Hand Cranked Charger
Design Document

Team Number -
14

Team Members -

Shreyasi Ray (ray17)
Achyut Agarwal (achyuta2)
Rubhav Nayak (rubhavn2)

TA -
Matthew Qi

02/20/2023

ECE 445 - Senior Design Laboratory

Contents

Introduction

Problem

In today's day and age, the dependence on technology and devices keeps increasing by the day. The reliance on technology (especially our mobile phones) has risen to an all-time high, with about 90.4% of the global population owning a mobile phone. This would also lead to the claim that individuals have become very conditioned to using devices for every aspect of their lifestyle. Even in times of emergency, we find the need to be able to use our devices to call, text or take care of basic services and utilities by using a host of applications. Due to this increasing subservience to technology, powering such electronic devices also finds the highest priority.

This problem has led to an increased need for charging points to be able to provide power to their devices, and what we often notice is that people do not always have access to a power outlet. One does not always think of sufficiently charging their devices, and sometimes devices lose power rapidly, so often, we find ourselves helpless when our devices are running out of charge. In areas where electricity is also scarce, one needs an alternative to charge their devices, since charging ports will not work in such situations. In times of emergency in low-charge instances, we find ourselves helpless and not being able to contact anyone.

There are products in the market that tackle this issue, but often they come with many features that are pretty unnecessary for the average consumer. Often, these extra features also make the product bulkier and result in increased cost. They have a very simple circuit design which involves the hand crank constantly charging the battery and then the battery charging the USB output. This has many long-term problems since it causes constant battery usage, increasing the number of battery cycles used, which decreases longevity. Furthermore, the conversion of energy and transmission loss means these products lose most of the energy from the hand crank and are not able to provide electricity efficiently enough to charge the device in a reasonable amount of time.

In order to solve this problem, we plan on designing a product that is able to generate sufficient electricity to charge devices by a few percentage points, which is enough to give them some more time to run, decreasing the general low-battery related anxiety that persists among individuals, and we would do so with very limited energy loss by providing direct energy to the USB Out from the Hand Crank.

Solution

Our device is a portable hand-cranked charger that enables the user to generate electricity with the help of their own kinetic energy. This would charge their device so they can use it in times of emergency, be that calling for services, or for transportation as per their requirement. This hand-crank generator is the solution to a problem that is still prevalent as many parts of the world still lack all-day electricity and many travelers are often unable to charge their phones in such locations.
The product will include a small 1000mAh battery that can be charged via the hand crank or USB Input should the user be anticipating that it may be required later down the line, therefore acting also like a power bank as well. This product is meant to be affordable and compact enough to be carried around with the user, and the system being automated enables the usage of this product efficiently without having to walk through any complicationThis product is able to provide power at any location, enabling the devices to run long enough to be of use during emergency situations.

Visual Aid

The product would be ideally used in such a configuration as shown in the figure below.

The hand-cranked charger gets as input either the user's own kinetic energy through hand cranking. Through the hand crank, users can crank at an optimal rate to make sure their device can be charged to a sufficient point, as per their requirements. The product provides enough instruction to the user to help them understand what the ideal rate of cranking is.
There is an alternate USB power source that will then be converted and regulated in our product to produce a steady output to the mobile or other USB device to be charged.
Since there are two input methods, we need to ensure that we handle the case when both inputs supply energy. This is to ensure that the product is as easy to use as possible. The user would not need to perform any manual adjustments when using it, as everything would be automated.

Therefore, we make sure the product is able to handle and work with the different possibilities of input and output such as: both inputs but no output, one input without any output, both inputs as well as the output present and many other such combinations.

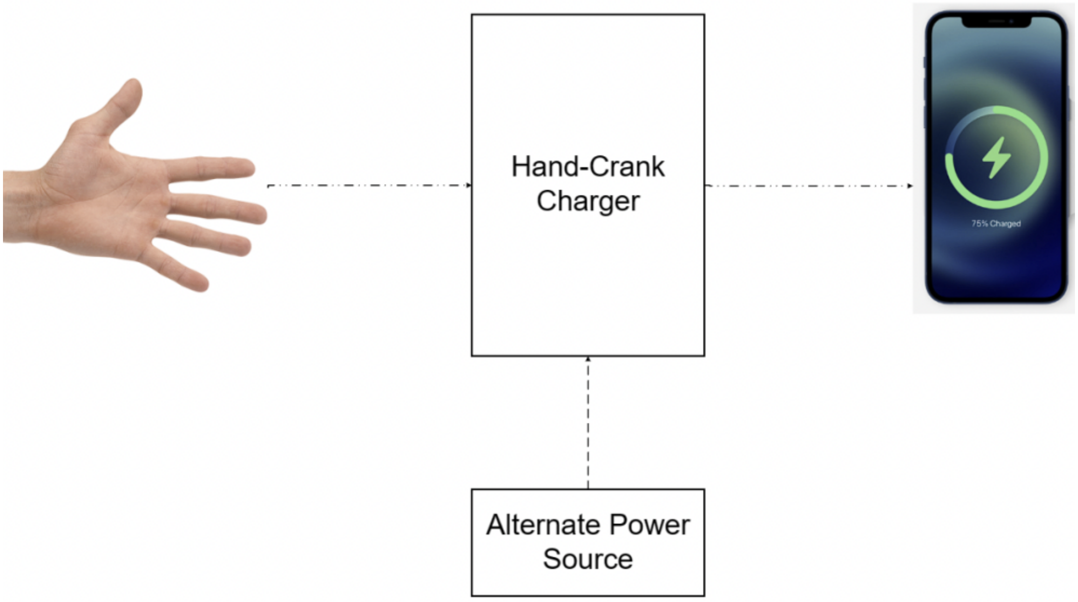

Figure 1. Overall Use Case

Note: The dashed lines represent direct physical connections

High-Level Requirements

- ☐ All of our power MUXs work correctly with the use of our microcontrollers pinouts acting as select signals, to ensure a change of flow of current happens within 50ms in order to charge the right component at the right time.

- ☐ Our product should work autonomously without any user input in what charges and when, the Microcontroller should be able to get the correct inputs and accordingly send the MUX the right signals to ensure our product works without the need of human interaction.

- ☐ The microcontroller should be able to successfully cut off supply to the battery when the battery is fully charged, and also send current to the battery when the battery level is low

- ☐ The microcontroller is able to accurately calculate the voltage output of the motor, to ensure users crank at the maximal efficiency rate. This speed suggestion is displayed on the product through the display, so that users can follow this and crank accordingly. The display should also allow users to track the battery level of the internal battery of the product. The display is able to give the users a more comprehensive method of understanding whether they need to crank at a lower or higher speed, without knowing the deeper specifics and intricacies of the product.

Design

Block Diagram

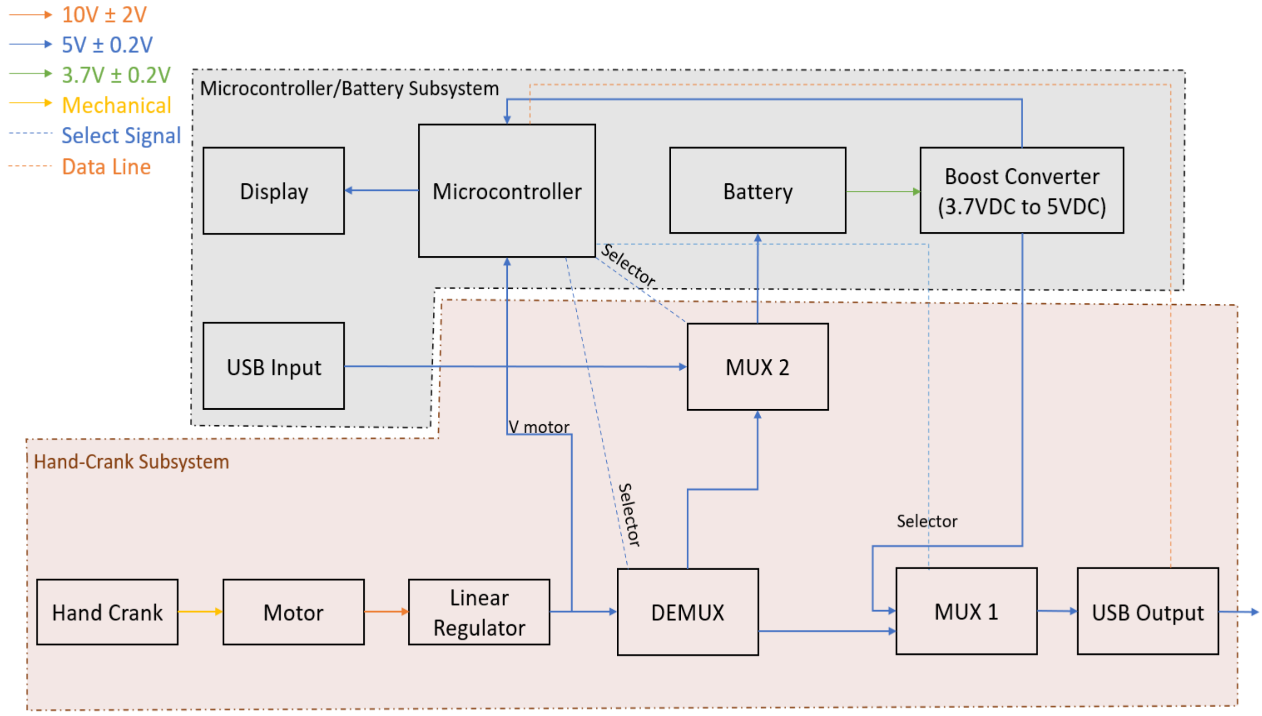

Figure 2. Block Diagram

This product mainly consists of two subsystems: the **microcontroller or battery subsystem** and the **hand-crank subsystem:**

- ☐ The microcontroller subsystem determines the correct flow of electricity from the input (USB In or Hand Crank) to the necessary outputs (Battery or USB Out). Furthermore, it contains the Display Unit which tells the user information about the current status of the device. Additionally, it handles battery management ensuring the battery level is within our desired range.
- ☐ The hand-crank subsystem is responsible for converting the kinetic energy of hand cranking to electricity through the use of a motor, it also steps down the voltage to a more usable 5V as that is what is required by USB Outputs. This also includes the entire Relay and Power MUX setup which is critical to the automation of our product.

Truth Table

The following table represents all cases for our product based on 5 parameters.

H C → Hand Crank Turned (1) or not (0)

USB In → USB Input supplied power (1) or not (0)

USB Out → USB Output has device connected (1) or not (0)

Batt Low → Indicates if the battery is low (1) or not (0)

Batt High → Indicates if the battery is high (1) or not (0)

H C	USB In	USB Out	Batt Low	Batt High	Outcome	Consequence
0	0	0	0	0	Sleep	
0	0	0	0	1	Sleep	
0	0	0	1	0	Sleep	
0	0	0	1	1	Sleep	
0	0	1	0	0	Charge from Battery	
0	0	1	0	1	Charge from Battery	
0	0	1	1	0	Display LOW	Goes to Sleep
0	0	1	1	1	Not Possible	
0	1	0	0	0	Charge Battery	
0	1	0	0	1	Stop flow at MUX 2	
0	1	0	1	0	Charge Battery	
0	1	0	1	1	Not Possible	
0	1	1	0	0	Charge Battery	Battery Charges USB Out
0	1	1	0	1	Stop flow at MUX 2	
0	1	1	1	0	Charge Battery	
0	1	1	1	1	Not Possible	
1	0	0	0	0	Hand Crank Charges Battery	
1	0	0	0	1	Hand Crank to USB Out	
1	0	0	1	0	Hand Crank Charges Battery	
1	0	0	1	1	Not Possible	
1	0	1	0	0	Hand Crank to USB out	
1	0	1	0	1	Hand Crank to USB out	
1	0	1	1	0	Hand Crank to USB out	
1	0	1	1	1	Not Possible	
1	1	0	0	0	Hand Crank Charges Battery	Stops USB in Flow at MUX 2
1	1	0	0	1	Hand Crank to USB out	Stops USB in Flow at MUX 2
1	1	0	1	0	Hand Crank Charges Battery	
1	1	0	1	1	Not Possible	
1	1	1	0	0	Hand Crank to USB out	USB In charges Battery
1	1	1	0	1	Hand Crank to USB out	Stops USB in Flow at MUX 2
1	1	1	1	0	Hand Crank to USB out	USB In charges Battery
1	1	1	1	1	Not Possible	

Truth Table

Physical Design

In the images below, we see the expected physical design of the hand-crank generator. The crank should be long enough for the ideal crank speed, for maximizing user efficiency as well as increasing user comfort.

The casing of the generator is made out of plastic, rather than metal to avoid safety hazards through conduction and electrocution. However, the base of the product is made of metal to make sure the inner components have a strong foundation to be mounted on. This also gives users the benefit of placing the product on many types of surfaces without having to worry about degradation of the material. The weight of this also assists in keeping the product upright and in-place while being cranked.
This all-encompassing casing protects all the inner components from any kind of external mechanical abuse, so that damage can be avoided as much as possible.

The case surface also has a display that shows users optimal rate of cranking, helping them crank efficiently with guidance while telling them to crank slower or faster. It also displays battery level percentage to help users keep track of how much power the battery still stores, so they can be prepared for future requirements and situations.

This is a rough idea of what the product should look like in its final form. We expect that in the prototyping phase, it will be a bit larger, and also may have some changed placement.

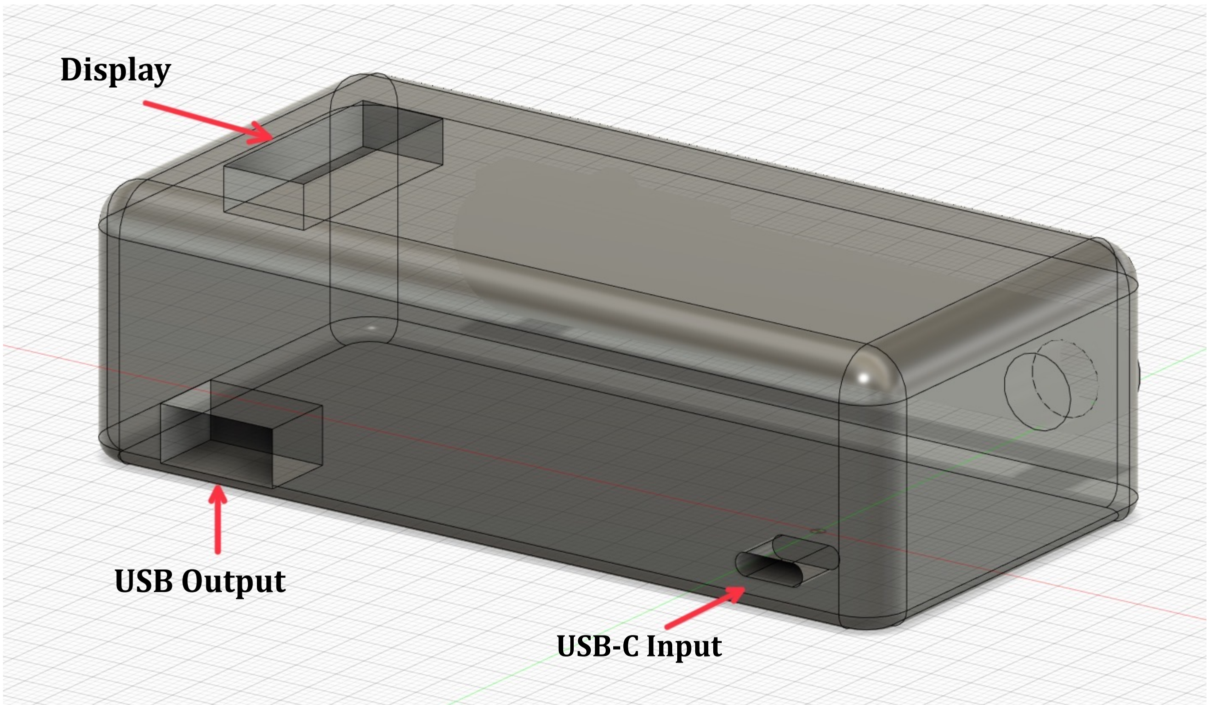

Figure 3. Charging side

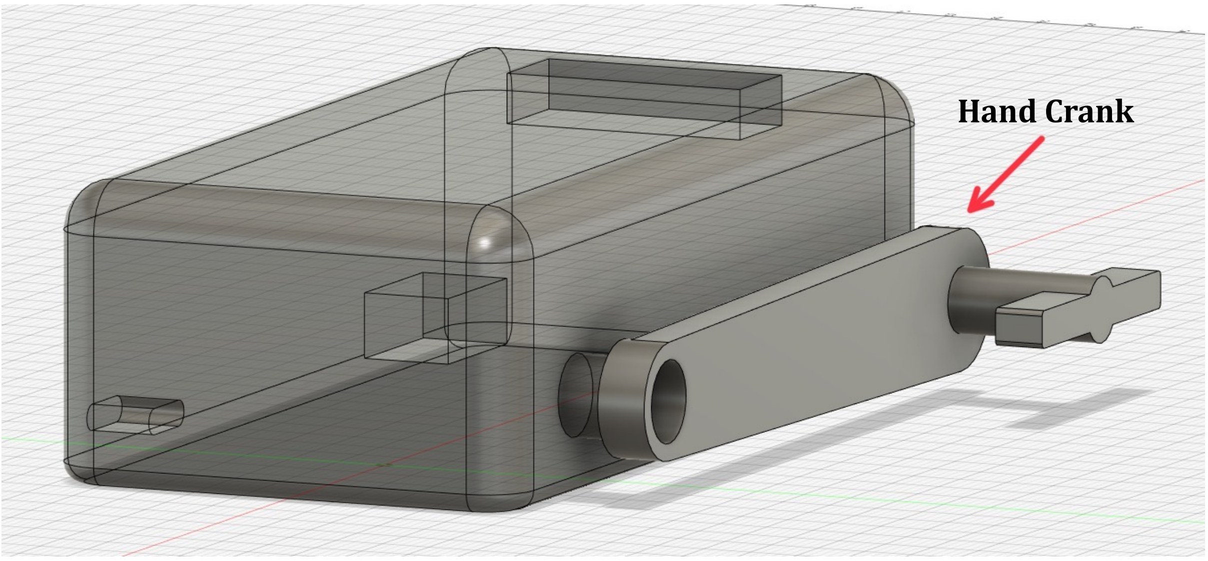

Figure 4. Hand Crank side

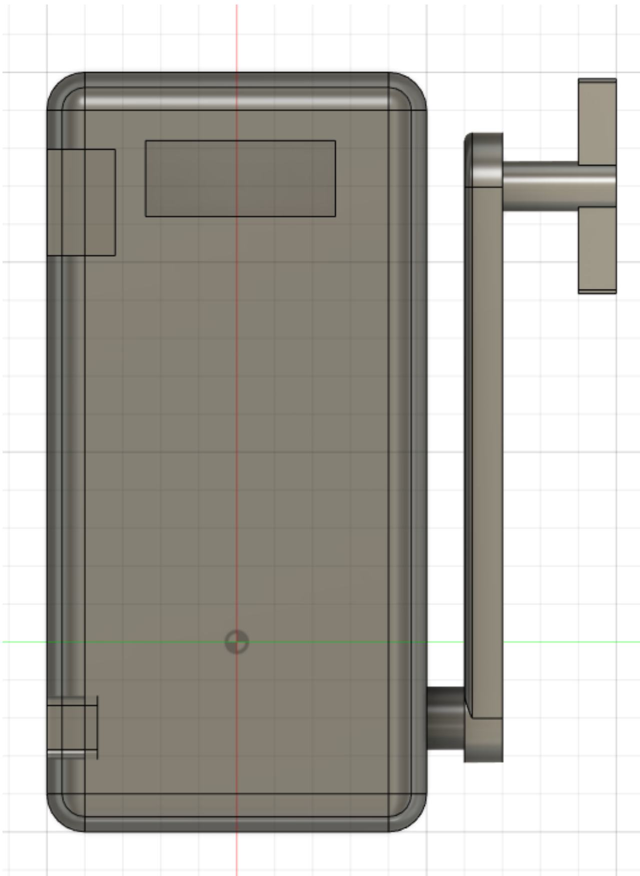

Figure 5. Top View (Display)

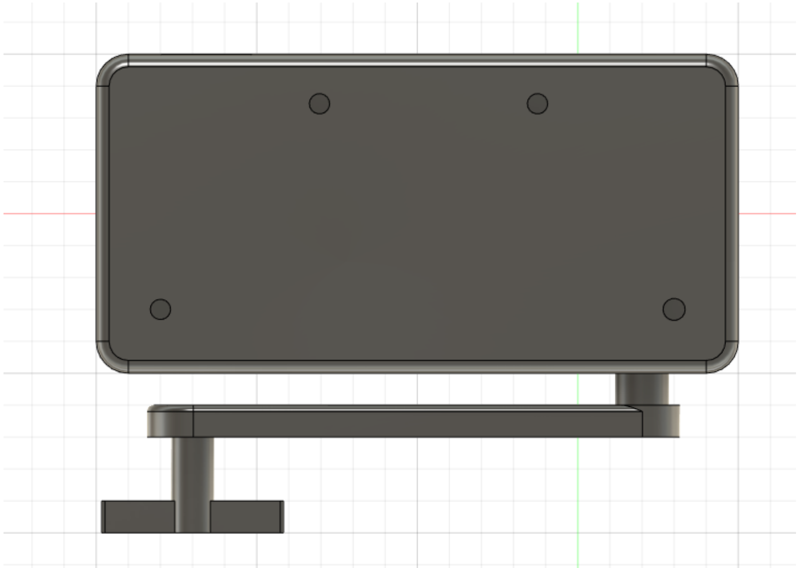

Figure 6. Bottom View

Subsystem 1

Microcontroller/ Battery Subsystem

This subsystem consists of the microcontroller, the battery, the circuitry that involves the boosting of the battery voltage as well as the USB IN Charging Module.

The subsystem is in charge of enabling the right circuitry path in the hand-cranking subsystem as the microcontroller has pinouts that will pass the signal to the Power MUXs that are being used. Furthermore, voltage readings help us observe if we can see the crank being turned at the optimal RPM or not, and this would allow us to indicate to the user if they should go faster or slower with the help of the display.

We will be using the ATmega328p[8] as our microcontroller, which will be powered by a Lithium Ion Battery (roughly 1000mAh) through a boost converter. This enables us to provide enough power to the microcontroller for it to send current to the different Power MUX as well as to power the display. The battery would be used to power the different ICs we have on our circuit.

We have an AdaFruit 7-segment display[9] that has four 7-segment units on it allowing us to display the battery level of the battery along with the speed of the hand crank to determine if the user is cranking at an optimal rate or not. This works with the w/I2C Backpack and thus, we are able to use significantly fewer pins of our microcontroller, keeping them for other use cases.

The 4-digit 7-segment display would be split into the 2 left digits for indicating Crank Speed, representing an F if we should crank "Faster" or an S (or 5) if we need to crank "Slower", and the right-hand side would represent the battery level. This would be detected using battery voltage on Vin, along with performing some basic calculations. This would prove to be useful as we want to ensure that the battery level does not drop low enough to turn off the microcontroller, and so we will always put the microcontroller to sleep when inactive while having a threshold of roughly 15% to ensure the battery is never completely depleted.

In addition, we will have a USB Charging Input to allow our internal battery to be charged in case the user wishes to keep the device charged for future use. This would hence allow the device to be used as a power bank, rather than just a hand crank.
To program the ATMega 328P we will be using an arduino which we can connect to our computer and write our software for the ATMega 328P. This allows us to easily program the Microcontroller and debug our code without risking shorting any of our parts.

Interfaces

Most of our devices will be centrally linked to the microcontroller, via 5V Lines except for the battery which has an input and output of 3.7V, so we will use a boost converter on its output to get it to 5V.
The boost converter we are using is the Adafruit Powerboost 1000C[10] which contains a built-in load sharing battery charger circuit that can convert a 3.7V Lithium Ion Battery input to 5.2V Output. This 5V Output would be more than sufficient to run the microcontroller and its various pin outs without any instability.
The display will be connected via the Analog Pin out (A4) of the Microcontroller and through the I2C Backpack mentioned earlier. This would allow us to transmit information without using most of the other pin outs.
The microcontroller sends enable signals to the 2 Power MUXs and the DEMUX as part of the Hand-Crank Subsystem. This is to ensure the current flow is directed to the right place in the particular circumstances.
The microcontroller will also get the voltage of the motor (from the Hand-Crank Subsystem) to determine if the hand cranking speed is optimal or not.
Additionally it will get input from the USB Device Out, to determine when a device is plugged in This is to help the DEMUX decide if the current should flow to the battery or the USB output.

Requirements and Verifications

Requirements	***Verification***
The battery can supply 3.7V to the Boost converter	Tested through multimeter
The Microcontroller outputs the correct signals to the Power MUX and DeMUX	The pin outs and PCB are configured correctly to ensure the right signal controls the right Power MUX
The software on the Microcontroller correctly identifies when a USB device is plugged in	Using Print Statements within Boolean conditions to test if it goes in the right condition
The software relays the information correctly to the Power MUXs	The pinouts are tested at various conditions
The Voltage of the Motor output is correctly measured	The pin we are using to detect the Voltage of the Motor gets the right value
The Battery Voltage is correctly measured	The Vin pin is also used to measure battery percentage and does it successfully
The microcontroller goes to sleep after period of inactivity	Observing states of Microcontroller after 5 minutes of inactivity
The battery does not get completely drained	The software mentions a minimum threshold (equivalent to 15% battery level) under which battery stops supplying
The battery is prioritized when low level	The software sets the Power MUXs to direct current flow to battery under 5%
The display correctly indicates the device mode	The two colon LEDs in the Display are on when the device is on.
The display indicates the correct battery level	The LED + Backpack are set up correctly and the battery level data is sent to the right LEDs.
The display correctly outputs the recommended crank speed modification	The LED + Backpack are set up correctly and the current voltage is compared to the ideal, and accordingly “F” or “S” is sent to the right LEDs.
The battery stops charging once 100% has been reached	The Power MUX stops current flow to the battery once the voltage of the battery is at 100%

Prioritize the hand crank even when the USB In is connected	The software ensures priority to Hand Crank
The battery circuitry is completely fine (essential for safety reasons)	We will first not use a battery and test the current flow into and out of the battery using a 3.7VDC power supply to see if the correct switching is happening.
The boost converter boosts the batteries 3.7VDC to 5VDC	We will measure voltages at all points using a multimeter to see if the desired boost is achieved

Subsystem 2

Hand-Crank Subsystem

The subsystem consists of the motor, linear regulator, the circuitry to direct the current flow one of two ways (DEMUX) and also the circuitry to select between one of two inputs (MUX). The motor is connected to a hand crank which is cranked by the user to generate power.
To account for energy losses, we will calculate the required motor voltage, by first looking at what is required at the device output. A standard USB A 2.0 can handle 5VDC ± 0.5V, so we want the output of the power MUX to be able to deliver a stable 5VDC. The power MUX will determine if the power comes from the battery or the hand crank. However, because we also use the hand crank to charge the battery, we use a DEMUX where the current flow from the hand crank splits into the battery or the MUX before the USB output.
Accounting for impedances, we can see that we need a stable 5V at all times. For this purpose, we use a 6V linear regulator. The linear regulator can output a stable 6V as long as it has an input of over 8V.

Our motor is a DC Motor that can easily provide 8-20V with reasonable crank speeds. To break it down, the Energy Input and Output at each stage would be as follows.

As mentioned before, we have a fairly complex circuit design due to the power MUXs and the DEMUX as we wish to automate our entire hand cranked charger. This is because we want to keep it as simple for the user as possible. To achieve this we have 2 power MUXs and one DEMUX. These are all controlled by the Microcontroller and work as follows:

- ☐ MUX 1: This is a power multiplexer that is present before the USB out to determine if the current flow comes from the hand crank or from the battery. This works by the microcontroller detecting if a USB is present and if the hand crank is not being turned (detected through V_motor) then the Power MUX will direct flow from the Battery to the USB out. Meanwhile, if the hand crank is being turned, it will prioritize the USB Out getting power from it instead (preserving battery level and health)
- ☐ MUX 2: This is a power multiplexer that comes before the battery to determine if the battery should be charged by the hand crank or by the USB Charging Input. This works by the microcontroller by default choosing the USB Charging Input as the default path to the battery and if the hand crank is being turned (via V_motor) it would select the Hand Crank to send power to the Battery. This is because if the Hand Crank is being turned we would like to use its power rather than the USB Charging in.
- ☐ DEMUX: This is a power demultiplexer that can direct the hand-crank power to either MUX1 or MUX2. This is essentially deciding if the end goal of the hand cranked energy is the battery or the USB output. This again works by the microcontroller sending a signal and using the battery level along with the presence of a device to determine where to send it.

Interfaces

The subsystem has various steps of voltage conversion as we start from a 10V Output from the motor which goes down to 7V after the linear regulator and eventually is a 5V Output from the USB. Throughout the subsystem, there are power MUXs and a DEMUX that directs the current flow and these are controlled by the microcontroller/ battery subsystem. These all run on 5V Lines and we can see that the microcontroller through its pinouts will signal the MUXs and DEMUX where the current should flow.
The overall MUX and DEMUX logic is shown below, this entire thing would be programmed on our Microcontroller and allows us to completely automate the procedure.

The following is the current flow at MUX 1:

Hand Crank Status	Battery Status	Outcome
Hand Crank not Turned	Battery Level Low	USB Out gets no current
Hand Crank not Turned	Battery Level Normal	USB Out gets current from Battery
Hand Crank Turned	Battery level low	USB Out gets Current from Hand Crank
Hand Crank Turned	Battery Level Normal	USB Out gets Current from Hand Crank

The following is the current flow at MUX 2:

Hand Crank Status	USB In Status	Outcome
Hand Crank not Turned	USB In not Supplying	Battery Gets no Current
Hand Crank not Turned	USB In Supplying	Battery charged by USB In
Hand Crank Turned	USB In not Supplying	Battery Charged by Hand Crank
Hand Crank Turned	USB In Supplying	Battery Charged by Hand Crank

The following is the current flow at DEMUX:

Battery Status	USB Out Status	Outcome
Battery Level Normal	USB Out Not Plugged	Current goes to MUX 2
Battery Level Normal	USB Out Plugged In	Current goes to MUX 1
Batter Level High	USB Out Not Plugged	Current goes to MUX 1
Battery Level High	USB Out Plugged In	Current Goes to MUX 1

Requirements and Verifications

Requirement	Verification
We must get 10V ± 2V from the Hand Crank Motor at typical hand cranking speed	We will use a Digital Multimeter to monitor the output of the motor for a range of RPMs
We must get a stable 5V Output after the Linear Regulator (±10% to account for noise and tolerances of both the regulator and the measurement device)	We will connect the multimeter to observe the output
Our DEMUX 1 can successfully route the crank power as per the signal from the microcontroller	We test the DEMUX without any input from the microcontroller (default case) and then pass in a high signal from a power supply to see if it switches to the correct current flow required using a multimeter
Our MUX1 can effectively choose between Hand Crank and Battery Power as per signal from the microcontroller	We test the MUX without any input from microcontroller (as the default case) and then pass in a high signal from the power supply to see if it switches to the correct path required using a multimeter

Our MUX1 can effectively choose between Hand Crank and USB Charge In as per signal from the microcontroller	We test the MUX without any input from microcontroller (as the default case) and then pass in a high signal from the power supply to see if it switches to the correct path required using a multimeter
The MUXs and DEMUX receive select signals correctly from the Microcontroller	The software enables and disables the MUX and DEMUX correctly and the correct pin outs are assigned to the right pin on the POWER MUX and DEMUX
The device automatically switches to device out when a device is plugged in	The current flow is observed using a multimeter and also through software print statements when the device is plugged in.
We test out overall circuitry without using actual batteries	We will replace the crank input with a power supply that can supply 10VDC and replace the battery with a 5VDC. We simulate plugging in a device by adding a resistor to the end of MUX1. In this scenario, if the 10V supply is off, the microcontroller is able to select the 5V supply. If the 10V supply is on, the microcontroller is able to select the 10V supply.

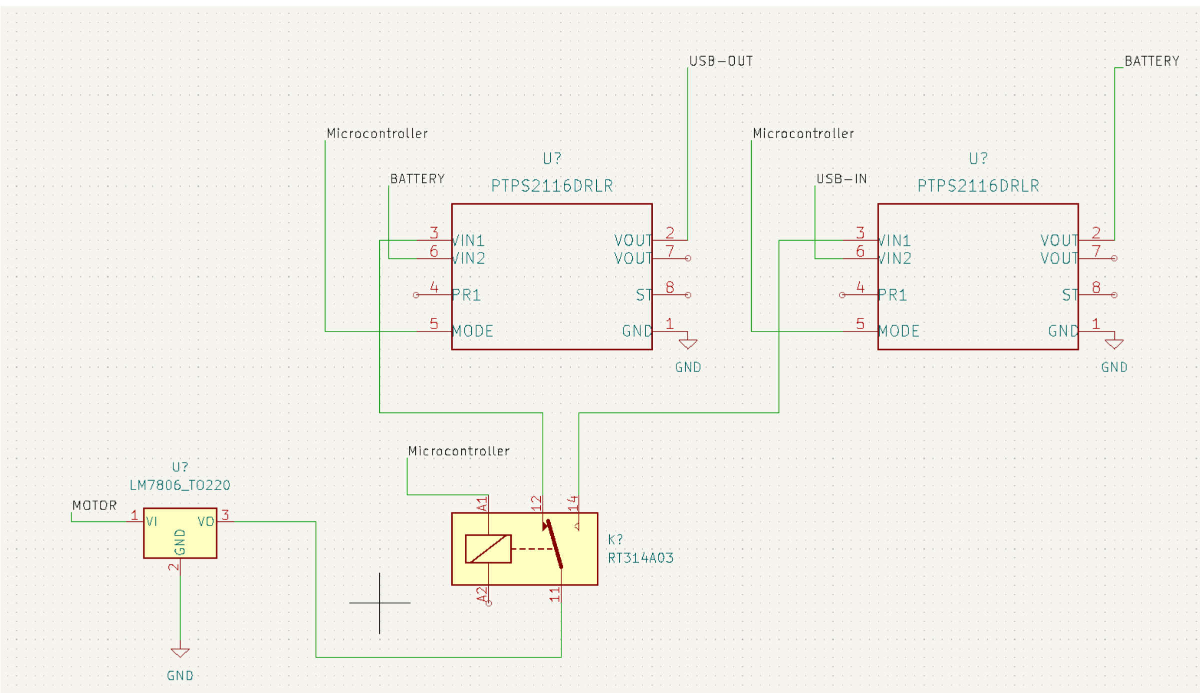

Figure 7. Circuit Schematic for the Hand-Crank Subsystem

Tolerance Analysis

Tolerance of Hand-Crank Subsystem

The hand-crank motor *Pittman Brushed DC 12VDO* is our input for the hand-crank subsystem. As the power is generated using human input, we can have a large variation in input voltage based on how fast the user can crank the motor. From our testing, the motor can generate 12V with no load at around 60 rpm. We can safely assume that a human can generate between 10-20V while cranking the motor. If the motor isn't cranked between these ranges, the regulator isn't powered and no current flows through the system.

Our linear voltage regulator *LM7806* has an input voltage range of 8V-20V, and gives us an output voltage of 6V ± 5% . Our DEMUX is a J103 single-pole dual-throw relay, with a coil voltage of 3V, but since the part is rated for a maximum coil voltage of 3.9V, it can be powered using our microcontroller. The contacts have a negligible resistance of 50 mΩ, so the voltage drop across the relay is negligible.

Since our power MUX *TPS2116* only has an input voltage range of 1.6 - 5.5 V , we must drop the voltage to 5V. To accomplish this, we can use a 1N4001 diode in forward bias. The diode has a voltage drop of 0.93V - 1.1V. There is negligible voltage drop across the power mux, with a typical on-resistance of 40 mΩ. With these tolerances in mind, we end up getting the following values:

Part	Minimum	Typical	Maximum
Motor Output (Hand-Cranked)	8V (for operation)	12V	20V
LM7806 Output	5.7	6	6.3
1N4001 Diode Output	4.77	5	5.2

Since the output of the 1N4001 Diode connects to the two MUXes, we end up with a typical output of 5V at the USB Output, with minimum and maximum values of 4.77 and 5.2 V. This is well within the specifications of USB, which states that the power pin must be between 4.5 - 5.5 V.

Tolerance of Battery Subsystem

The battery we use is the ASR00012 1000mAh Li-Po battery. The battery has a voltage of 4.2V when fully charged and a voltage of 3V when completely depleted. The battery handles over-charging and over-depletion protection, but our design prevents the battery from going below 3.3V to maintain constant power to the microcontroller.

The TPS61090 Boost Converter[11] found in the Powerboost 1000C can handle inputs from 1.8V to 5.5V, and it outputs a stable 5.2V, which is once again within the limits of the USB specification. By supplying the microcontroller with 5V as well, we reduce any risk of instability that comes with a low Vin to the microcontroller and all of our pinouts will work as per specifications. This further ensures that the signals to our Power MUX and DEMUX are supplied correctly.

Cost and Schedule

Cost Analysis

The total cost of our project is a cumulation of the labor cost of team members, the labor cost of the machine shop, and all the parts we need. The following table represents all the parts we need for this project and their respective quantity and price.

Description	**Part #**	**Manufacturer**	**Quantity**	**Total Price in USD**
Microcontroller	ATMEGA328P-PU	Microchip	1	$3.11
Boost Converter	Powerboost 1000C	Adafruit	1	$19.95
Display and I2C Backpack	ADA1002 w/I2C Backpack	Adafruit	1	$10.95
Linear Regulator	LM7806	ROHM	1	$0.50
Relay (DEMUX)	J1031C3VDC.15S	CIT Relay	1	$1.34
Power MUX	TPS2116	Texas Instruments	2	$1.82
Arduino (Programming)	Arduino Uno Rev3	Arduino	1	$27.60
Rectifier	1N4001	onsemi	3	$0.90
DC Motor	GM9213C177-R1	Pittman	1	$78.60
Lithium Ion Battery	ASR00012	TinyCircuits	1	$9.95
USB A Female	KUSB-6-3-4-3-6-1-10	KinnexA	1	$0.74
USB C Female	UJC-VP-G-SMT-TR	CUI Devices	1	$0.82
1kΩ Resistor	RC0201FR-071KL	Yageo	10	$0.37
2kΩ Resistor	RT0402BRD072KL	Yageo	10	$3.36
1µF Capacitor	M39014/02-1415V	Kyocera AVX	1	$5.85
2.2µH Inductor	ADL2012-2R2M-T01	TDK Corporation	10	$5.37

The total cost of parts comes to $171.23 above which we can estimate roughly 10% for tax, and 7% of the total cost for shipping. The final cost of parts with tax and shipping comes to $200.33, furthermore, there are 3 members who would also obtain a salary. The standard salary we expect is $40/hour and we intend to work on the project for roughly 100 hours each, therefore, the total salary per person is $10,000 and the overall labor cost for the team would be $30,000.

$40/hour * 2.5 * 100 hours = $10,000 per person

The labor cost of roughly 25 hours of the machine shop at an hourly rate of $50/hour would be $1250/-

$50/hour * 25 hours = $1,250/-

*This brings the total cost of labor and parts to **<u>$31,450.33</u>**.*

Schedule

<table>
<tr><th>Week</th><th>Tasks</th><th>Members</th></tr>
<tr><td rowspan="4">Feb 20th - Feb 26th</td><td>Finalize the components we will be using</td><td>Shreyasi / Rubhav</td></tr>
<tr><td>Write up Design Document</td><td>Shreyasi / Achyut</td></tr>
<tr><td>Start PCB Design</td><td>Rubhav / Achyut</td></tr>
<tr><td>Write up Team Contract</td><td>Team Work</td></tr>
<tr><td rowspan="4">Feb 27th - Mar 5th</td><td>Finalize Design as per review</td><td>Shreyasi</td></tr>
<tr><td>Complete PCB Design</td><td>Rubhav / Achyut</td></tr>
<tr><td>Test through CAD Softwares</td><td>Rubhav</td></tr>
<tr><td>Order Parts</td><td>Shreyasi</td></tr>
<tr><td rowspan="2">Mar 6th - Mar 12th</td><td>Work on Teamwork Evaluation</td><td>Team (separately)</td></tr>
<tr><td>Finalize Machine shop designs</td><td>Achyut</td></tr>
<tr><td rowspan="2">Mar 13th - Mar 21st</td><td>SPRING BREAK</td><td>Team</td></tr>
<tr><td>Begin Soldering (if parts and PCB Arrive)</td><td>Shreyasi / Rubhav</td></tr>
<tr><td rowspan="2">Mar 20th - Mar 26th</td><td>Perform unit testing to ensure the functionality of each part and subsystem</td><td>Achyut</td></tr>
<tr><td>See if our PCB is completely working (before next round of PCB orders)</td><td>Rubhav</td></tr>
</table>

Mar 27th - Apr 2nd	Complete Individual Progress Report	Team (separately)
	See if hand crank output works	Achyut
	Ensure Relay set up is working	Rubhav
	Program Microcontroller	Shreyasi
Apr 3rd - Apr 9th	Evaluate Progress and attempt complete the entire circuitry	Team
	Make sure any last minute changes should be made	Team
Apr 10th - Apr 16th	Prepare for Demonstration of project	Team
	Complete Team Contract	Team (separately)
Apr 17th - Apr 23rd	Do thorough testing before Mock Demo	Team
	Start writing up the final presentation and paper	Team
Apr 24th -Apr 30th	Finish up respective parts of presentation	Team
	Do Final Demos	Team

Ethics and Safety

This project deals with safe regulations on various parts of the design, to ensure no concerns on the end of the user. Since our product is contained in a case of its own with just a simple design for a hand-crank, the user will not find it complicated to operate. Since the system is automated, the user just needs to plug in the device they wish to charge, and on the input side either rotate the hand crank or plug in a power source to the type C port.

With every element of this product, we have kept in mind the highest standards of caution and safety to abide by the IEEE Code of Ethics Section 1.1[2], while disclosing any feature or factor that might cause any type of danger to the community of users or their surroundings.
We have successfully eliminated the risk of shock or electrocution by replacing the casing of the product with a plastic one, instead of one made with any kind of metal. It is recommended to use such products in dry conditions, without water or moisture in direct contact, for general safety concerns. The product does not generate enough to cause the user any type of serious injury, and they need not be concerned about conduction or potential static discharge dangers. Along with these claims, we would also like to mention that this does not mean all types of injury and danger can be prevented and hence, the user cannot risk being careless. We would like to bring the user's attention to Section 1.5 of the IEEE Code of Ethics[2], and warn users to handle the product with caution for their own personal safety, since we can only make claims and assumptions based on available data.

The most important hazard to accentuate is the presence of lithium-ion batteries in this product. These have warnings printed on their labels cautioning against charging, excessive heat and other external conditions[1]. Not guarding the device against mechanical abuse (deformation, crushing, damage to the outer casing) could result in an unfavorable reaction.
Another major point of damage is overcharging of the battery through prolonged periods of time which could cause damage to user safety and their surroundings[4]. However, the team ensures to tackle this problem, since our microcontroller ensures that the battery does not get overcharged, cutting off supply to the battery before the limits of the battery are reached. Most accidents surrounding Li-ion batteries[3] are found to be caused by mishandling of the device or unintended abuse of the battery in some way. The presence of the battery also makes it mandatory to design the circuit with utmost care, so that we are able to prevent short circuits or any fires. The team will not be handling the battery for most of the testing phase to avoid any battery-related accidents, to maintain lab safety. We will eventually experiment with how well the electromechanical system works with the battery, but with adequate training and safety measures.

We suggest not opening the casing of the product unless with proper training or regulation, and to keep checking upon any damage incurred by the product. It is sensible to keep flammable or combustible materials away from the working space of the product, while also not overheating the product or keeping it in unfavorable temperatures.

Referring to Section 1.6 in the IEEE Code of Ethics[2], we also ensure working on this project with adequate training and safety regulations, all details of which are made transparent and available by the team members. We also guarantee to provide ample information about our product and make users knowledgeable about the design and potential risks to safety.

Citations

[1] University of Illinois at Urbana Champaign, "Safe Practice for Lead Acid and Lithium Batteries," April.13, 2016
[Online]
Available:https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf
[Accessed Feb. 19, 2023]

[2] Institute of Electrical and Electronics Engineers, "IEEE Code of Ethics," June 2020
[Online]
Available: https://www.ieee.org/about/corporate/governance/p7-8.html
[Accessed Feb. 20, 2023]

[3] Electrochem Solutions, Raynham, Massachusetts, United States, *Primary Lithium Battery Safety and Handling Guidelines,* Revised 2017
[Online]
Available:https://s24.q4cdn.com/142631039/files/doc_presentations/Resources/Battery-Safety-and-Handling-Guide.pdf
[Accessed Feb. 19, 2023]

[4] Massachusetts Institute of Technology, "Lithium Ion Battery Safety Guidance," April.13, 2016
[Online]
Available:https://ehs.mit.edu/wp-content/uploads/2019/09/Lithium_Battery_Safety_Guidance.pdf
[Accessed Feb. 19, 2023]

[5] ROHM Semiconductor, "1A Output 78 series Regulators" #12019ECT01 Datasheet, [Revised Mar. 2012]
Available:https://fscdn.rohm.com/en/products/databook/datasheet/ic/power/linear_regulator/ba78_series-e.pdf
[Accessed Feb 19, 2023]

[6] CIT Relay and Switch, “Relay Catalog” #E197851 Datasheet,
[Revised Mar. 2012]
Available:https://www.citrelay.com/Catalog%20Pages/RelayCatalog/J103.pdf
[Accessed Feb 19, 2023]

[7] Texas Instruments, “Low IQ Power MUX with Manual and Priority Switchover” #SLVSFG1A Datasheet,
[Revised May. 2021]
Available:https://www.ti.com/lit/ds/symlink/tps2116.pdf
[Accessed Feb 19, 2023]

[8] Atmel, “8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash” #7810D-AVR Datasheet,
[Revised Apr. 2000]
Available:https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
[Accessed Jan 15, 2015]

[9] Melissa LeBlanc-Williams, “Adafruit LED Backpacks,” Revised Jan.13, 2023
[Online]
Available: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-led-backpack.pdf
[Accessed Feb. 19, 2023]

[10] Adafruit Industries, “Adafruit Powerboost 1000C,” Revised Jan.12, 2022
[Online]
Available:https://cdn-learn.adafruit.com/downloads/pdf/adafruit-powerboost-1000c-load-share-usb-charge-boost.pdf
[Accessed Feb. 19, 2023]

[11] Texas Instruments, “Synchronous Boost Converter with 2A Switch” #SLVS484A Datasheet,
[Revised April. 2004]
Available:https://cdn-shop.adafruit.com/datasheets/tps61090.pdf
[Accessed Feb 20, 2023]