players deciding between rooms to explore, to challenges in those rooms the players need to complete. If the device reaches a low power state, we will store the game's state, including things like the player's stats and any rooms that have been completed, into the device's memory, and we will then restore the game state from the memory once the device has been sufficiently powered by energy harvesting.

2 Project Plan

2.1 Task Decomposition

1. Gameplay Design:
    a. Define the category of game that will be designed.
    b. List out the required aspects of the game play.
    c. Set “stretch goals” to implement once required goals are completed.
2. Energy Harvesting:
    a. Research the resources available to generate power for the gaming device.
    b. Decide how the user will operate the energy harvesting components to interact with the gameplay.
    c. Measure Part Specifications
    d. Harvesting Detection Design
3. Hardware Design:
    a. Research and determine what microcontroller, display unit, and other hardware elements will be used.
    b. Design Power System
    c. Make PCB
    d. Assemble Final Device Prototype
4. Software Design:
    a. Graphical User Interface for use with E-ink display
    b. Environment generation algorithm
    c. Character creation and motion feature
    d. Room challenges feature
    e. Multiplayer feature
    f. Save/Load state on while on low power or returning from a low power state
5. Microcontroller Interface Design:
    a. E-ink display
    b. Button & RPM detection
    c. Energy harvesting detection
    d. Multiple device communication
6. Test Design:
    a. Energy harvesting & power generation
    b. Display & User Interface
    c. User input detection
    d. Environment Algorithm
    e. Room Challenge Algorithm
    f. Multiplayer gameplay
    g. Boss fight logic
    h. Saving/loading states saved to fRAM
    i. Room Challenge power production produces enough energy for room challenge loop
    j. Enough energy stored when boss fight is reached

2.2 Risks And Risk Management/Mitigation

Gameplay Design Risks: 0.1
Energy Harvesting Risks: 0.4
Hardware Design Risks: 0.7

> *Mitigation*: Review the design documents and datasheets for each hardware component to ensure all connections are correct. If so, reevaluate the uses for each component and ensure each component is receiving the right amount of power. If so, consider replacing certain hardware components that seem to be causing the problem.

Software Design Risks: 0.8

> *Mitigation*: Debug all errors in the program. Ensure the error is occurring in the software, not the hardware. Review datasheets for the microcontroller to debug any miscommunication between the hardware and software.

Test Design Risks: 0.6

> *Mitigation*: If the test fails was it due to hardware/software or the design of the test. If test failure was due to the design of the test reevaluate test design. Otherwise, use hardware/software mitigations.

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Gameplay Design: The game will be a dungeon crawler that is intuitive to play, and does not lose data or prevent the user from continuing during it's full gameplay cycle.

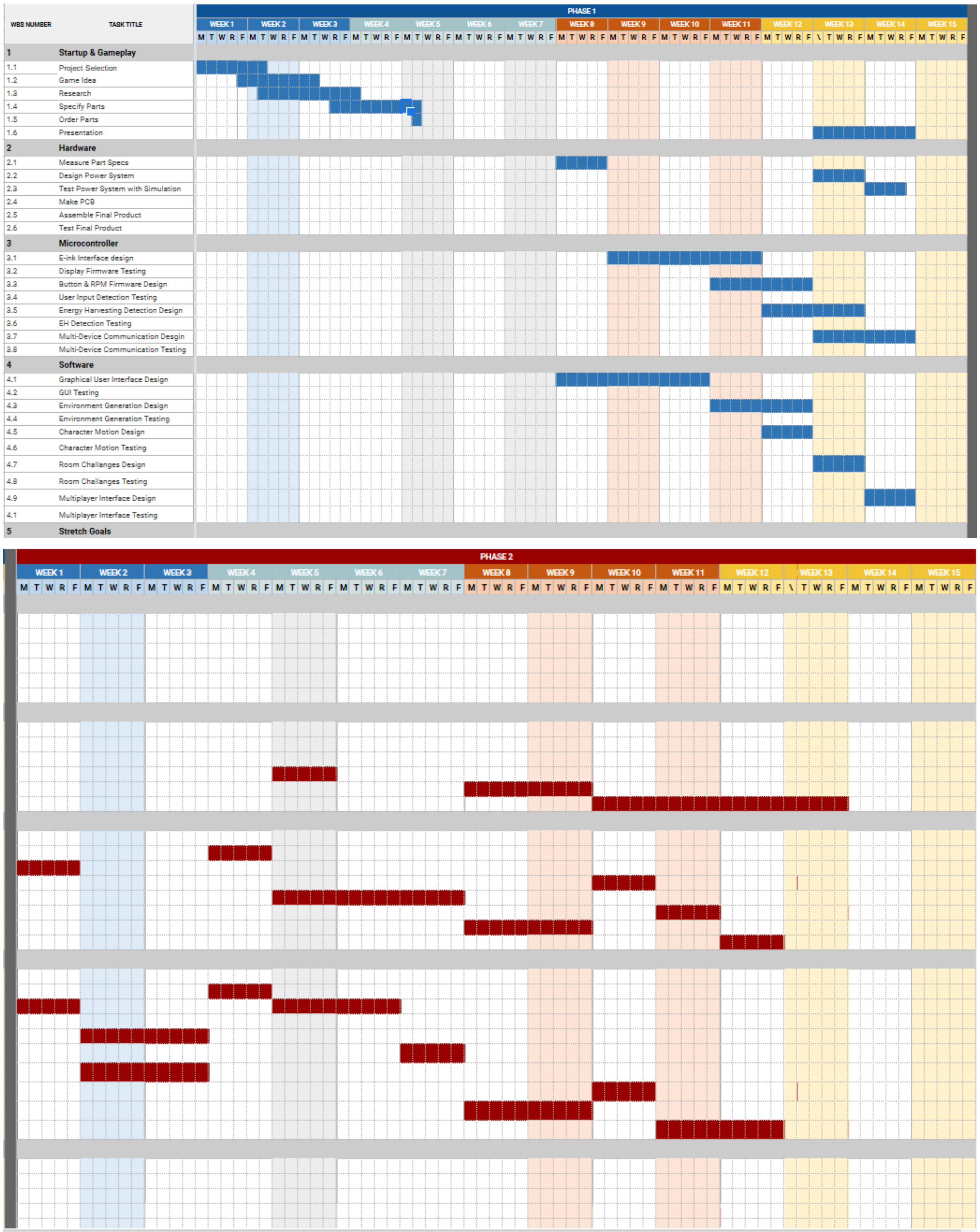
Energy Harvesting: The users will be able to generate 100% of the necessary power for the proceeding turn by completing the current task. The device will turn off once it loses the necessary power to remain on. The user will be able to generate enough energy to turn the device back on in under 5 seconds.

Hardware Design: The hardware will connect the power producing energy harvesting components to the supercapacitor, microprocessor, and E-ink display. This will allow the powering of the device as well as the ability to tell how fast the user is producing power. The hardware will have a constant current consumption in the nanoamp range while not updating the display to minimize power consumption.

Software Design: The software will be able to run the full game from start to finish, generating rooms to give player's stats through challenges, and display all of this to the user through an intuitive user interface. The software should also be able to successfully save and load the game state during low power without any data loss.

Microcontroller Interface Design: A fully charged supercapacitor on the microcontroller will be able to update the e-ink display, and generate a new room once. When the microcontroller is in the "saved state" mode and it receives enough power to resume gameplay, it will reboot to the last saved UI and gameplay state in under one second. Multiplayer communications will occur with less than 500 ms delay.

2.4 Project Timeline/Schedule



2.5 Project Tracking Procedures

The group has used Trello to communicate what tasks need to be completed, what tasks are being completed, and what tasks have been completed. The team also used a Discord server as a means of communication and posting related information. Lastly, Google Drive and Github were used to store files and programs used in the design and documentation of the project. We will continue to use these platforms as the project continues.

2.6 Personnel Effort Requirements

Tasks	# Hours Req	Details
Gameplay Design	5-10 hours	What kind of game we want to create, how it will implement energy harvesting, how it will function.
Energy Harvesting	25-35 hours	Find out how we can collect energy with devices, and how we can implement these devices in our overall design.
Hardware Design	30-40 hours	Creation of the schematic connecting microcontroller, circuits, and all devices together. Layout of the PCB based off of the schematic.
Software Design	30-40 hours	Developing GUI and gameplay.
Test Design Software	40-50 hours	Testing gameplay and functionality, debugging, ensuring no errors are found.
Test Design Hardware	40-50 hours	Testing voltages and currents at certain nodes in schematics, measuring power dissipation across components, ensure the capacitor is charging and discharging correctly.

2.7 Other Resource Requirements

Lab equipment (oscilloscope, multimeter, dc power supply), KiCad PCB software, LTSpice simulator, Spectre simulator, solder station, PTC Creo, Code Composer Studio, CLion, 3D printer.

2.8 Financial Requirements

Below are the core components needed to create the batteryless handheld game and their financial breakdown:

200x200, 1.54" E-ink Display	$20.50	1	$20.50
Kinetic Button Harvester	$11.15	4	$44.60
Mini Motor Generator	$9.99	1	$9.99
BQ25504 Boost Converter	$4.70	1	$4.70
TPS610995DRV Boost Converter	$1.84	1	$1.84
MSP430 Dev Board	$20.39	1	$20.39
TOTAL			**$82.68**

3 Design

3.1 Previous Work And Literature

The Battery-Free Game Boy is an example of something that comes close to realizing the vision of a batteryless game. However, the batteryless game boy does not have any mechanics which implement energy harvesting as a core aspect of the game, nor are there multiplayer aspects to the game. While this was an interesting exercise in converting a battery powered game into batteryless, our battery-free handheld game uses the lack of batteries and energy harvesting as a core aspect of the game, not as a supplement to normal batteries.

3.2 Design Thinking

One of the biggest limitations on our project is the fact that the game needs to have interaction between multiple devices. Because of this, we need to make sure that the multiplayer aspect of the game is present from the beginning, and that it's engaging. The other limitation is that the energy harvesting needs to be a part of the gameplay, and there needs to be an engaging way to make sure the user performs the energy harvesting.

After figuring out our problem and the limitations on our project, we began to list out several possible game ideas, and everyone brought their own takes on what would be fun and engaging gameplay that involved energy harvesting. One of our first ideas was a dungeon crawler, where players would be tasked with going through a dungeon and defeat a boss at the end. We improved upon this idea by adding challenges to the rooms that require energy harvesting, and making the final battle a RPG battle system. Our next idea was a card game, where two players go head to head with decks, and could earn cards by energy harvesting. We improved upon this idea by adding in a way for the player to interact with their opponent when it was not their turn, using energy harvesting to create more resources for them when it became their turn. Our next idea was a party game, similar to Mario Party, where we would have several minigames that involve energy harvesting with a board game feel to the game. Our last idea was a laser tag game, where users would have light gun-like devices they could tag other people with on campus, charging shots with energy harvesting. After discussing with the others, we also added the idea of objectives to capture around campus, as well as a website to login to track scores. We decided to pick the dungeon crawler as that was the idea most people were interested in.

3.3 Proposed Design

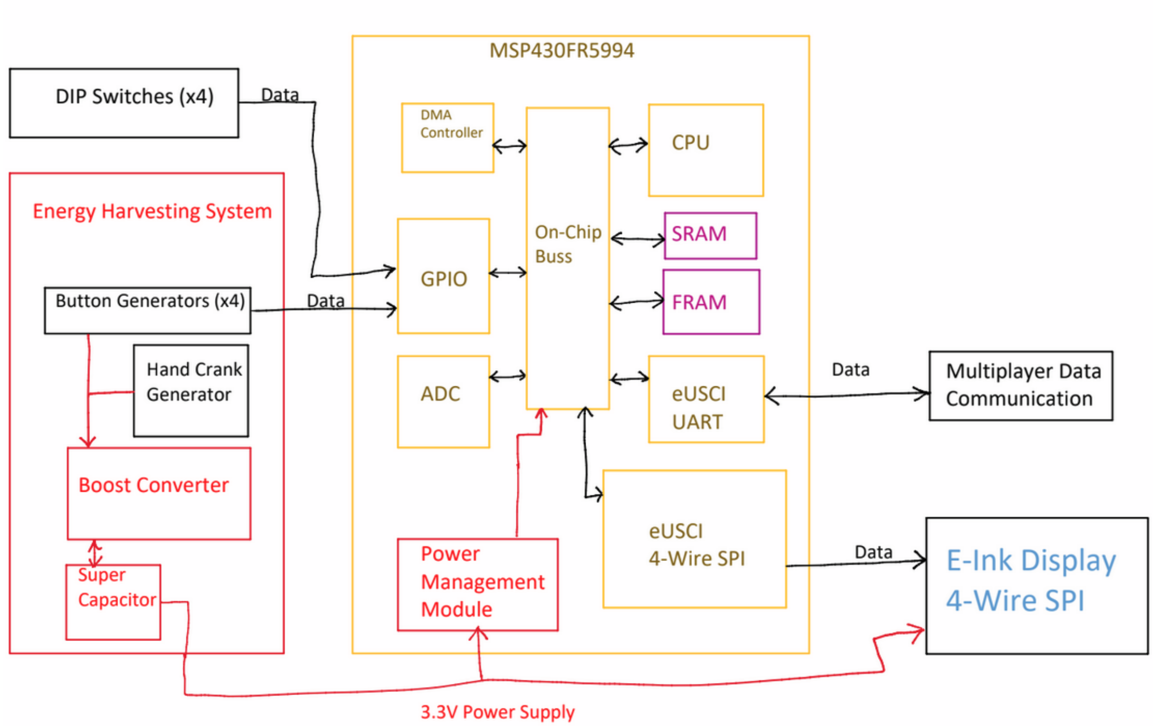


Figure 3.1: Functional Diagram

The design of the hardware consists of five main components shown in *Figure 3.1*: the energy harvesting system, microcontroller, DIP switches, E-Ink display, and multiplayer communication (shown above). The energy harvesting system powers the microcontroller which enables all aspects of the gameplay to take place. This system also transmits input signals to the microcontroller to be read for gameplay purposes which allows the microcontroller to integrate energy harvesting into the gameplay so that the system is continuously powered as the user progresses through the game (see the *Power System* section for more details on energy harvesting). This system is the sole power source for every component in our design.

All system inputs are transmitted specifically by the buttons and the crank generator and are sent to the MSP4030 microcontroller. The button signal will either be read as either read as pressed when a rising edge is detected, or not pressed when the signal is constant. The GPIO will read this signal as a binary input. The crank generator will produce a varying input value that changes with the speed of its rotation. The ADC on the microcontroller will read the voltage produced by the crank generator and allow the software to use this value within the gameplay.

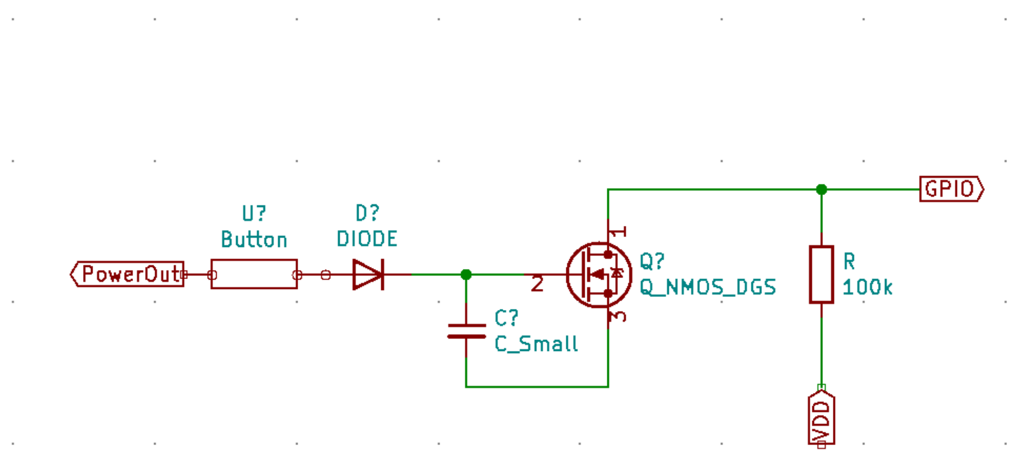


Figure 3.2: Button GPIO

The E-Ink display is the only user interface output for the device and instructions will be written to using a 4-wire SPI data communication. The E-Ink display will be rewritten using the charge stored on the supercapacitor. The data displayed will describe the current state of the device and information pertaining to changing to another state. This event occurs every time the user enters a new room, starts a new game, or the microcontroller loses the required power to run the game.

Multiplayer states are communicated between two devices over USB using a UART communication protocol. This allows both devices to transmit current state information and receive next state information between the two devices with minimal delay and power loss.

One of the most critical areas of our design for the batteryless handheld game is the energy harvesting system. There are three main parts to the energy harvesting system: the generating of the power, the storing of the power, and the outputting of the power to the rest of the design.

The generating of the power is done solely by using energy harvesting from human interaction. The two components used to do this are a mini motor generator and the kinetic button harvester. The mini motor generator produces three phase sinusoidal power when turned by the user via a hand crank. The kinetic button harvester produces a pulse of power whenever it is pressed and whenever it is released. Both of these components produce AC voltages however, since the system requires a DC voltage both need to be rectified as shown below.

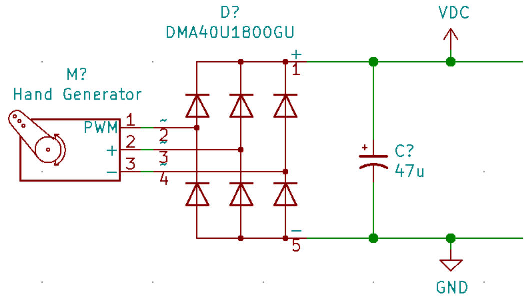


Figure 3.3: Generator Rectification

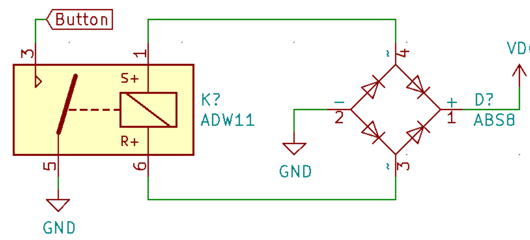


Figure 3.4: Button Harvester Rectification

The mini motor generator is rectified to a DC voltage by using a full-wave three-phase diode rectification circuit as shown and then a capacitor is added at the output to smooth out the ripples of the DC voltage. In a similar fashion, the kinetic button harvester is input into a full-wave single-phase diode rectification circuit so that it produces a short burst of constant DC voltage when pressed and released. Four of the kinetic button harvesters are used in the design so each will have its own rectification circuit and all will link to the same output with a smoothing capacitor.

The next major portion of the energy harvesting system is the storing of the power. Since one of the main functional requirements was that the handheld game had to be battery free a supercapacitor is used for the storage of any excess power. In order to properly manage the charging and discharging of the supercapacitor the BQ25504 Energy Harvester Controller will be used to take in the power produced by the mini motor generator as well as the kinetic energy harvesters. The BQ25504 is better than just hooking up the energy harvesters directly to the supercapacitor because it prevents overcharging of the capacitor and undercharging of the capacitor(a minimum voltage can be set). Most importantly the BQ25504 very efficiently boosts/bucks the input voltage from the energy harvesters at a range of 0.13V to 5.1V into a constant 3.3V so that the capacitor can always be charging even if the voltage on the capacitor is above the voltage that the energy harvesters produce. Below is the basic hooking up of the BQ25504 based off of datasheet examples (some of the basic components in the diagram may need to change once initial testing begins).

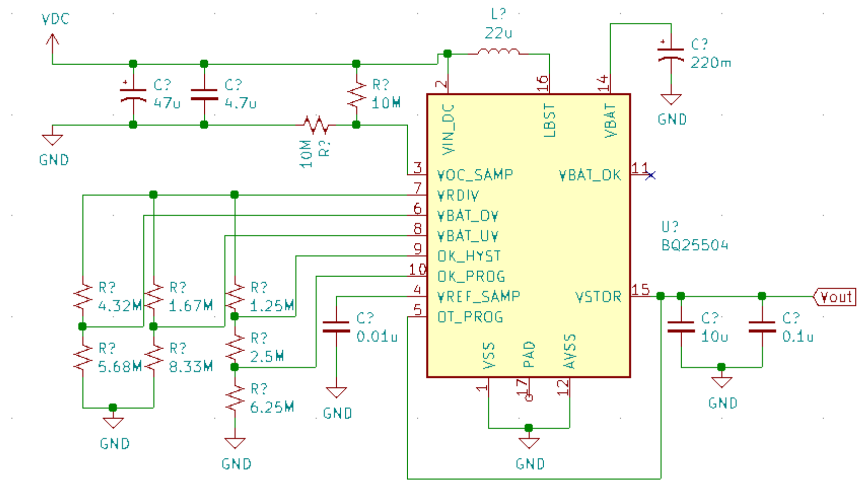


Figure 3.5: BQ25504 Basic Circuit Diagram

The last major section of the energy harvesting system is the boosting of the output voltage (Vout in the above schematic) to a constant 3.3V for use by the microcontroller and the E-ink display. Below is what that circuit looks like followed by a description of the boost converter.

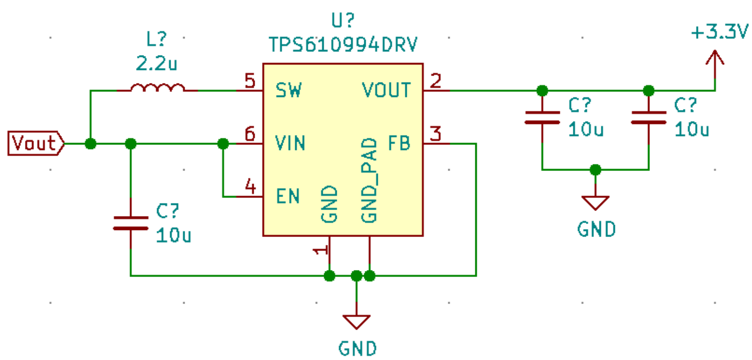


Figure 3.6: TPS610994 Boost Converter Circuit Diagram

The component used to boost the output voltage from the BQ25504 is the TPS610994 Synchronous Boost Converter with Ultra-Low Quiescent Current. The main advantage of this chip is that it has ultra low quiescent current draw of around 600nA which is ideal for our low power application. The boost converter also allows an input between 0.7V and 5.5V which is well within the super capacitor storage range. Lastly, the boost converter outputs a constant 3.3V which is the voltage used by the logic portions of the design. The schematic above is the non-tested setup of the boost converter based off of datasheet specifications.

Software will control user interface outputs and deterministically change the state of the game based on user input and power supply status. Critical information about the current state will be saved and utilized to maintain the state of the game through a low power event and restoration of power supply. This will allow players to maintain advancements throughout the game without the requirement of a constant power supply to the device.

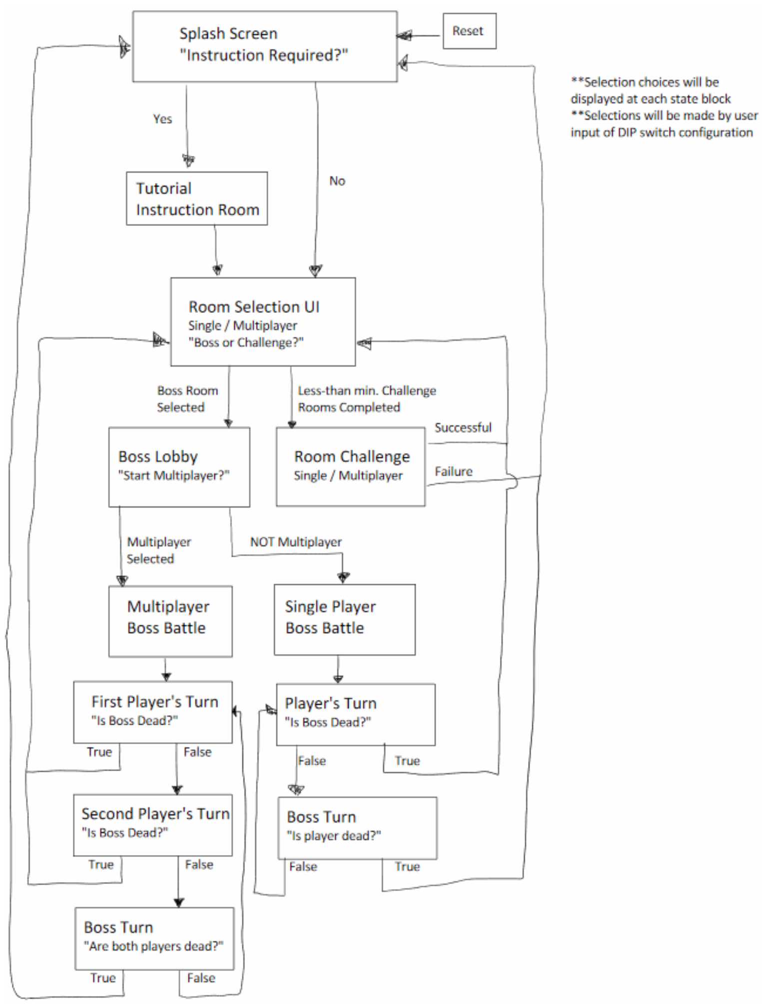


Figure 3.7: Software Flow Diagram

On game startup, the software flow (as seen in *Figure 3.7*) will show a splash screen, detailing a title as well as a prompt that asks the user if they want instructions for the game. If they do, the player will enter a tutorial room that walks through the basics of gameplay, like making choices with DIP switches and energy harvesting basics.

Once a player is ready to begin gameplay or if they don't want tutorial instructions, they are sent into the first room. The room selection UI will give the player the choice between three rooms. Once they select a room, a room challenge will begin. The room challenge will require a user to do an energy harvesting challenge, and if they are not able to generate enough power, they get the failure state and are sent back to the splash screen.

If a challenge is successfully completed, the player gains rewards that will make the boss fight at the end easier, and are sent back to the room selection UI. After a software defined number of challenge room loops, currently three, the player will see one of the rooms in the room selection UI changed to a boss lobby. The player can choose not to enter the boss lobby, and can go through the room selection UI/room challenge loop until the number of loops equals a maximum number of loops, currently 10, and the player is forced to choose the boss fight lobby as the next room.

The boss fight lobby provides the player a choice to engage in multiplayer, where they will wait for another player's device to enter a multiplayer state. *Figure 3.8* shows the communication of state information during multiplayer mode. The player can also choose to fight the boss singleplayer. The combat in the boss rooms will be a turn based RPG, where players will select moves from a list, using the DIP switches to make their choice. After they select their move and damage the boss's health, the boss will take a turn to fight back, decreasing the player's health. This will continue until all players' health reaches zero or the boss is defeated. Defeating the boss in multiplayer mode will allow players to continue a multiplayer version of the game loop where challenges will be tailored to multiple player interaction. Dying during the boss encounter in single and multiplayer modes will return the player to the splash screen.

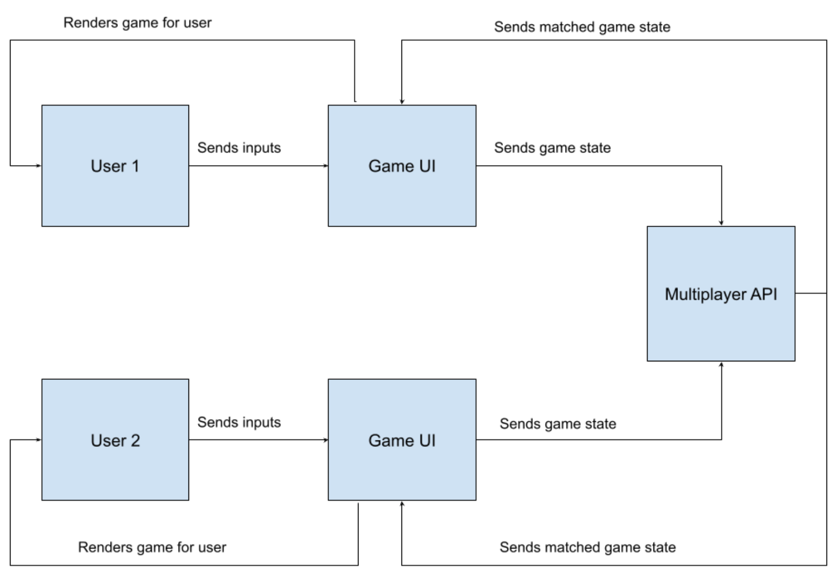


Figure 3.8: Multiplayer Communication Flow Diagram

When the MCU detects a low power event (as shown in *Figure 3.9*), software will save the state of the game to the fRAM before the device enters a low power state. In this state, a message will instruct the user to perform energy harvesting. As soon as a sufficient power threshold is achieved, the device will read the DIP switches and restore the saved state of the game, including rooms beaten, player character information, and current room state information.

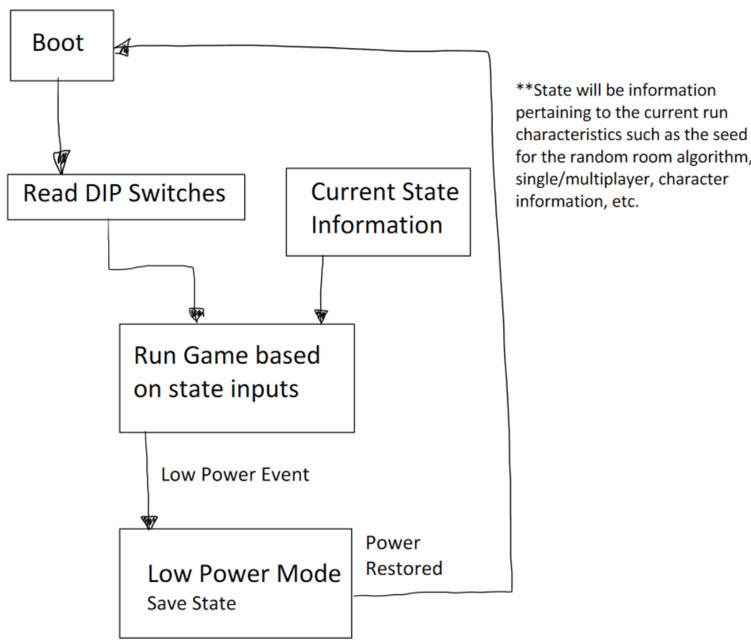


Figure 3.9: MCU State Flow Diagram

3.4 Technology Considerations

Due to lack of batteries in the project there were many trade-offs made in technological devices chosen. The two main devices discussed were the display and the energy harvesting instruments. The display had to take as little energy as possible since the only energy available was from the energy harvesting, thus an E-ink screen was chosen since it takes practically no power in standby and moderate power when changing. A major downside of the E-ink screen is that it cannot be brought into direct sunlight which limits the device's operational environment to indoors only. However, due to the low power consumed overall it was practically the only option available.

Choosing the energy harvesting instruments was a more difficult task due to the various forms of energy harvesting available. As a base, kinetic button harvesting was a necessity since the user would almost certainly need to be able to press buttons to move around. However, the kinetic button harvesters did not produce enough energy to power the system over multiple screen changes and other forms were looked into as supplements. Solar cells were a consideration due to their low cost and wide spread use, however one downside of using solar cells is that the energy production is mostly based on the surface area of the cell. Since the game had to be handheld the overall size of the solar cell needed to be reasonably small and thus not produce an abundance of energy. Upon testing we concluded that solar cells would not be able to contribute a reasonably useful amount of energy and therefore we would not use them in our design. Alternatively, the small AC motor crank generator would serve as a capable source to meet the majority of our design's power demands, and it would also serve as a new component to integrate into the device's gameplay.

3.5 Design Analysis

Currently, no major components of the project have been implemented. The only parts that have been analyzed are the rectification of the mini motor generator and the rectification of the kinetic energy harvesters. Both of these components worked as expected, however one issue that came up was a possibility of the voltage produced by these components being too high if the user cranks the shaft very fast. If the voltage is above 5.1V the BQ25504 control chip will burn out which would be catastrophic to the operation of the device. One idea to modify this aspect of the design to ensure the voltage stays below 5.1V is to use a zener diode set to 5V in parallel with the sources to sink any excess voltage. Another idea would be to use another voltage regulator before the BQ25504 to ensure the input never exceeds 5.1V.

When we begin to implement the separate components of our design we are anticipating to face a few new challenges. In order to minimize the number of setbacks we face, we are planning on testing each component immediately after it is integrated with the microcontroller and the gameplay software. This will make the debugging process more efficient and allow us more time to implement our proposed stretch goals.

3.6 Development Process

We will be using the waterfall development process for hardware because most of our tasks can only be achieved if the tasks before it has been completed. It would be difficult to try to develop our project any other way due to this cascading depence of tasks.

In terms of software development, we will be using the agile development model. The game logic itself will be developed as a standalone project, and then we will use APIs to get the hardware inputs controlling the game, and the e-ink displays to display the output.

3.7 Design Plan

The design plan for this device will incorporate two teams working in tandem. The Power Systems team will be responsible for developing the energy harvesting circuitry that will supply power to the device and provide physical interaction to the user. The microcontroller team will develop the firmware for interpretation of physical I/O and develop software enabling a user interface and gameplay.

The teams will need to coordinate deliverable deadlines so that dependencies are produced before they are required. The power system team will design a user input detection circuit incorporating energy harvesting with buttons and crank generator.

During this time the microcontroller team will design a firmware interface for the E-Ink display and software for a graphical user interface. These components will enable user interaction with the device and will be invaluable to testing the system as components are added.

The next stage will be for the power system team to design circuitry to regulate and power the device without external assistance. The microcontroller team will design gameplay features including random environment generation, character mobility, and room challenges. These components will fulfil the requirements for the device being batteryless and having a unique gameplay experience that incorporates energy harvesting.

The final stage will be developing multiplayer functionality. Both teams will work together designing a method of communication between two devices and software that will allow two users to interact in a common environment. This will meet the requirements of multiplayer functionality.

4 Testing

4.1 Unit Testing

Energy Harvesting

1. Crank Generator & Rectification
    a. Setup crank generator in a circuit with full-wave 3-phase rectification
    b. Connect capacitor to output of rectifier
    c. Have a user crank generator at different RPM rates including smallest, average, and maximum RPM.
    d. With an initial state of zero volts on the capacitor, crank generator at different speeds for x amount of time.
    e. Measure the final voltage (V) on the capacitor when the time limit is reached.
    f. Obtain the energy produced using $E = 0.5CV^2$ and the resulting average power using $P = \frac{E}{time}$.
    g. Repeat steps a through f using different speeds as well as different users to obtain a set of power outputs.
    h. Average the set of power produced from each trial to obtain a final average power for the crank generator.
2. Energy Harvesting Button & Rectification
    a. Set up an energy harvesting button into a full-wave diode rectifier.
    b. Connect the output of the rectifier to a capacitor (C).
    c. Start at zero volts on the capacitor and push the button as fast as possible for x amount of time.
    d. Measure the final voltage (V) on the capacitor after time has stopped.
    e. Obtain the energy produced using $E = 0.5CV^2$ and the resulting average power using $P = \frac{E}{time}$.
    f. Perform steps a through e for multiple trials using various capacitance values and amount of time to obtain a set of power produced.
    g. Average the set of power produced from each trial to obtain a final average power for the energy harvesting button.

Hardware

3. Speed Detecting Circuit
    a. Connect speed detecting circuit output to voltmeter.
    b. Ensure as the speed of the generator increases, the voltage read by the voltmeter also increases.
    c. Ensure maximum output voltage is below 3.3V (for voltage protection to the microcontroller).
4. E-Ink Display

   a. Determine the amount of energy needed to load an image onto the display. Specifically, what percentage of the super capacitor's capacity is used every time a screen update occurs.
   b. Determine the maximum amount of screen writes we can achieve with a fully charged supercapacitor.
   c. Measure the power usage when updating a different percentage of the display pixels. For example, how much power is used when all the pixels change from white to black vs. how much power is used when only half the pixels change from white to black (partial refresh versus full refresh).

Microcontroller

1. GPIO:
   a. Set all GPIO pins to inputs.
   b. Provide a logic high to all GPIO pins to confirm read functionality.
   c. Set all GPIO pins to outputs.
   d. Set GPIO pins to set all pins to logic high for 1 second then all pins to logic low for 1 second.
   e. Confirm write functionality of all GPIO pins by using a voltmeter to probe the voltage at each pin and observe the oscillation between logic high and logic low.
2. SPI:
   a. Hook up microcontroller SPI signal wires to a slave device.
   b. Perform a writing to the slave device and confirm correct sending of data to test SPI is functioning correctly.
3. UART:
   a. Connect the UART communication wires from the microcontroller to a UART compatible device.
   b. Send a test signal to a device from the microcontroller and from a device to the microcontroller to confirm correct sending of data and the working of the UART.
4. CTPL:
   a. Run a program on the MSP430 that slowly decreases the blink rate of the onboard LED using GPIO.
   b. Disconnect MSP430 from power.
   c. Check that the LED remains off to confirm that MSP430 no longer has access to power.
   d. Reconnect the MSP430 to power.
   e. Confirm the MSP430 enters the saved state on power loss by observing that the LED blink rate is the same as when the power was disconnected in step b.

Software

1. GUI: User input and software output testing

    a. Make sure each room and button prompts render correctly with all of their components and check components positions.
    b. Test and confirm each input combination for the room selection UI.
2. Environment Generation: Random room testing
    a. Test for paths from the start room to the end.
    b. Validate each room has the correct components.
3. Room Challenge:
    a. Test success state for each room and check if players are updated.
    b. Test failure state for each room and check if players are updated.
    c. Make sure each room challenge is rendered correctly.
4. Multiplayer API:
    a. Validate messages sent via UART are the same as messages received via UART.
    b. Check after two players finish their turn that the boss health and player health is the same on both devices and is calculated correctly from both player's turns.
    c. API is able to send game state as a package.
5. Power loss:
    a. Check game state is sent to fRAM on power loss.
    b. Check that the game state, including the players' stats, room information, boss stats, and current turn in the boss fight, can be restored from fRAM upon recharging from power loss, and are exactly the same as the game state before power loss.

4.2 Interface Testing

1. E-Ink SPI:
    a. Perform a write to the E-ink display from the MSP430 using SPI communication.
    b. Confirm the E-Ink receives the SPI signal properly by observing that the display matches the intended image as detailed by software.
2. Multi device UART:
    a. Connect two MSP430 microcontrollers together via UART communication.
    b. Send a series of data between the microcontrollers and confirm that the data matches expected values using comparisons via code as well the debug window in Code Composer Studio.
3. Power to Capacitor:
    a. Crank generator and press buttons
    b. Confirm each component is providing power to system by observing voltage increase on super capacitor
4. Power to System:
    a. Fully charge the supercapacitor using a DC power supply.
    b. Slowly drain the energy harvesting system (the power system of our device) by connecting a load (simulated system) to the output.

    c. Confirm that 3.3V is output to the load over the full working range of the supercapacitor.
5. Speed Detecting Circuit with Microcontroller:
    a. Connect the speed detector circuit to an ADC input on the microcontroller.
    b. Use the ADC to measure the voltage from the detector circuit and confirm correctness with voltmeter.
6. Buttons and DIP switches:
    a. Setup energy harvesting buttons and DIP switches into their correct GPIO pin.
    b. Confirm the logic high and logic low reads by the microcontroller of the DIP switches and confirm the rising edge reads by the microcontroller of the energy harvesting buttons.

4.3 Acceptance Testing

The functional requirements consist of three main pieces, the device is powered only from energy harvesting, energy harvesting is part of the gameplay, and the device supports some multiplayer functionality. Energy harvesting being part of the gameplay, and devices supporting multiplayer functionality will both be clearly demonstrated when the client plays the game and sees both of these aspects incorporated into gameplay. The device being powered only from energy harvesting can be demonstrated by showing to the client that there is clearly no battery or charging port as part of the hardware, meaning the power must be coming from the energy harvesting devices.

The nonfunctional requirements have tests designed for each case. Portability shall be tested by seeing if the device can fit into a normal backpack. Intuitiveness will be tested by seeing if a user who has not played the game before can play the game from start to finish with little to no outside help. For the game running through a room challenge state, we will run at least ten tests where we start a room challenge. After energy harvesting, we should be able to get through the room challenge state and be sent back to the room selection UI at least nine times out of ten. Lastly, for the low power screen, every time the device reaches low power, it should switch to the low power screen.

4.4Results

Currently we have not completed most of the unit tests, nor any of the interface testing. This will be completed in the second semester of senior design. The two unit tests we have completed are the Crank Generator & Rectification and the Energy Harvesting Button & Rectification. Below reiterates the test procedures, our data collected, and our results. The main takeaway from this initial testing is the proof of feasibility of the battery free handheld game.

In order to prove feasibility we took the route of being able to update the E-ink display as confirmation that the making of the battery free game design we had was feasible. The reason being that the E-ink display, by a large margin, will be the largest draw of power for our system. So if we are able to comfortably update the E-ink display often strictly from energy harvesting devices then that proves our design is feasible.

Based on the data below the energy harvesting buttons produce an average of 1 mW and the mini motor generator produces an average of 10.6 mW while being used. From these values the energy harvesting button will be able to update the E-ink display every 50 seconds and the mini motor generator will be able to every 5 seconds. Thus, while the energy harvesting buttons may not be super useful in generating the majority of our power, the mini motor generator will be able to easily produce enough power which confirms that our design is feasible. Moreover, if needed we can introduce a gear ratio to increase the speed and thus power production of the hand generator for even more power production which further confirms the proposed design with function.

Mini Motor Generator

Test Procedure: Set up mini motor generator into a full-wave 3-phase diode rectifier. The output of the rectifier is connected to a capacitor. Start at 0V on the capacitor and turn the generator using drill or hand at various speeds for x amount of time. Measure the voltage after time has stopped. The average power production of the generator is equal to energy divided by time (in seconds).

Trial Number	Drill Speed	Capacitance(uF)	Initial Voltage(mV)	Final Voltage(V)	Time(s)	Energy(mJ)	Power(mW)
1	Hand Turned	6000	100	4.3	5	52.92	10.584
2	Hand Turned	6000	100	6.73	10	131.8707	13.18707
3	Hand Turned	6000	100	7.45	20	162.0675	8.103375
4	slow(~60rpm)	6000	2	1.38	5	5.696652	1.1393304
5	medium	6000	2	5.7	5	97.401612	19.4803224
6	fast	6000	20	6.31	5	118.6923	23.73846

Figure 4.1: Mini Motor Generator Tests

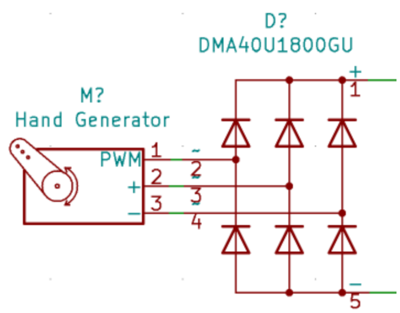


Figure 4.2: Mini Motor Schematic

Results		
Max DC voltage on capacitor after long time turning with drill		
6.7		
Average Power(mW)		
10.624815		
Can Easily introduce a gear ratio and get much higher power output since it is dependent upon rpm		
Time to produce enough energy to update E-ink once (based off of datasheet)		
Average Power (Hand Crank)	Energy to update E-ink once (mJ)	Seconds
10.624815	52.8	4.969498292

Figure 4.3: Mini Motor Test Results

Energy Harvesting Buttons

Test Procedure: Set up an energy harvesting button into a full-wave diode rectifier. The output of the rectifier is connected to a capacitor. Start at 0V on the capacitor and push the button as fast as possible for x amount of time. Measure the voltage after time has stopped. The average power production of someone pressing the button fast (our use case for a skill check) is equal to energy divided by time (in seconds).

Trial Number	Capacitance(uF)	Initial Voltage(mV)	Final Voltage(V	Time(s)	Energy(mJ)	Power(mW)
1	2000	10	2.52	5	6.3001	1.26002
2	2000	20	3.8	10	14.2884	1.42884
3	2000	20	4.06	20	16.3216	0.81608
4	4000	10	1.65	5	5.3792	1.07584
5	4000	5	2.47	10	12.15245	1.215245
6	4000	10	3.51	20	24.5	1.225
7	4000	2	1.46	5	4.251528	0.8503056
8	4000	2	2.28	10	10.378568	1.0378568
9	4000	2	3.07	20	18.825248	0.9412624
10	6000	12	1.15	5	3.885132	0.7770264
11	6000	10	1.88	10	10.4907	1.04907
12	6000	2	2.58	20	19.938252	0.9969126

Figure 4.4: Energy Harvesting Button Tests

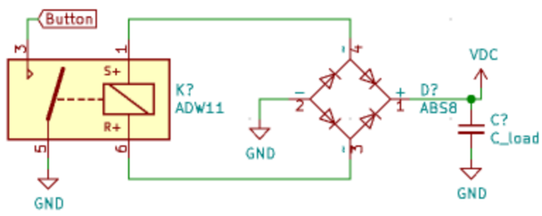


Figure 4.5: Button Schematic

Results			
Max DC voltage on capacitor after long time pressing			
4.8V			
Average Power(mW)			
1.056121567			
No Clear increase or decrease of power output when capacitance increases			
Time to produce enough energy to update E-ink once (based off of datasheet)			
Average Power (Hand C	Energy to update E-ink once (mJ)		Seconds
1.056121567	52.8		49.99424467

Figure 4.6: Button Test Results

5 Implementation

Next semester we plan to start our process by first completing all our unit testing. From there we will add the final details to our schematic, updating parts based on unit testing if needed, and order all the necessary parts to build an initial prototype. Once we order our parts we will begin to connect each device together and perform interface testing to confirm all functionality. Once all functionality has been confirmed the initial prototype will be put together and a mockup of the acceptance testing will be performed. Once the acceptance testing works fairly well with the initial prototype we will design a PCB for our schematic, order it and create our final product. Throughout these stages we will find any discrepancies or issues that arise between devices and develop strategies to overcome them.

Software development of the game will happen in parallel with the hardware development. The development strategy we will implement will be the Agile development style. This process will allow the core functionality of the game to be implemented and tested over time while adding degrees of complexity throughout the development cycle. We will begin by developing the user interface displays and integrating them with the e-ink display functionality for testing. The next stage will develop the random room generation algorithm and integrate it for testing. The third stage will develop user input detection for controlling the game. This will be integrated with the energy harvesting power system for testing. Fourth we will develop these gameplay functionalities by evolving them to include a character system which will act as a customizable, upgradeable avatar for the player and room variation to include challenges, the boss lobby, and boss battle rooms. The final stage will develop multiplayer functionality and communication for integration and testing. Once these core requirements are implemented we can continuously develop the complexity of each element to the game as time permits.

6 Closing Material

6.1 Conclusion

This semester was focused on mainly the research and feasibility of creating a batteryless handheld game. So far we have developed the idea of our game, selected and purchased several of our components, completed a couple unit tests, and created several diagrams and schematics for use in the next semester with the initial prototype. We have taken steps to prepare ourselves for actual implementation and design of our project so we can get to work immediately in the Fall. Most importantly we have proved that our device should be able to be powered solely from energy harvesting devices through the testing of the mini motor generator and energy harvesting buttons.

6.2 References

Mastellari , Alessandro. "A Quick Guide to Powering Smart Objects with Kinetic Energy Harvesting." *Avnet*, 22 Jan. 2019, www.avnet.com/wps/portal/abacus/resources/article/a-quick-guide-to-powering-smart-objects-with-kinetic-energy/.

"MSP430FR599x, MSP430FR596x Mixed-Signal Microcontrollers." Texas Instruments, Jan. 2021. https://www.ti.com/lit/ds/symlink/msp430fr5994.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1614352864965

"MSP430 SPI Peripheral." *ARGENOX*, www.argenox.com/library/msp430/msp430-spi-peripheral-chapter-9/.

"Rectification of a Three Phase Supply Using Diodes." *Basic Electronics Tutorials*, 3 Sept. 2018, www.electronics-tutorials.ws/power/three-phase-rectification.html.

"200x200, 1.54inch E-Ink Display Module." *Waveshare*, Apr. 2021, www.waveshare.com/1.54inch-e-paper-module.htm.

"1.54inch e-Paper Module." *1.54inch e-Paper Module - Waveshare Wiki*, Apr. 2021, www.waveshare.com/wiki/1.54inch_e-Paper_Module.

"Wireless Switches – Generator ." Mouser Electronics, 30 Apr. 2015.https://www.mouser.com/datasheet/2/833/Energy_Harvesting_Generator_US-1075748.pdf

"BQ25504Ultra Low-PowerBoostConverterWith BatteryManagementFor EnergyHarvesterApplications." Texas Instruments, Nov. 2019. https://www.ti.com/lit/ds/symlink/bq25504.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1619382959060

"TPS6120xLow Input VoltageSynchronousBoostConverterWith 1.3-A Switches." Texas Instruments, Mar. 2007. https://www.ti.com/lit/ds/symlink/tps61202.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1619382985859

6.3 Appendices

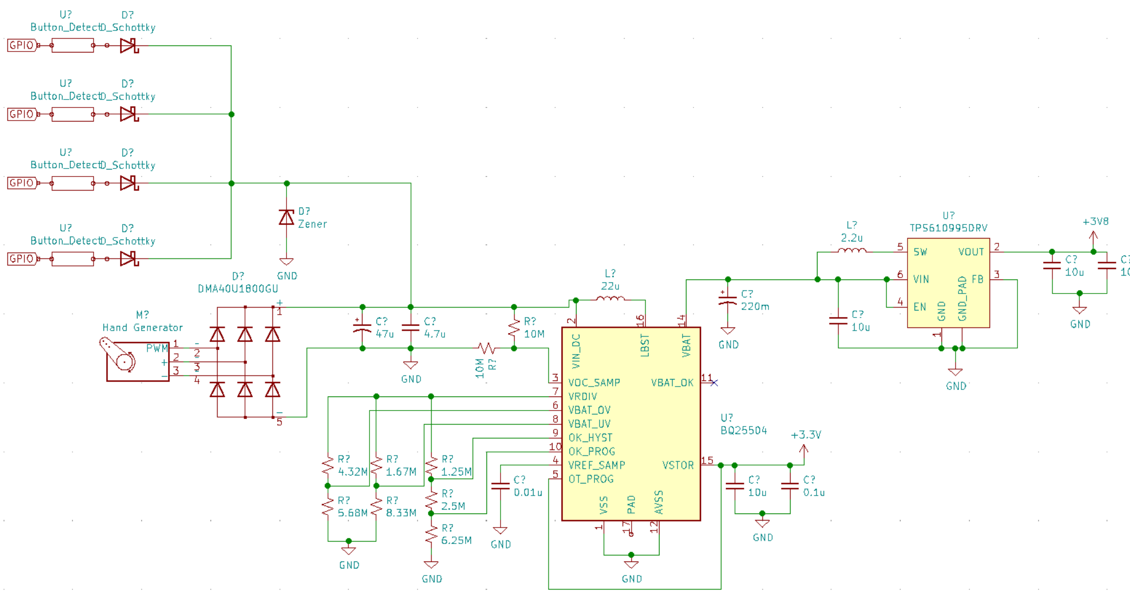


Figure A1: Full Energy Harvesting and Power Schematic