EE175AB Final Report

Adaptive Battery Charging Unit

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

Project Team Member(s)	Timothy Perrier, Tae Hyun Kim, Siwon Kim, Brandon Lam
Date Submitted	March 15, 2020
Section Professor	Dr. Roman Chomko
Revision	Revision 2.0
URL of Project Youtube/Webpage	Youtube Link for working project Link to control charging profile
Permanent Emails of all team members	Timothy Perrier: tperr003@ucr.edu, Siwon Kim: skim258@ucr.edu, Tae Hyun Kim: tkim058@ucr.edu, Brandon Lam: blam020@ucr.edu

Summary

This report presents technical specifications on a Battery charging unit. The unit charges the battery constant current and constant voltage method adequate for battery profile.

1

Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
0.5	First edits: Executive summary and section 2.1	Timothy Perrier	02/24/2020	
1.0	Completed first draft	Timothy Perrier, Tae Kim	03/14/2020	
2.0	Final Draft	Timothy Perrier, Tae Kim, Siwon Kim	03/15/2020	
3.0	ENGR 181W Final Edits	Tae Kim, Timothy Perrier	06/12/2020	

Table of Contents

1 Executive Summary

The Adaptive Battery Charging Unit is a battery charger which safely and efficiently charges Lithium-Ion batteries (LiB). This charger uses a BJT Active Constant Current Source for the Constant Current (CC) charging phase and a Buck Converter for the Constant Voltage (CV) charging phase. The charge time is approximately 3 hours, and the temperature of batteries charged does not exceed 25 ℃.

Most of the available chargers do not charge LiBs correctly, causing degradation of internal battery components. These chargers are tailored to charge fast battery charging. However, charging LiBs quickly by applying large currents heats up the battery which results in overheating. Overheating during both charging and discharging is directly correlated with a decrease in cycle life of LiBs. Charging LiBs with a large current for too long risks overcharging. Overcharging LiBs will also accelerate the internal degradation of materials, but more importantly the battery risks venting and, in some cases, exploding.

Decreasing the cycle life of LiBs results in more battery waste which is hazardous to the environment. Therefore, the motivation for this project is to minimize the amount of battery waste that individuals produce by maximizing the cycle life of LiBs. The purpose of this project is to design a charging unit for ICR 18650 LiBs that efficiently and safely charges them to maximize the cycle life of LiBs.

This charger uses the CC method to charge the bulk of a LiB, followed by CV for the remaining 1%. An LCD provides information about the battery voltage and percentage. A website is used to select the type battery that the consumer wants to charge. This website changes the value of charging current depending on the selected battery. LiBs in this charger are safely charged in the standard time of 3-4 hours. The tests were all proven to be successful as the buck converter provided correct voltage, active current source provided correct current, and battery was able to be charged with the correct rate.

2 Introduction

2.1 Design Objectives and System Overview

This project is an Adaptive Battery Charging Unit for LiBs. A Buck Converter is used to step down and stabilize a DC voltage for the CV charging phase A BJT Active CC Source (BJT ACCS) was also designed to keep a stable current for the CC charging phase. This project is mainly designed to safely charge ICR 18650 LiBs at a standard rate. The purpose of safely charging LiBs is to maximize their life spans and thus reduce battery waste. Concepts from power engineering, solid state electronics, and embedded systems were used to design and develop this project.

The main components of this battery charger are a Buck Converter, BJT ACCS, ATMega 1284 microcontroller, and an Arduino. The Buck Converter is used for the CV charging phase. It steps down a large DC Voltage value to the appropriate 4.2 V required for the CV phase. The BJT ACCS delivers a large DC current to the battery for the CC phase. A digital potentiometer in the circuit, controlled through the website, varies the current output of this circuit depending on the battery specifications. The ATMega 1284 microcontroller reads and displays battery information onto a LCD. This microcontroller also controls the switching between CC and CV charging phases using a relay. The Arduino is set up to receive information from a website through USB communication and controls the digital potentiometer based on the information received.

The battery charger circuitry should be encased in a case, and a LCD, LED indicator, and battery holder are all visible from the outside. However, we were not able to encase our circuitry due to lack of time. This product charges ICR 18650 LiBs individually in the span of 3-4 hours [7]. A LED indicator displays the State of Charge (SoC) of the battery using the LM3194 [13], an LCD screen displays information about charging current and battery voltage, and two different LEDs turn on depending on the present charging phase. The operating environment is between 0 and 45 °C [7] to prevent damage to batteries. An ICR 18650 LiB needs to be placed into the battery holder followed by flipping the On/Off switch to the On position to begin the charging process.

The LM2678 simple switcher for the buck converter operates at a 260 kHz switching frequency [11]. The LCD displays battery voltage and charging current within an error margin of ± 5% due to error from the microcontroller hardware. LED indicator has an accuracy within 10% of the actual voltage value. This higher error margin is due to the nature of the LM3914 analog chip used. Serial communication between the website and Arduino has a 3 ms delay. The buck converter provides 4.2 ± 0.03 V and the constant current circuit provides 1.3 A which are within the battery charging requirements. The ATMega 1284 operates at 125 kHz with ADC operating at 960 Hz and samples battery voltages every 15 seconds [3].

This project serves its purpose to maintain battery life longer than other chargers that do not follow the proper charging method. With a battery that stays alive longer than intended, toxic waste is eliminated which helps eliminate economic and environmental problems. Also since battery waste will decrease from the charger, consumers will more likely buy electronics with lithium ion batteries improving the market as well.

The project is intended to charge any type of Lithium Ion batteries used in any electronics. The charger can also adapt to the capacity/mAh to match the amount of amps needed to charge the battery. Since the currents can match the battery exactly, the batteries can stay alive longer as they are not overheated or overcharged in the process of charging. The project is related to Electrical Engineering because it uses power engineering and circuitry to make a battery charger. The goal overall is to properly charge a lithium ion battery with a proper charging method to maintain battery life.

Responsibilities:

- Timothy Perrier: LED Indicator, LCD Screen, CC circuit, Circuit Schematics, Relay Circuit, Systems Engineering documents, Charging Current Control using Localhost, Arduino, Battery SoC calculations
- Siwon Kim: Buck Converter, Digital Potentiometer
- Tae Hyun Kim: Systems Engineering documents, LED Indicator, LCD Screen, Battery SoC calculations, assistance in help test/build CC and CV circuits
- Brandon Lam: Assistance with charging current control using localhost, Assistance with Arduino

2.2 Backgrounds and Prior Art

Existing LiB chargers are quite small and can charge multiple batteries at the same time; however, many chargers are very basic and do not display battery characteristics such as percentage and voltage. Chargers on the market do not have any website or application connectivity, and therefore cannot adaptively change their charging current value. This means that these chargers are only able to safely charge batteries of a single mAh rating. Some chargers are able to charge multiple types of ICR LiBs, AAs, AAAs, Ni-CD, and many more. These chargers have adaptive-length battery holders which use a spring and rail mechanism to either shorten or lengthen the compartment depending on battery length. Less expensive chargers sacrifice many features including displays for battery information [5].

For the design we did not have enough time to build an encasement for the electronics, nor did we have time to put everything on a PCB. Designing a charger that would be able to charge multiple batteries at once would have also needed much more time. Although these design specifications were not able to be achieved, the LCD screen which displays battery voltage and percentage is an advantage in comparison to most battery chargers on the market. Having the website to control the charging current is also a significant merit over any charger on the market.

2.3 Development Environment and Tools

Software development was done in multiple programs. Atmel Studio 7.0 was used for microcontroller programming of the ATMega 1284. The Arduino programming software was used for programming the Arduino, and Sublime Text 3 was used to program the website along with serial communication between the Arduino and the website.

Hardware development required the use of the BK Precision 1760 power supply, BK Precision 5492B multimeter, BK Precision 2190E oscilloscope, and Keysight 33210A signal generator. The latter two were only used for designing the buck converter.

2.4 Related Documents and Supporting Materials

Industry Standard Materials:

Great Power, “ICR18650 2600mAh Datasheet”, ICR18650 2600mAh Datasheet, Jul. 2012

Texas Instruments “LM2678 Simple switcher High efficiency 5-A Step Down Voltage Regulator”

SNVS029K data sheet, March 2000[Revised February 2017]

Texas Instruments “LM3914 Dot/Bar Display Driver” SNVS761B datasheet, January 2000[Revised March 2003]

2.5 Definitions and Acronyms

ACCS – Active Constant Current Source

ADC – Analog to Digital Conversion

BJT – Bipolar Junction Transistor

CC – Constant Current

CV – Constant Voltage

Digipot – Digital Potentiometer

Fosc – Oscillating Frequency

LCD – Liquid Crystal Display

LED – Light Emitting Diode

LiB – Lithium Ion Battery

mAh – MilliAmp-Hours

MCU – Microcontroller

MOSI – Master Out Slave In

PWM – Pulse Width Modulation

SCK – Serial Clock

SoC – State of Charge

3 Design Considerations

3.1 Realistic Constraints

This design had two major constraints: high power and microcontroller architecture. Charging LiBs requires relatively high current outputs, usually between one and two Amperes. Therefore, components in the CC circuit had to be chosen for their high absolute maximum power ratings. The microcontroller used for reading and calculating battery information limited the accuracy due to its somewhat slow operating frequency. Although the ATMega 1284 read battery voltage with relatively high accuracy, a microcontroller with a faster operating frequency would have yielded much more accurate battery readings.

Power constraints:
Due to high power consumption, many high-power circuit elements were required for this design. The TIP-31C BJT [6] was required for the CC circuit because it has a maximum collector current rating of 3 A, collector-base/emitter voltage ratings of 100 V, and a maximum junction temperature of 150℃. Since the CC charging phase outputs extremely high power, this BJT was one of the only options available to use. Heat sinks were also required on the TIP-31C to help it cool down for improved safety. Resistors with a 10 W power rating were also required because of 5 W power dissipation across it during CC charging. High power relays were also required to support high currents and voltages.

Microcontroller constraints:

The ATMega operating frequency of 125 kHz also limited the frequency at which it could perform the ADC process. Since the team uses a prescaler of 128 in the software, the ADC operates at 0.976 kHz (125 kHz ÷ 128) [15]. Serial communication between the localhost website and Arduino has a 3 ms delay; however, this is not significant as the charging current only needs to be selected once per charge at the beginning [5].

3.2 System Environment and External Interfaces

The charging current is controlled with a digital potentiometer using an Arduino. The Arduino receives information from a localhost website using serial communication. The Arduino then uses SPI to communicate with the digital potentiometer to change its resistance value.

The ATMega 1284 uses analog to digital conversion (ADC) to read battery voltage. The battery's voltage was then converted to percentage using calculations based on the 18650 battery discharge curve. Battery voltage and percentage are displayed on the LCD. The annotated discharge curve can be found in

Appendix F. This microcontroller also switches between charging phases by controlling a relay. When the battery voltage read using ADC is above 4.18 V, the relay switches the power output from CC to CV.

3.3 Industry Standards

Specific voltage and current values are required for charging the ICR 18650 LiBs which are given in battery datasheets. Industry standards require that the primary charging phase (CC) requires a CC of 1.3 A, and the CV phase requires a stable voltage of 4.2 ± 0.03 V. In order to achieve a charging current of 1.3 A, a high input voltage of 11 V needed to be used. As a result, a buck converter was needed to step down input voltage for the CV charging phase so that a single power supply could be used. The charging environment's temperature must be between 0-45℃, but this never became an issue in the design that had to be worked around. These high-power standards required the use of a variety of high-power circuit elements [7].

SPI communication is used to program MCP4131 (Digipot). The digipot resistance is mapped 70 Ωto 9.6 kΩwith bits 0 to 125. Fosc is 125 kHz and SCK is set to a quarter of Fosc by writing zero to registers SPR1 and SPR0 as described in ATMega1284 datasheet [3]. The SCK pin on the ATMega 1284 gives 8 pulses when the CS pin on the digipot is enabled. The MOSI pin on the ATMega 1284 writes mapped digipot values for each cycle for the communication to be successful.

3.4 Knowledge and Skills

- Courses taken related to this project: EE 100A/B: Electronic Circuits, EE 120B: Intro to Embedded Systems, EE 133: Solid-State Electronics
- Prior Knowledge: Circuiting, coding for Atmel, and other various course materials that is too long to list

Timothy Perrier:

- New Knowledge and Skills Learned:
    - Power electronics (buck converters)
    - Battery charge/discharge curves
    - CC, CV charging methods
    - SoC of LiB based on voltage measurements
    - Documentation
    - Systems Engineering

Tae Kim:

- New Knowledge and Skills Learned

a. Power electronics such as Buck Converter, Active Current Source, Buck Boost
b. Battery properties and charging method
c. Lithium Ion battery charging curves
d. team environment, documentation, systems engineering
e. Professional and ethical work

Siwon Kim:

- New Knowledge and Skills Learned:

Brandon Lam:

- New Knowledge and Skills Learned

3.5 Budget and Cost Analysis

Our budget is around 160 dollars and we can reduce some because we use individual single parts from the total amount. Many duplicate parts were purchased for extensive testing. We calculate the budget to be this amount because we used the most optimal parts for our project and believe that it cannot go much lower than this.

Part	Seller	Amount	Price ($)
LM2678T-ADJ/NOPB	Texas Instruments	2	16.21
Resistor Kit (500)	Sparkfun	1	7.95
GL5E Relay	Sparkfun	2	3.9
18650 Li-Ion Battery	Sparkfun	7	41.65
Battery Holder	Sparkfun	2	2
IFRZ44N NMOS	Amazon	20	10
PMOS	Sparkfun	5	9.27
TIP-31C BJT	Amazon	2	6.2
Super Capacitor	Sparkfun	2	9.9
LM3914 LED Display Driver	Amazon	5	9.99
MCP4131 Digital Potentiometer	Amazon	2	11.5
Boost Converter	Amazon	5	9.99
Knob Potentiometer Pack	Amazon	1	6.99
10k Ohm Potentiometers	Amazon	10	5.99
Power Inductors	Amazon	20	9.99
Total			**161.53 $**

3.6 Safety

Overcharging batteries:

The main safety concern for this project is overcharging the batteries. Overcharging in the CC phase can cause irreversible damage to the internal battery components, decreasing the battery's life cycle. Overcharging can also cause battery venting. This is the process in which the battery heats up too much, causing a pressure build up, and eventually the release of toxic gas. To prevent this the main focus was to measure the battery level at all times so that there would be no overcharging.

Power levels that can destroy project:

A secondary safety concern involves the use of power levels above circuit element maximum ratings. Due to this project's nature of requiring high power in certain circuits, elements that can withstand these power levels must be chosen. The power output of this project is in the range of 5-10 W which is very high, and certain elements get extremely hot. Certain precautions had to be taken such as using heat sinks and isolating the circuit elements which got hot.

3.7 Performance, Security, Quality, Reliability, Aesthetics etc.

- Battery does not deliver power to source
  - Certain power is required to charge a battery. Charging the battery below specified power level to protect the battery from damage.
- Battery does not exceed set voltage
  - Each cell in the battery pack can handle up to a set amount of voltage. The team determined the battery voltage limit according to datsheet.
- Stable power to charge battery
  - We considered the design that the battery charger can efficiently deliver desired power to the battery. The team also considered leakage voltage/current to minimize power loss so we get expected power delivered to the battery.

3.8 Documentation

For the most part, maintaining documentation for the project was done in Systems Engineering. The first document that was worked on was the Statement of Work which helped with the overall project planning in the very early phase of the project. A system requirements document was created to list and describe the core requirements that the project needed to fulfill by the end. A Gantt Chart was created shortly afterwards to set a timeline and dates for specific objectives to be completed by. The Preliminary Design Review (PDR) took place much later on during the first quarter and helped organize and bring all previously completed documents together. During the second quarter, the Critical Design Review took place which was mainly an updated version of the PDR. This was followed by the completion of the Acceptance Test Procedure (ATP) document which outlined the plan and procedure for how each subsystem of the project would be tested. After completing the ATP, the verification matrix was filled out to show that testing was successful.

3.9 Risks and Volatile Areas

Some risks include the fact that our system can only take a certain amount of voltage as the buck is set to step down to 4.2 V from 11-12 V. An environment where the circuit takes more than 11-12 V, the charging current would be too high causing unsafe charging of the battery. This could potentially lead to the battery venting while it charges. Other problems that stem from higher voltages include burning circuit elements and the buck converter not being able to correctly step down voltage. To mitigate any risk of injury, we worked on the project with no other groups around and with safety glasses available.

4 Experiment Design and Feasibility Study

A proof of concept experiment was done to evaluate the feasibility of the buck converter. This proof of concept of the buck converter was tested using the simplest theoretical form of a buck converter. This experiment worked, so from there the LM2678 was purchased to design a buck converter more suited for industrial standards instead of a traditional schematic buck converter.

To evaluate the feasibility of the CC circuit, multiple simulations were completed in PSpice and proof of concept circuits were tested before ordering the TIP-31C BJT and building the final circuit.

The feasibility of PC connectivity was performed with blue-tooth and a second time through serial communication. The blue-tooth was tested using a HC-05 blue-tooth chip, the ATMega 1284, and a smartphone. After working on blue-tooth for a week, it was deemed infeasible and serial communication was the alternative. Wired connectivity was a big trade-off to wireless connectivity, but it was much easier to work with.

4.1 Experiment Design

Buck Converter: Siwon Kim was responsible for designing and testing the buck converter.

The circuit is expected to provide 4.2 V during CV phase. A set of experiments are done to observe the output. BK Precision 1760 A power supply is used to provide 11VDC input.

Digipot: Siwon Kim was responsible for designing and testing the Digipot.

The circuit is expected to provide different resistance levels depending on different profiles. We put two buttons to choose different profiles. ATmega1284 is used to communicate with MCP4131(digipot). Different state of profile is displayed on a LED register.

Timothy Perrier was responsible for designing and testing the CC circuit.

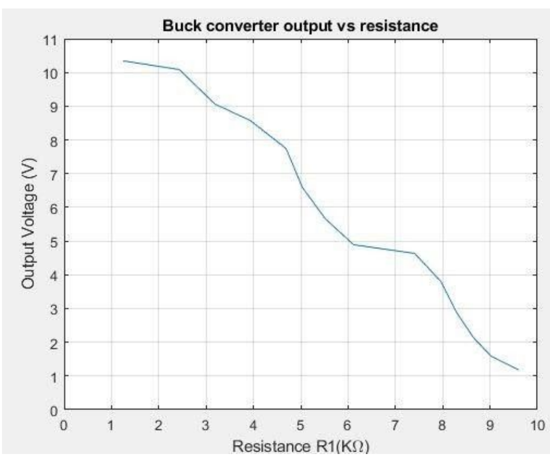
The design of this circuit is a BJT ACCS circuit. The objective of this circuit is to provide 1.3 A of current to the battery for the CC charging phase. A PSpice simulation was done to observe the effects of adding more loads to the output of an ACCS. After observing that there was no current drop at the output regardless of the load, a proof of concept test was designed. The proof of concept experiment was completed using a 2N222A BJT and LEDs at the output. Using this transistor, an ACCS was built to supply a constant current value regardless of the load at the output. Current was measured at the output using the BK Precision 5492B multimeter. Currents of under 30 mA were expected for this circuit because the 2N222A BJT has very low output current and voltage values.

Timothy Perrier was responsible for testing PC connectivity.

PC connectivity was attempted using an HC-05 blue-tooth module, ATMega 1284, and a smartphone. Using the UART library for AVR, a simple program was to be designed to turn on the HC-05 and connect with the smartphone. The HC-05 and ATMega 1284 would then be turned on to connect to the smartphone.

4.2 Experiment Results, Data Analysis and Feasibility

Siwon Kim was responsible for experimenting with Buck Converter. The buck converter is designed to have a voltage divider with a potentiometer to vary feedback voltage. The relationship between increasing and decreasing resistance versus output voltage is linear.



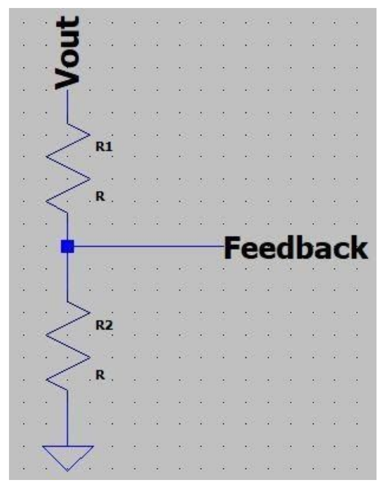


Figure 4.2.1 - Output voltage vs resistance (Left) and voltage divider (right, model for experiment)

Figure 4.2.1 (Left) shows linear relationship with varying voltage. As expected by the formula:

$V_{Feedback} = 1.25V = V_{out}(\frac{R2}{R1+R2})$, increasing R1 decreases output voltage to keep $V_{Feedback}$ constant [11].

Siwon Kim was responsible for experimenting with the digipot. The desired output values are hard coded into the digipot. Two buttons were used to increase and decrease resistance. The resistance varied as expected with the corresponding coded level of resistance.

Timothy Perrier was responsible for carrying out the experiment and testing of the CC circuit. The simulation of the circuit with the TIP31-C and a 4.2 V input can be seen in the figure below. The values on the circuit are when the R2 value was 10 kΩ. The output current at the collector and emitter terminals of the BJT show that this circuit was feasible. The only aspect that needed changing was the input voltage in order to achieve an output current of 1.3 A. The effect of variation of the R2 value on the output current can be observed below in Figure 4.2.2a.

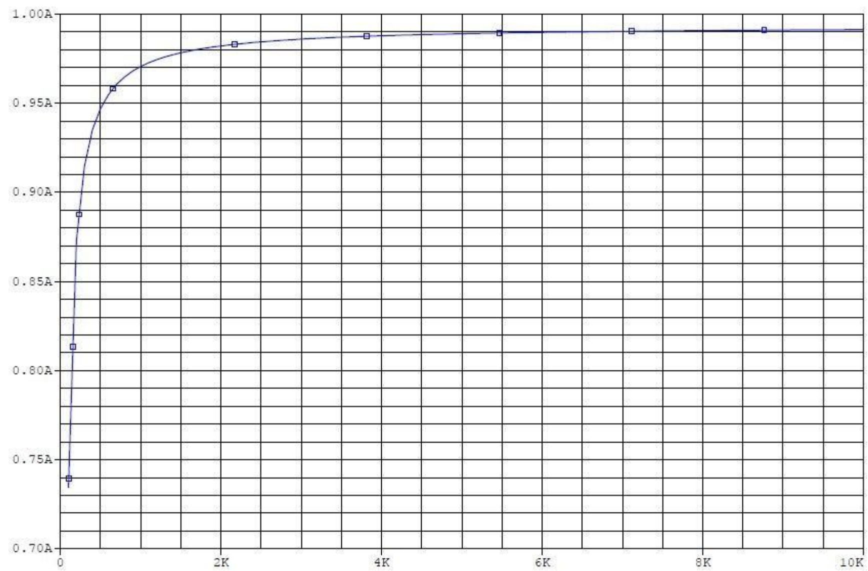


Figure 4.2.2b: Current-Resistance Relationship of ACCS Circuit

After simulating this circuit, it was built and output current was measured using the BK Precision 5492B multimeter. With an input voltage of 4.2 V, an output current of 0.92 A was achieved. This measurement is not exactly the value from the simulation; however, it is close enough due to natural error causes such as imperfect resistance and voltage values. Changing the potentiometer value also exponentially changed the output current in the same relationship that is shown in Figure 4.2.2b above.

Timothy Perrier was responsible for testing the PC connectivity using bluetooth. The smartphone was unable to connect to the HC-05 bluetooth module after many iterations of code and trials. Thus bluetooth was considered infeasible after a week of testing. Rather than using bluetooth, PC connectivity was completed using serial communication between a localhost website and an Arduino. Wireless communication was the preferred design objective, but serial communication through USB proved to be much easier to develop and use.

5 Architecture and High Level Design

5.1 System Architecture and Design

High Level overview of the system:

1. Battery is put into the charger
   As stated in this part of the project the battery is put into the charger ready to be charged before the charger is plugged in giving the user control of when they want to charge the battery instead of charge and plug which makes the project more user friendly.
2. The Charger supplies exact amount of power needed to charge at desired time
   In this part our calculated CC and CV values are correctly inputted into the battery giving the exact amount of power to be charged within the standard charging time of a lithium ion battery. (From datasheet)
3. The LCD screen and LED displays the state of charge
   The ADC conversion that is done through the microcontroller and the LED display chip allows the correct display. We have also made the LCD more accurate by sampling every couple of seconds of where the battery voltage is at. That is controlled through a relay as well like the switching between CC and CV except the relay here would stop supplying power to the battery while our microcontroller is sampling the voltage.
4. Website allows control of current
   The website coded by simple arduino allows the control of current being supplied to the battery by varying the value of digipot input data which changes the resistivity of the digipot allowing variance in current. It is very simple to use as there are buttons on the website for the type of mAh for certain lithium batteries and the buttons change the current input according to the mAh.

5.2 Hardware Architecture

In the “Battery Charging Core” module, Timothy Perrier was responsible for operation and design of the ACCS, relay, and data set from the Arduino. Timothy made sure all of the parts were working by doing specific tests to each part with Tae Hyun Kim assisting on the side. For the ACCS the testing was just to check if it was working properly. For the Relay both members set specific voltages that matched the data sheet for the Relay to see if it was functioning. Then the two members set voltages out of the range from the data sheet to make sure Relay was not functioning in that range. For the data set of Arduino, Timothy did the calculations and other data matching with the data sheet to make sure everything is functioning within the parameters such as making sure that the team does not put too much power to short or burn the microcontroller.

Siwon Kim was responsible for the circuit design and building of the buck converter. He was also responsible for testing the buck converter. He made sure the output/input values were in our desired range of 4.2 volts by having the duty cycle set correctly within our buck converter system. That was controlled by a potentiometer and Siwon Kim matched the data sheet to set the correct value for the potentiometer to allow the Buck Converter to give the teams' desired range of output.

In the "Main System" module, Timothy Perrier was responsible for the entirety of the module in building the whole project together and providing calculations for each test with assistance from Tae Hyun Kim. The two were responsible for making sure the battery was charging correctly with the correct output current/voltage. Made sure all the circuits were connected and no shortage was happening. Tested multiple voltages to make sure it gave the correct output no matter what the condition was.

Timothy Perrier was responsible for creating the "Website for Battery Type" module through simple arduino coding. He tested his website through buttons that he coded and made sure the buttons sent the correct value and signal to the main system.

The Battery Charging Core block is the core of the project as it states. It is where the power is delivered from to the battery and without it the project would not work. The core has an input voltage which then is sent to the relay that switches between CC and CV. The core outputs either the current or voltage depending on the battery voltage which is taken into account by the comparator. A Digipot is in the block just to vary the current for certain types of lithium ion batteries.

Website for Battery Type block is a subsystem where it's an added feature for customer convenience. The user is allowed to press a button on the website which sends the data to the Arduino that sends data to the digipot to vary resistance inside our active current circuit. That allows the current to be adjusted accordingly to the type of lithium ion battery.

LED Display block is also another subsystem that is an added feature. This block is very simple as the LM3914 IC would read the battery voltage and with the calculations done by Timothy Perrier and Tae Hyun Kim, the LED bus outputs the correct SoC for the battery.

Overall concept and SBD of the project is given below. All of the modules described above are mapped and are able to be easily followed. (Battery Charging Core, Website for Battery Type, and LED Display are all individual blocks that connect together in the Main System block.) The SBD was drawn and made by Tae Hyun Kim.

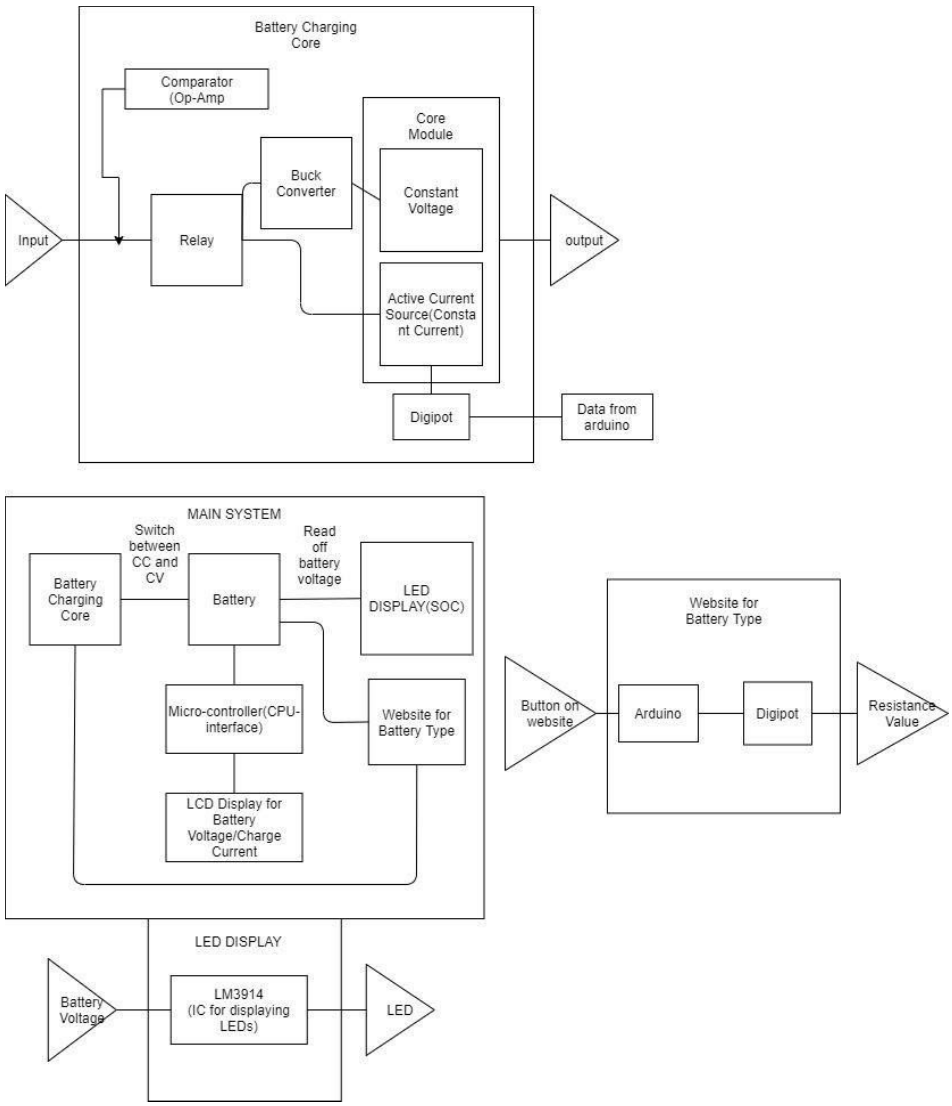


Figure 5.2.1: System Block Diagram

5.3 Software Architecture

Timothy Perrier was responsible for all the software architecture. The software architecture includes the website with arudino. The website provides an user interface where they are allowed to choose which type of battery they are charging depending on their capacity and mAh rating. Then the website communicates with our project by varying the Digipot input and output data which can vary the output current of the project giving the correct current to charge the selected type of battery.

The Block Diagram for the ADC is given below. How they are specifically done is within the ATMEGA1284's system and the explanation is given in the manual for ATMEGA1284.

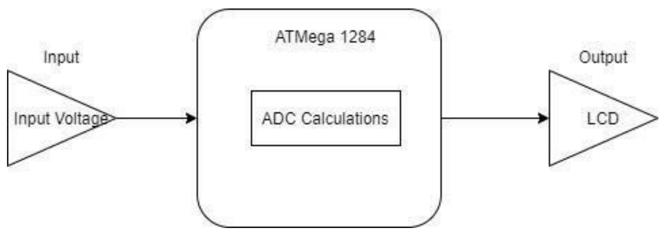


Figure 5.3.1: Software Block Diagram for ADC

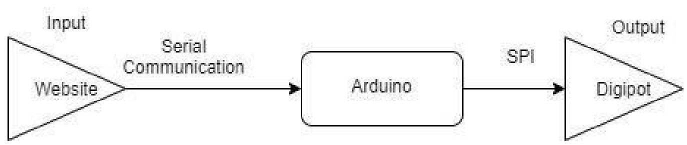


Figure 5.3.2: Software Block Diagram for Website

5.4 Rationale and Alternatives

The chosen hardware architecture was used because there were not many more possible choices. Switch would not work in our circuit because the user of our project would not be able to automatically switch between Constant Current and Constant Voltage phase which will cause inconvenience. To allow automatic switches, we also thought of other forms of automatic switches, but they were not able to take the power the team is trying to input which made the team end up choosing the high power relay. The relay allows for automatic switching between CC and CV. It can handle enough power as it is a high power relay so there would be no shortage of any sort and no malfunctioning because too much power is being put into the relay. Therefore it is the most optimal choice even though there's only about 3 options possible for the project.

The charger also needs voltage input. The voltage input to match the required constant current source is much too high for the battery to handle. First voltage divider was considered, however the voltage from the divider is not stable therefore it is not applicable to the system. So, the second choice, buck converter is used. The buck converter stabilizes and steps down DC voltage, and the ACCS circuit is required to deliver high current to the battery. Some design tweaks could have been possible for either the buck converter or the ACCS circuit, but they would not have significant output changes. If the buck converter and ACCS circuit were not used, a pulse charging design would have been required which is made up of different circuits as mentioned above. The circuits other circuits considered are too complex and will not provide a stable output as the chosen buck converter does.

The charger also needs a constant current source. For the constant current source there were a couple of options that were able to be found online, but after testing with all of them the Active Current Load Circuit was the only choice that worked. Other current sources did not supply enough current or were limited which would cause our project to not work. One of the alternative choices was the active transistor current circuit, but with the amount of power the project needed the transistor would burn even with a heat sink.So the team finalized their decision on using an active current load circuit source for the CC phase.

The charger could have made it so that the buttons could be on the Charging Unit itself, however to give more statistical data of the Battery such as its health, making a website was the best choice which is why the team made a website instead of buttons. Also buttons would have limited the amount of lithium ion battery types as more and more batteries with different mAh are being produced in the industry and the project can simply adjust to that by changing the code on the website. The reasons above make the website for the types of battery a better choice than buttons on the unit. Also the buttons would make the unit harder to use making it less user friendly and dangerous as a child can press the button without knowing what it does maybe causing the battery to burn because of too much power being supplied to it.

These simple software architectures are used because this project does not involve much software. Other softwares that were considered were too difficult and not user friendly making the project have a downside. (One of the biggest goals of the project was to make it so anyone can use it.) Also Arduino was a lot easier to use because it already connects with a microcontroller , ATMEGA1284, allowing a simpler design for the project.

The ADC was the method that made the most sense to read battery voltage because it could be directly calculated using the following equation [3]. Also the other alternatives of reading the battery voltage were not as simple as ADC. One of the other alternatives that the team considered was the coulomb counter. The team attempted to try and understand how it worked, but there was no way to fit the coulomb counter or read off it. Since it was reading the battery voltage from the coulomb counter was already a problem the team just decided to use something more simple which is the ADC. Even though the coulomb counter is much more accurate than reading the battery voltage with a microcontroller, the risk of not being able to completely learn how to implement the counter in the project outweighed the need for accuracy. After several tests the team concluded that ADC was already accurate enough and the ADC was already part of ATMEGA1284 itself, the decision to use ADC was made.

$$ADC = 1024 * \left(\frac{V_{in}}{V_{ref}}\right)$$

Serial communication was chosen to connect the website and the Arduino because it was the simplest form of communication that could be used. Alternatives could include using UART for bluetooth. The UART was a little bit more complex and difficult to use making serial communication the choice for the team. The Arduino must communicate with the digipot, but the digipot does not have to respond. The Enable pin must be high for the MOSI to communicate to the slave with SCK. Alternatives could include using UART for bluetooth communication between the computer (website) and the ATMega 1284 or Arduino; however, more time would have been required to develop this.

The project is made to be user friendly as possible so the biggest part of why certain modules and parts were decided to be used was because of how user friendly it could be. The Adaptive Battery Charging Unit can be used by anyone and is safe that no power will be supplied while the switch is off and is easily distinguishable between the on and off state. Besides that some parts were considered much too expensive and even though some parts could have given more accurate results, the project itself already gives close enough results with a margin of error of less than 1% for every thing that is calculated. Therefore, the team has come up with the most ideal parts and modules to be used making the overall project successful.

6 Data Structures (N/A)

6.1 Internal Software Data Structure

Not applicable.

6.2 Global Data Structure

Not applicable.

6.3 Temporary Data Structure

Not applicable.

6.4 Database Descriptions

Not applicable.

7 Low Level Design

The main battery charging system consists of 3 different subsystems: the Battery Charging Core, LED Display, and Website for Battery type. In general, the system works as described below:

a. Battery charging core supplies power to the batteries in the form of CC and CV.
b. Battery voltage is read by the ATMega 1284 and the LED display. The ATMega 1284 displays battery percentage and voltage onto the LCD display.
c. The website controls resistance values on a digital potentiometer to control the output current of the ACCS circuit for the CC charging phase.

The subsystems, or modules, that make up the overall system are explained below.

7.1 Module 1: Battery Charging Core

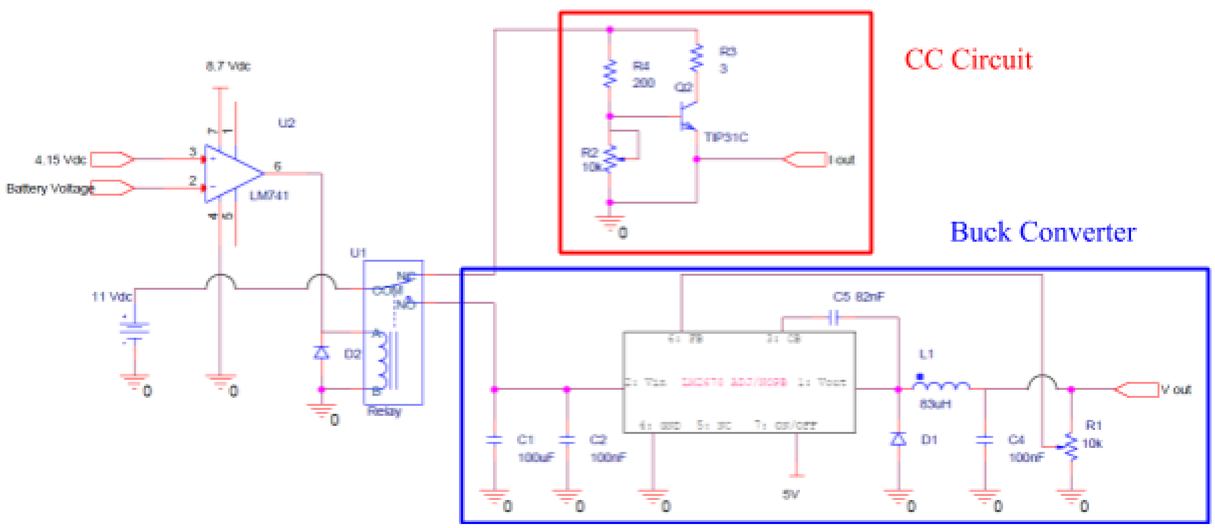


Figure 7.1: Battery Charging Core Circuit Schematic

7.1.1 Processing Narrative for Battery Charging Core

The Battery Charging Core consists of all the input power circuits that feed into the battery. In the schematic above, V_{out} and I_{out} are locations where the battery's positive terminal is connected. The top left of the schematic consists of the comparator which will switch the relay from CC to CV when the battery voltage reaches 4.18 V. Note the flyback diode D2 next to the relay which is important for safety reasons with the induction coil inside the relay [12].

7.1.2 Battery Charging Core Interface Description

The I_{out} and V_{out} nodes are locations where the positive terminal of the battery is connected. Because of this, these nodes are also shared with the Sig pin of the LM3914 of the LED indicator module and the ADC input of the microcontroller. The digipot (R2) is directly controlled by the website module to control the charging current based on the battery type.

7.1.3 Battery Charging Core Processing Details

This circuit receives an 11 V input from the power supply which is fed into the relay. Another input of 8.7 V is applied to pin seven of the comparator to apply voltage to switch the relay. The comparator does the automatic switching between CC and CV by comparing the voltage value of the battery to a preset voltage of 4.18 V. When the relay is at the NC terminal (CC), the ACCS circuit delivers 1.3 A to the battery. The digipot (R2) in the ACCS circuit controls the output current by varying its resistance. When the relay is switched to the NO terminal (CV), the buck converter steps down the 11 V input to desired value of 4.2 ±
0.03 V.

7.2 Module 2: LED Indicator

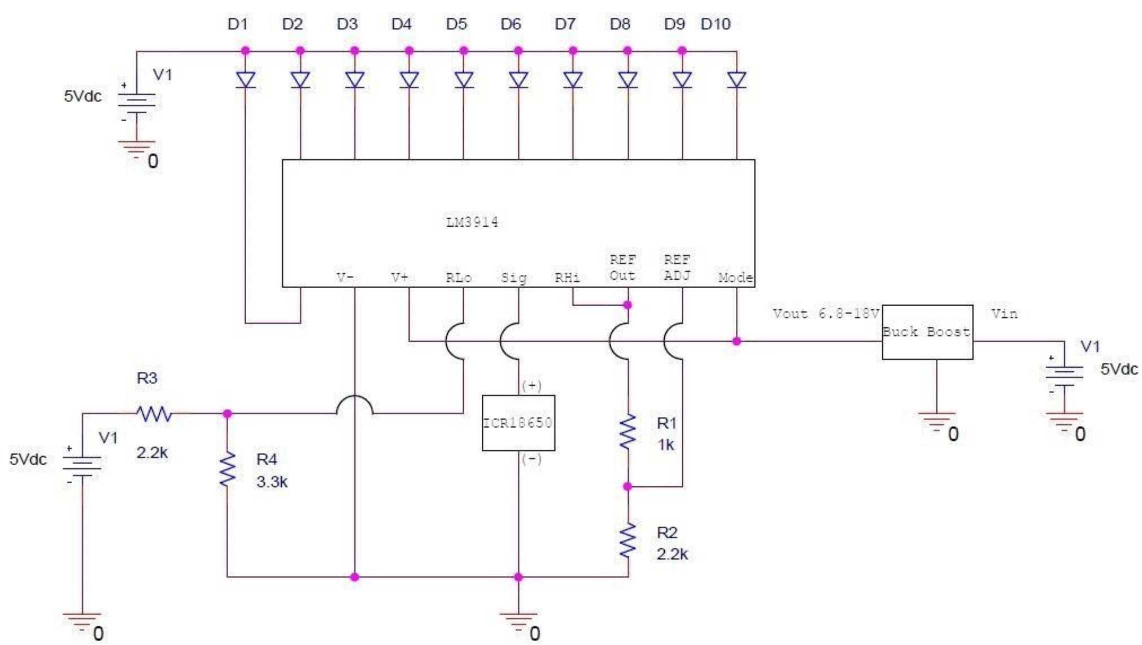


Figure 7.2: LED Indicator Circuit Schematic

7.2.1 Processing Narrative for LED Indicator

The LED indicator's function is to display the SoC of the battery in the form of a bar graph. LEDs turn on from left to right at 10 different voltage values to represent the battery's SoC. However, it is to be noted that the LEDs do not represent equal divisions in battery percentage.

7.2.2 LED Indicator Interface Description

The Sig pin of the LM3914 is directly connected to the battery's positive terminal, meaning that it also shares a node with the output of the buck converter and the ACCS circuit.

7.2.3 LED Indicator Processing Details

The LED indicator's main processor is the LM3914 IC. This chip automatically splits a specified analog voltage range into 10 equal subdivisions and turns on LEDs from left to right depending on the voltage that is read. The minimum voltage value is input to the RLo terminal and the maximum voltage value is input to the RHi terminal. Input voltages to the Sig pin that are smaller than the voltage at RLo will not turn on any LED. Input voltages to the Sig pin that are larger than the voltage at RHi will not have any extra effect.

7.3 Module 3: Website

The flowchart for website giving a generic idea of how pressing buttons work and how the data is sent through between each state.

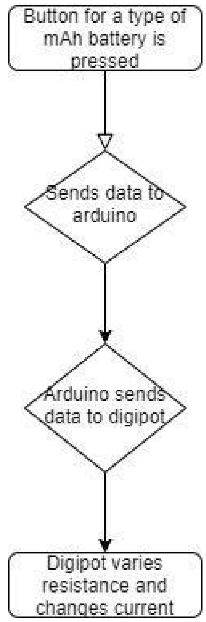


Figure 7.3: Flowchart for Website Communication with Arduino

7.3.1 Processing Narrative for Website

The website's function is for the user to be able to choose what type of battery they want to charge. Depending on the button pressed on the website, the user will be able to vary the output current of the battery charging core making it adjustable to anytype of lithium batteries as long as it's within the system.

7.3.2 Website Interface Description

The website sends information to the Arduino [8], [9] which sets the digipot resistance values. This digipot is the one that is present in the ACCS circuit (labeled R2 in the schematic). Changing the resistance value of the digipot alters the current output of the ACCS circuit. An example of the relationship between charging current and R2 resistance value can be observed in section 4.2.

7.3.3 Website Processing Details

When one of the four buttons on the website is pressed, information is sent to the Arduino through serial communication. For example, if the 2600 mAh button is pressed, a digitalWrite command is sent to the arduino to set pin 3's value to 1. The Arduino then checks the pin value and sets the digipot value depending on what value the pin was set to. Details regarding what the website looks like can be found in section nine.

7.4 Summary of Low Level Design

In the overall system, the "Battery Charging Core" module sends its outputs to the battery for the charging process. The LCD and LED indicator modules read the battery voltage and display it in their respective output forms. The LCD displays the battery voltage along with the SoC in percentage form, and the LED indicator displays SoC in the form of an LED bar graph. Charging current in the CC phase is controlled from the Website module which sends information to an Arduino through serial communication. Depending on the information that the Arduino receives, the Arduino will send values to the digipot through SPI to change its resistance, resulting in the variation of charging current.

8 Technical Problem Solving

8.1.1 The Buck Converter Problem

When testing the ideal buck converter model as a proof of concept, it did not adequately step down voltage when a load was applied on the output. Siwon Kim was responsible for identification of this problem.

8.1.2 Solving the Buck Converter Problem

The problem with the buck converter not stepping down voltage correctly could not be solved. This was due to the fact that the PWM has to change every cycle to compensate for the load at the output. To overcome this, the LM2678 buck converter IC was purchased and a new buck converter circuit was developed. This IC was chosen because it operates at 260 kHz and automatically adjusts the PWM to compensate for a load at the output. Using this IC and a new circuit fixed all problems with the buck converter.

8.2.1 The Digipot Problem

When attempting to implement the digipot with the Arduino, the MOSI pin was not properly sending information. Siwon Kim was responsible for identification of this problem.

8.2.2 Solving the Digipot Problem

This problem occurred because the wire connecting the MOSI pins on the Arduino and digipot was broken. As a result, SPI could not work and communication between the Arduino and digipot never occurred.

8.3.1 The LCD Problem

While attempting to implement the LCD screen to display battery voltage and SoC, the LCD did not initialize no matter what iteration of code was used. Therefore, nothing was ever being displayed on the LCD. Timothy Perrier was responsible for identification of this problem.

8.3.2 Solving the LCD Problem

After searching for a fix, the solution to the problem was found in the library file which contained the control of the LCD. The pins labeled in the library file did not correspond to the pins that the LCD was connected to, so the LCD could never communicate with the ATMega 1284. Rather than change the pin connections between the LCD and ATMega 1284, the library file was changed to reflect the chosen pin connections.

8.4.1 The ADC Problem

Testing and implementing ADC with the ATMega 1284 was done with LEDs. A red LED was programmed to turn on when input voltage was below 2.5 V while a green LED was programmed to turn on above 2.5 V. No matter which iteration of code, the LEDs never turned on or off with respect to applied voltage. Timothy Perrier was responsible for identification of this problem.

8.4.2 Solving the ADC Problem

Since LEDs were not working with ADC, the solution was found by displaying ADC output data on the LCD. The ADC was actually working the entire time, but for an unknown reason the LEDs would not react to ADC values. To make sure the displayed voltage values on the LCD were correct, the BK Precision 5492B was used to measure the actual input voltage and the compared values were within a 5% error margin.

9 User Interface Design

9.1 Application Control

The website has the title "Lithium Ion Battery Selector" in the top center of the page. The website also contains four buttons organized in a row that are labeled with mAh ratings. Clicking the buttons with the mouse causes the buttons to change to a blue color while the click is held. The website was designed using HTML programming on Sublime Text 3.

9.2 User Interface Screens

An example output of the LCD screen can be observed in Figure 9.2a below. On the top row, "BattV" is the abbreviation for "Battery Voltage". On the bottom row, "BattPcnt" is the abbreviation for "Battery Percentage". This screen is updated every 15 seconds.

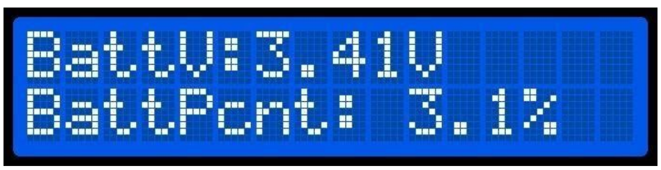


Figure 9.2a: LDC Interface

A screenshot of the website - when first accessed - can be observed in Figure 9.2b on the left (below). When a button is pressed, the mAh value will display at the bottom where "Selection:" is located. This change can be observed in Figure 9.2b on the right.

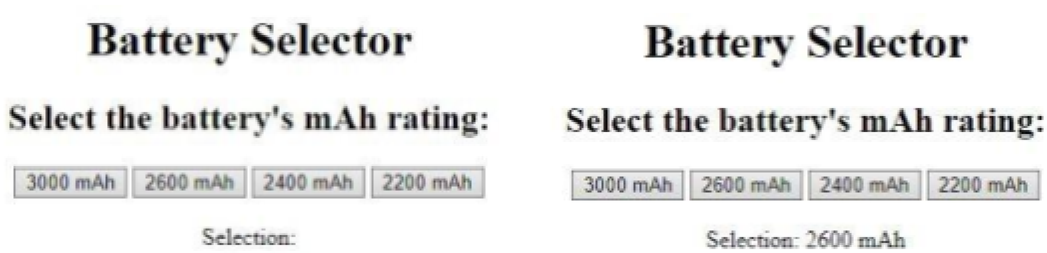


Figure 9.2b: Website Interface With and Without Selection

10 Test Plan

10.1 Test Design

Tests and expected responses

1. Buck Converter
   a. Person responsible: Siwon Kim
   b. Expected response: output of 4.2 ± 0.03 V
2. CC Circuit
   a. Person responsible: Timothy Perrier
   b. Expected response: output of 1.3 A
3. LCD Screen
   a. Person responsible: Timothy Perrier
   b. Expected response: display battery voltage within a ± 5% error margin in comparison to a multimeter reading
4. LED Indicator
   a. Person responsible: Timothy Perrier
   b. Expected response: turn on 10 LEDs from left to right at the following voltages. Error margin allowed is ± 10%

LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED8	LED9	LED10
3.098V	3.221V	3.333V	3.452V	3.571V	3.685V	3.805V	3.927V	4.043V	4.167V

5. Digital Potentiometer and Website
   a. Person responsible: Timothy Perrier
   b. Expected response: Clicking buttons on the website will change charging current value in CC circuit. The current values for each button are listed below. Error margins are the same as in the CC Circuit test case.

3000 mAh	2600 mAh	2400 mAh	2200 mAh
1.5 A	1.3 A	1.2 A	1.1 A

Test Case 1:

1. The objective of this experiment is to test the output of the buck converter and make sure it meets the required standards.
2. Experiment setup:
The buck converter circuit is built, the power supply is off before testing, and the multimeter is probing the output of the buck converter so that it is ready to measure the voltage.
3. Experimental Procedure and Data Collection

    a. Connect the positive (red) and negative (black) banana plugs to the corresponding pins on the BK Precision 1760A's B supply (left). Refer to Figure 1 in Appendix B for labeled pictures to help. Connect a red wire to the crocodile clip on the other end of the banana plug that is connected to the positive pin and plug it into the positive (red) rail on the breadboard. Connect a black wire to the crocodile clip on the other end of the banana plug that is connected to the negative pin and plug it into the negative (blue) rail on the breadboard.

    b. Connect the BK Precision 5492B multimeter to the output of the Buck Converter to measure the output voltage. Insert the positive banana plug (red) into the top-right pin of the multimeter and the negative banana plug (black) to the pin directly underneath. Refer to Figure 2 in Appendix B for labeled pictures to help. Connect the positive wire to the output of the Buck Converter and the negative wire to the negative rail on the breadboard.

The banana plug connections can be seen in the picture below.

    c. Turn on the power supply using the On/Off switch in the center of the front panel. Set the power supply voltage to 11 V by turning the voltage adjustment knob in the clockwise direction. This knob is labeled on the picture from Step 1.

    d. Turn on the multimeter. By default, DC V should already be selected. If not, select the "DC V" button to measure DC Voltage. The location of this button is shown in the picture above.

    e. Check the multimeter display to confirm that the value is within the range of 4.2 ± 0.03 V.

    f. If not turn the knob on the potentiometer until the required value of 4.2 ± 0.03 V is achieved.

4. Expected results:

With an input of 11 V, the output obtained must be 4.2 ± 0.03 V.

Test Case 2:

1. The objective of this experiment is to test the output of the CC Circuit and make sure that it meets the required standards.
2. Experiment setup

Our Active Current Circuit is open on the part where we need to measure current and the multimeter is probed in the open area to measure the current of our circuit. The power supply is off and the circuit doesn't have any power flowing through it until our experiment starts.

3. Experimental Procedure and Data Collection
   a. Connect the positive (red) and negative (black) banana plugs to the corresponding pins on the BK Precision 1760A's B supply (left). Refer to Figure 1 in Appendix B for labeled pictures to help. The connections are labeled in the picture below. Connect a red wire to the crocodile clip on the other end of the banana plug that is connected to the positive pin and plug it into the positive (red) rail on the breadboard. Connect a black wire to the crocodile clip on the other end of the banana plug that is connected to the negative pin and plug it into the negative (blue) rail on the breadboard.

   b. Connect the BK Precision 5492B multimeter to the output of the CC circuit to measure the output voltage. Insert the positive banana plug (red) into the middle-left pin of the multimeter and the negative banana plug (black) to the pin directly to its right. Refer to Figure 3 in Appendix B for labeled pictures to help. Connect the positive wire to the output of the Buck Converter and the negative wire to the negative rail on the breadboard.

The banana plug connections can be seen in the picture below.

   c. Turn on the power supply by flicking the On/Off switch on the center of the front panel to the On position. Set the voltage to 11 V by turning the voltage adjustment knob in the clockwise direction.

   d. Turn on the multimeter by pressing down on the power button on the bottom left of the front panel. To display current, press the "Shift" button in the bottom right followed by the "DC V" button in the top left. These buttons are labeled in the picture above.

   e. The current displayed on the multimeter should be at 1.3 ± 0.02 A. If not, rotate the knob on the potentiometer until the required value of 1.3 ± 0.02 A is achieved.

Warning: do not touch any component in the circuit as they may get very hot.

f. For safety reasons, upon achieving the required value, set the input voltage back to 0 V and turn off the power supply.

4. Expected Results:

With an input of 11 V, the output current should be 1.3 ± 0.02 A.

Test Case 3:

1. The objective of this experiment is to test the functionality and accuracy of the LCD screen.
2. Experiment setup

The power supply isn't needed as a 5V voltage regulator is in our circuit. LCD is connected to ATMEGA1284 microcontroller unit. The battery is yet to be in the charging case. There's no need for a multimeter as our LCD screen probes and measures the voltage and percentage of the battery. Power is off as usual before the experiment starts.

3. Experimental Procedure and Data Collection

a. Connect the positive (red) and negative (black) banana plugs to the corresponding pins on the BK Precision 1760A's B supply (left). The connections are labeled in the picture below. Connect a red wire to the crocodile clip on the other end of the banana plug that is connected to the positive pin and plug it into the ATMega1284's Pin A0. Connect a black wire to the crocodile clip on the other end of the banana plug that is connected to the negative pin and plug it into the negative (blue) rail on the breadboard.

b. Turn on the 5 V supply that is connected to the power rails of the breadboard by pressing down on the white rectangular button.

c. Turn on the BK Precision 1760A power supply by flicking the On/Off switch in the center of the front panel to the On position.

d. The top line of the LCD should display "BatV:0.00 V" and the bottom line should display "BatPct: 0.00 %". Slowly turn the voltage adjustment knob on the BK Precision 1760 power supply counter clockwise. Do not go above 6 V as there is a risk of damaging the microcontroller. Check the LCD screen and make sure that it displays a voltage that is within 5% of the input voltage. Make sure that the percentage at the input voltage correctly corresponds to the range on the annotated discharge curve in Figure 1 of Appendix D.

4. Expected Results:

The battery voltage on the LCD screen should be correct with an error margin of ± 5% in comparison to the multimeter reading. The battery percentage (SoC) should correspond to the values calculated from the discharge curve. For example, if the battery voltage is between 3.53 V and 3.57 V the value of "BatPcnt" should be within the calculated range of percentage from the annotated discharge curve in Appendix F.

Test Case 4:

1. The objective of this experiment is to test the functionality and accuracy of the LED indicator.
2. Experiment Setup

The power supply isn't needed as a 5V voltage regulator is in our circuit. LED IC LM3914 is circuited correctly to each LED bus of our indicator. The battery is yet to be connected and the multimeter is probed at each end of the battery so it can match the voltage read with the % we calculated.

3. Experimental Procedure and Data Collection

   a. Connect the positive (red) and negative (black) banana plugs to the corresponding pins on the BK Precision 1760A's B supply (left). Refer to Figure 1 in the Supplements section for labeled pictures to help. Connect a red wire to the crocodile clip on the other end of the banana plug that is connected to the positive pin and plug it into Pin 5 of the LM3914 IC chip. Connect a black wire to the crocodile clip on the other end of the banana plug that is connected to the negative pin and plug it into the negative (blue) rail on the breadboard.

   b. Turn on the 5 V supply that is connected to the power rails of the breadboard by pressing down on the white rectangular button.

   c. Turn on the BK Precision 1760A by flicking the On/Off switch to the on position.

   d. Turn the voltage adjustment knob, in the clockwise direction to increase the voltage to 2.5 V. The LED bus should still remain off.

   e. Turn the voltage adjustment knob clockwise until 4.2 V is achieved. The LEDs on the bus should turn on from left to right approximately at the input voltages in the table in the beginning of the Test Design section. Upon reaching an input of 4.2 V, turn the voltage adjustment knob counter clockwise to reduce the voltage back to 2.5 V. The LEDs should turn off from right to left approximately at the voltages in the table above.

4. Expected Results:

LEDs on the array should turn on from left to right at the voltages listed in the table which is in the beginning of section 10.1. An error margin of approximately 10% should be accounted for.

Test Case 5:

1. The objective of this test is to test the communication between a website and an Arduino which controls the charging current.
2. Experimental Setup

The experimental setup is the same as in test case 2, but with the Arduino, digipot, and the website.

3. Experimental Procedure and Data Collection

    a. Connect the positive (red) and negative (black) banana plugs to the corresponding pins on the BK Precision 1760A's B supply (left). Refer to Figure 1 in Appendix B for labeled pictures to help. The connections are labeled in the picture below. Connect a red wire to the crocodile clip on the other end of the banana plug that is connected to the positive pin and plug it into the positive (red) rail on the breadboard. Connect a black wire to the crocodile clip on the other end of the banana plug that is connected to the negative pin and plug it into the negative (blue) rail on the breadboard.

    b. Connect the BK Precision 5492B multimeter to the output of the CC circuit to measure the output voltage. Insert the positive banana plug (red) into the middle-left pin of the multimeter and the negative banana plug (black) to the pin directly to its right. Refer to Figure 3 in Appendix B for labeled pictures to help. Connect the positive wire to the output of the Buck Converter and the negative wire to the negative rail on the breadboard.

The banana plug connections can be seen in the picture below.

    c. Turn on the power supply by flicking the On/Off switch on the center of the front panel to the On position. Set the voltage to 11 V by turning the voltage adjustment knob in the clockwise direction.

    d. Open the website.

    e. Click on each button and observe the change in current that is displayed on the multimeter.

4. Expected Results:

Clicking on buttons on the website should change the charging current value according to the table in the beginning of section 10.1.

State clearly who is responsible for which experiment

10.2 Bug Tracking

This is more of a future development as there is no database, but we have kept some bugs in a notebook along the way of the project such as:
LCD SCREEN NOT DISPLAYING CORRECTLY
BUCK CONVERTER DOES NOT OUTPUT CORRECTLY
BATTERY DOES NOT TAKE CHARGE CURRENT CORRECTLY

The person assigned to fix bugs was always the person responsible for that specific subsystem. Occasionally, group members would help out if needed or if the bug was difficult to solve. All of these bugs were solved and if there are any more to come, we'll develop a debugger in our system that can easily identify and fix the bugs. No more bugs exist in the final version of the project.

10.3 Quality Control

The test plan and procedure from sections 10.1 was used to document the results in the test report section. Expected values were obtained for every test.

10.4 Identification of critical components

The two most critical components of the design are the ACCS circuit for CC and the buck converter for CV. The ACCS circuit demands great attention because it is the most important part of the project as it charges the bulk of the battery. If the ACCS circuit does not meet the requirements, it would have to be redesigned. The buck converter is less important than the ACCS circuit, but it is still significant because the battery needs to be fed a constant voltage with very minimal fluctuations for the CV phase.

10.5 Items Not Tested by the Experiments

No subsystems were left untested.

11 Test Report

11.1 Test 1: Buck Converter

1. Person performing the test: Siwon Kim Figure 11.1.1: Test Results for Buck Converter

Test Iteration	Input Voltage (V)	Output Voltage (V)
1	11	4.211
2	11	4.198
3	11	4.208
4	11	4.205
5	11	4.181
6	11	4.212
7	11	4.201
8	11	4.227
9	11	4.223
10	11	4.225

2. Every measured output value was within the 0.03 V margin.
3. Every resulting output voltage was within the 0.03 V margin of 4.2 V that was specified in the battery datasheet. Fluctuations in the output voltage results are most likely due to the power supply not delivering exactly 11 V or imperfect resistance values.
4. No corrective actions needed.
5. The test passed with pass/fail rate: 10/0 out of 10 tests

11.2 Test 2: CC Circuit

Person performing this test: Timothy Perrier

1. Figure 11.2.1: Test Results for CC Circuit

Test Iteration	Input Voltage (V)	Output Current (A)
1	11	1.284
2	11	1.256
3	11	1.313
4	11	1.296
5	11	1.315
6	11	1.307
7	11	1.291
8	11	1.322
9	11	1.316
10	11	1.325

2. The measured current values over 10 tests are always close to the required 1.3 A error: 0.02A.
3. Just like the buck converter output voltages, the CC circuit output current values are not perfect. This is due to random errors such as imperfect input voltage and resistance values. Another factor that could have caused these fluctuations between measured values is the fact that the BJT heats up with high currents. High heat can have impacts on the output performance of the circuit.
4. No corrective actions needed.
5. The test passed with pass/fail rate: 9/1 out of 10 tests
    a. the test error is rounded to nearest hundredths

11.3 Test 3: LCD Screen

Persons performing this test: Timothy Perrier, Tae Kim

1. Figure 11.3.1: Test Results for LCD Screen

Test Iteration	Input Voltage (V)	Multimeter Voltage Reading (V)	LCD Voltage Reading (V)
1	3.08	3.0856	3.08
2	3.16	3.1563	3.15
3	3.3	3.2883	3.28
4	3.4	3.3982	3.39
5	3.5	3.4878	3.47
6	3.6	3.5977	3.58
7	3.7	3.6956	3.69
8	3.8	3.7869	3.79
9	3.9	3.8894	3.88
10	4.0	4.0452	4.03

1. Test pass requirement: within 5% error of input voltage
2. The test passed with pass/fail rate: 10/0 out of 10 tests

11.4 Test 4: LED Display

Person(s) performing this experiment: Timothy Perrier, Tae Hyun Kim

1.

LED1	**LED2**	**LED3**	**LED4**	**LED5**	**LED6**	**LED7**	**LED8**	**LED9**	**LED10**
3.098V	3.221V	3.333V	3.452V	3.571V	3.685V	3.805V	3.927V	4.043V	4.167V

We compared at which percentage on the LCD screen the LED Display would turn on and this table gives the exact values when a single LED turns on.

2. The calculated LED values turn on at that battery voltage all the time as expected
3. There are times where the LED stops working or either displays wrong, but usually when debugged we are able to fix it most of the time. Since the LED reads off the battery voltage we tested by using a power supply first to match the values indicated above to test
4. No corrective actions needed

11.5 Test 5: Digipot and Website

Person(s) performing the experiment: Timothy Perrier

1. Test results, person performing the test

Test Iteration	Type of Battery pressed on website	Resistance Value for Digipot sent	Multimeter current readings
1	3000mAh	10	1.52A
2	2600mAh	5	1.34A
3	3000mAh	10	1.54A
4	2600mAh	5	1.31A
5	3000mAh	10	1.51A

2. The multimeter reading is +/-0.05A for desired current
3. Although there’s some noise and differences between different iterations of the same type of button and digipot value, the values are within the range that is sufficient enough for the project results to be considered correct.
4. No corrective actions needed

12 Conclusion and Future Work

12.1 Conclusion

This project met most of the objectives that were set in the beginning of the process. All of the technical aspects of the project met the set requirements. The charger is able to switch between the CC and CV charging phases at the correct time, and it supplies the correct amount of current and voltage. The charger was able to charge a battery in the targeted time frame of 3-4 hours. No safety issues occured when the battery was being charged, and the temperature was well in the safe zone of 0-45℃. The LCD displayed battery voltage and SoC with an error margin that is much smaller than what was anticipated. Timothy Perrier was responsible for the LCD, LED indicator, website controlling the Arduino, and the design of the CC circuit. Siwon Kim was responsible for designing the buck converter and programming the digipot. Tae Kim was responsible for systems engineering documentation, helping in LED indicator, LCD indicator, calculations for SoC, and assisted circuit building and testing for CCCV.

The only technical objective that was not met was bluetooth control using either a website or a smartphone application. This objective was not met due to lack of time; however, serial communication was used instead for communication with a localhost website. Other objectives that were not met were non-technical. These include having an encasement for all the circuitry and having everything on a PCB. Timothy Perrier was responsible for not being able to get Bluetooth to function. Brandon Lam was responsible for the lack of having an encasement and the circuit on a PCB.

Timothy Perrier: This project taught me a lot about power electronics and batteries. Working on this project also helped me grow professionally a significant amount. Project leadership and group communication were the main aspects that I was able to grow. When nobody stepped up to make decisions, I was the one who took the initiative. I realized how important communication between group members is especially when trying to design certain elements or fix problems.

Siwon Kim: This project helped me to understand that the practical and ideal circuit is completely different. There have been many times where theoretical circuits work, but practical circuits do not because some component fails. There were some obvious and not so obvious instances of component failure. For an obvious reason, an IC fails because I put more voltage than the datasheet says the IC can handle. For a not so obvious reason, a wire for the ISP pin broke and not transmitting bits correctly. The difference in practical and ideal circuits made me realize the need to keep trying to understand the limits of circuit designs. This project taught me to look at designs in detail so I don’t waste time later trying to debug the circuit.

12.2 Future Work

Since there are chargers on the market that can charge a variety of battery types (not just LiBs), our project can be changed in the future to provide a wider range of charging currents and voltages. Our LED indicator can be improved upon and displaying on our LCD can be improved as well. In terms of the CV phase, the buck converter is well built but CC can be improved by building another type of CC circuit. A significant improvement that can be worked on in the future is the ability to charge multiple batteries at the same time. Overall, our project shows a considerable amount of success with minimal future improvements for other features to be added.

12.3 Acknowledgement

Dr. Roman Chomko

- Associate Professor of Teaching, Department of Electrical and Computer Engineering, UCR
- Project Director

Dr. Hossny El-Sherief

- Lecturer, Department of Electrical and Computer Engineering, UCR
- Documentation, Project Planning, and Systems Engineering

13 References

[1] "Active Transistor Constant Current Source", *Electronics Notes*, [Online]. Available: https://www.electronics-notes.com/articles/analogue_circuits/transistor/active-constant-current-sourc e.ph p. [Accessed: 14-Mar-2020]

[2] "Adaptive Charging: How it Works", *Victron Energy*, [Online]. Available: https://www.victronenergy.com/upload/documents/Adaptive_charging_how_it_works.pdf. [Accessed: 14-Mar-2020]

[3] Atmel, "ATmega 1284", 42718A, Oct. 2016

[4] "Battery Chargers and Charging Methods", *The Electropaedia: Battery and Energy Technologies*,

[Online]. Available: https://www.mpoweruk.com/chargers.htm. [Accessed: 14-Mar-2020]

[5] clark1795. "15 Best 18650 Battery Charger Options: Your Buyer's Guide." *Heavy.com*, 11 May 2018, heavy.com/vape/2017/11/top-best-18650-battery-charger-vape/.

[6] Fairchild, "TIP31A TIP31C - NPN Epitaxial Silicon Transistor", TIP31C Datasheet, Nov. 2014

[7] Great Power, "ICR18650 2600mAh Datasheet", ICR18650 2600mAh Datasheet, Jul. 2012

[8] Instructables. "Controlling Arduino With Python Based Web API (No Php)." *Instructables*, Instructables, 9 Oct. 2017, www.instructables.com/id/Controlling-Arduino-with-python-based-web-API-No-p/.

[9] Instructables. "Pyduino, Interfacing Arduino With Python Through Serial Communication." *Instructables*, Instructables, 9 Oct. 2017, www.instructables.com/id/Pyduino-Interfacing-Arduino-with-Python-through-se/.

[10] Leng, F., Tan, C. & Pecht, M. Effect of Temperature on the Aging rate of Li Ion Battery Operating above Room Temperature. *Sci Rep* 5, 12967 (2015). https://doi.org/10.1038/srep12967

[11] Texas Instruments "LM2678 Simple switcher High efficiency 5-A Step Down Voltage Regulator" SNVS029K data sheet, March 2000[Revised February 2017]

[12] “Relay Switch Circuit and Relay Switching Circuit.” *Basic Electronics Tutorials*, Electronics Tutorial, 16 Feb. 2018, www.electronics-tutorials.ws/blog/relay-switch-circuit.html.

[13] “Test of Samsung ICR18650-26F 2600mAh (Pink)”, *Lgyte Info,* [Online]. Available: https://lygte-info.dk/review/batteries2012/Samsung%20ICR18650-26F%202600mAh%20(Pink)%2UK. html. [Accessed: 14-Mar-2020]

[14] Texas Instruments “LM3914 Dot/Bar Display Driver” SNVS761B datasheet, January 2000[Revised March 2003]

[15] “The ADC of the AVR”, *maxEmbedded*, [Online]. Available: http://maxembedded.com/2011/06/the-adc-of-the-avr/. [Accessed: 14-Mar-2020]

14 Appendices

Appendix A: Parts List

- TIP31C High-Power NPN BJT
- High-Power N-MOSFETs
- High-Power Relay
- Toroid Core Inductor Copper Wind Wound 100uH 6A Coil
- Capacitors
- Diodes
- LM3914 Dot/Bar Display Driver
- ATMega1284P
- ICR 18650 Lithium Ion Batteries
- Resistors
- LTA1000G LED Bus
- ICR 18650 Battery Holder
- LM2678T-ADJ/NOPB

Appendix B: Equipment List

Figure 1: BK Precision 1760A Power Supply

Figure 2: BK Precision 5492B Voltage Reading

Figure 3: BK Precision 5492B Current Reading

Appendix C: Software List

Github repository containing all of our code:
https://github.com/tperrierrr/UCR-BatteryCharger-SeniorDesign-2019-20

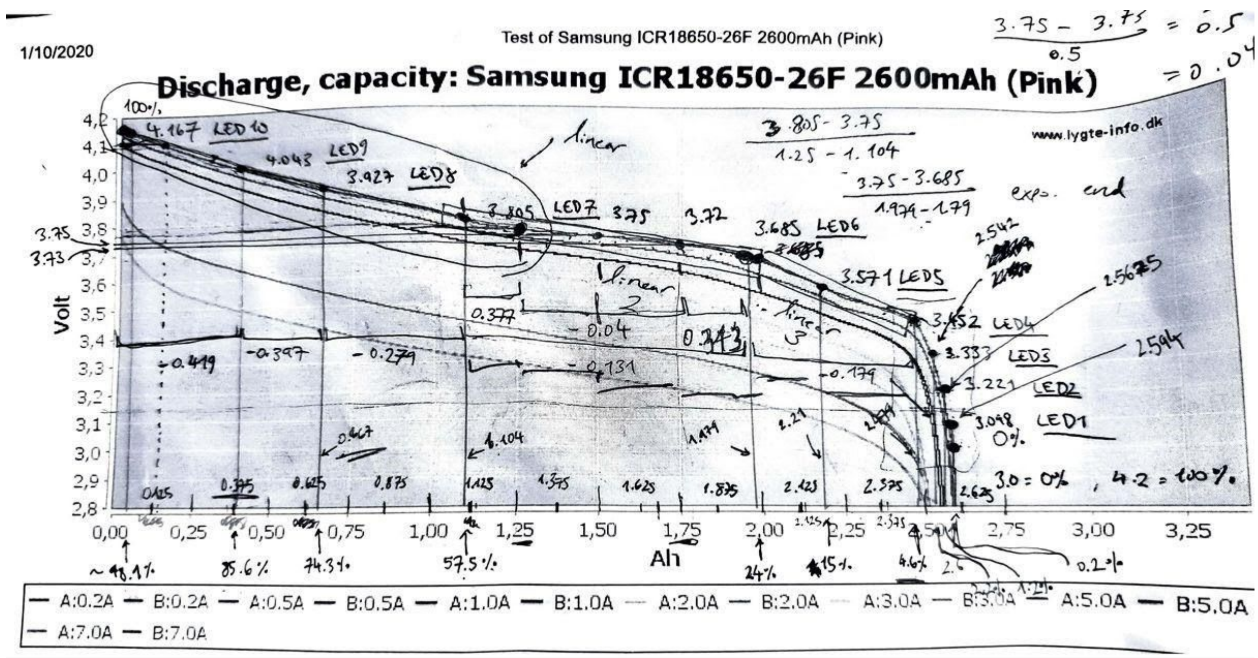
- Atmel Studio 7.0
- Arduino DLE
- Sublime Text 3

Appendix D: Special Resources

- LCD screen
- ATmega1284 (MCU)
- LED registers
- MCP4131 (Digipot)

Appendix F: Battery Discharge Curve

Figure 4: Annotated Battery Discharge Curve used to Calculate SoC Percentage



The graph was taken from [13] and annotated by Timothy Perrier. The regions were split up linearly until the last 5 points where an exponential function was used.