# Table of Contents

Abstract ............................................................................................................................................................................. 3
Introduction ........................................................................................................................................................................ 4
Background ....................................................................................................................................................................... 6
Design Requirements .................................................................................................................................................. 7
Design and Analytical Results ...................................................................................................................................... 10
Testing and Results ........................................................................................................................................................ 18
Conclusion ..................................................................................................................................................................... 21
References ...................................................................................................................................................................... 22
Appendix ........................................................................................................................................................................... 23
Abstract

This project considers the design, implementation, and testing of an open-source dc-dc converter for microgrid prototyping. Unlike conventional dc-dc converters that are proprietary and require specialist knowledge, and are usually designed for a single function, the proposed dc-dc converter will comprise of a programmable MCU and a Raspberry Pi (RPi) interface to allow less-skilled consumers to monitor and modify a power converting system. We will develop an open-source library that contains voltage control, current control, maximum power point tracking, and battery charge control profiles. Each library will be easy to implement through a GUI on the Raspberry Pi and will be controlled using an Atmega328 located on the power conversion unit. C++ and the Arduino IDE will be used for testing and will retain functionality in the finished project for more knowledgeable customers to edit the pre-set profiles. Moreover, the Pi will need to communicate with multiple converters and monitor their set points in applications where more than one dc-dc converter is necessary. The supporting hardware around the microcontrollers is a dc-dc converter, while the connection with the RPi and any external hardware will be open-source and custom designed to accommodate multiple converters on a single system. As a result, the integration of our dc-dc converter will provide a way to easily set up a microgrid system without the use of proprietary voltage converting hardware.
Chapter 1: Introduction

More than 70% of the world’s power is reprocessed or recycled through power electronics [2]. With the push towards more electronic systems and the need for more efficient products this percentage is only going to grow in the future. Also, as more industries become electrified, it is essential that power electronics adapt to become more accessible so that efficient design is within reach of the average consumer. The Open-Source DC-DC converter (OSDDC) that we are developing creates a way for non-engineers to implement an efficient power converter, filling a niche that most other products do not achieve.

Most of this paper goes over the design specifications of the OSDDC, it briefly discusses current designs in the market, and our goals for this project. Currently whenever somebody wants to design an electronic device, the project gets held back or hampered by the lack of easy-to-use power supply controllers. The current interest in power electronics around the world stems from the ever-increasing importance of power conversion. With the increase in renewable energy, wide variety of loads used, and power efficiency the importance is only going to continue to grow.

We propose the OSDDC, so this project is designed to help projects get off the ground and running so the people can focus on the other aspects of the project without worrying about how much power it will be powered. This project attempts to streamline the process of creating a simple dc-dc converter for quick application use so more resources can be sent on the actual project instead of how it will be powered. Most buck converters require specific external hardware to function correctly which may not be achievable for newer hobbyists or people who have more of an interest in programming than they do electronics. Our device would allow those people an easy introduction to power electronics and allow them to build their projects with little hassle. Another party interested would be smaller companies or startups who are not able to hire a power engineer, as this project would reduce the need for a power engineer by providing the building blocks necessary to get started. By also providing open-source code such as battery charging and solar panel Maximum Power Point Tracking (MPPT) the power process is much more streamlined. Then as mentioned above the project can be focused on the programming of the project to implement such as MPPT or battery charging.

To summarize, the current reliance on power electronics to properly process voltage and current for our power supplies is only going to continue to grow. So, to help with that process our design will get small companies and potentially hobbyists as simple power supply so they can focus on the project at hand without worrying about having
to provide power for the project. By providing power as well as simple code for common power electronics configurations such as MPPT and battery charging the efficiency of the project will increase and the power dilemma becomes much more streamlined.
Chapter 2: Background

There are a few products and past senior projects that come close to what we are trying to make. For example, The Maxim integrated maximum power point tracking string-cell optimizer is an IC used to put power point tracking in a solar array [1]. Moreover, there are modules made by hobbyist electronics companies like Adafruit that provide low-power voltage conversion. [6] However, these products are not suitable for a microgrid on their own, require electrical knowledge to be properly implemented, and are only usable for a single application. Our design will be much more adjustable and easier to use in a much wider range of applications.

The design consists of three main parts: Low Level control, High level control and power hardware. Low level control will be just the Arduino and it has low level tasks to complete; sample voltage and current every 1ms, generate the PWM to switches, and to create critically damped current and voltage control loops. The High-level control communicates with the low-level control to implement the different power controls such as MPPT, Battery Charge control, or Current Control for LED drivers [4] [7] [9] [10]. The power electronics for the high-level and low-level code includes current and voltage sensors around the high power MOSFETs. This provides adequate monitoring of the converter to reinforce the control loops of the low-level code.

We expect the results of our power converter to be like those of the proprietary power converters. However, its strength lies in its customizability rather than its ability to outperform the competition. Daniel’s design for a 1kW converter is suitable for many different applications and we intend to implement four configurations with the ability for others to add more.

To confirm that our design is performing as expected we will compare it with previous studies and projects. For example, Leo Hernandez developed a similar proprietary controls scheme for Rantec Power in his senior project [3]. There are also dc-dc voltage converters made for the electrification of gym equipment that both boost and buck the voltage [7] and higher power devices made for an electric vehicle [8]. These examples show that there are applications for a design like ours, and there is no reason that one device cannot incorporate multiple power algorithms, which is why we plan to include maximum power point tracking, and current control.
Chapter 3. Design Requirements

This chapter provides an overview of the system as well as the requirements that guided our design choices. Figure 1 shows the high-level functionality of the system. Communication lines from a Raspberry Pi tell the converter how it should behave according to the incoming data lines. Those data lines relay information about the voltage and current on both the input and output of the converter. Therefore, the converter knows how to manipulate the input voltage to create the Raspberry Pi’s specified behavior on the output voltage.

![High-Level Block Diagram](image)

Figure 2 gives a more detailed look at the design of the converter. Here it can be seen that the Raspberry Pi will be controlling the code which the ATmega will be running, which then controls the MOSFETS present on the DC-DC converter to create a voltage output. Essentially, the transistors act as the switches in the converter and can either be in boost or buck topology depending on the relationship the PWM signal has with the input and output voltage. Moreover, a control loop is necessary to keep the output at a constant value since both input and output can change at any time. For this a proportional controller was run on the voltage error then translated into a duty cycle using the equations.

The ATmega will relay information about the converter, such as its current and voltage data, to the Raspberry Pi. This data will be shown in an easily viewable dashboard that the customer can interact with to change the behavior of the converter, like changing the voltage setpoints and topology. These topologies would be a buck, boost, and current controller, which gives us the controller the ability to drive almost any device. Moreover, Limits were built into the dashboard to make sure that the customer does not exceed the specifications and tolerances of the device, since clean operation is always expected.
Figure 3-2: Lower-Level Block Diagram

Table 3-1: Block Diagram Table

<table>
<thead>
<tr>
<th>Module</th>
<th>Open-Source DC-DC Converter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td></td>
</tr>
<tr>
<td>• Energy Elements</td>
<td></td>
</tr>
<tr>
<td>• Power for Raspberry Pi and AT Mega</td>
<td></td>
</tr>
<tr>
<td>• 5 Volts</td>
<td></td>
</tr>
<tr>
<td>• 2 Amps (Pi)</td>
<td></td>
</tr>
<tr>
<td>• Signal Elements</td>
<td></td>
</tr>
<tr>
<td>• UART from computer flashing code</td>
<td></td>
</tr>
<tr>
<td>• Data Elements</td>
<td></td>
</tr>
<tr>
<td>• Code blocks for the Raspberry Pi</td>
<td></td>
</tr>
<tr>
<td>• MPPT, Battery Charging Code, etc.</td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td></td>
</tr>
<tr>
<td>• Energy Elements</td>
<td></td>
</tr>
<tr>
<td>• Voltage and Current</td>
<td></td>
</tr>
<tr>
<td>• Signal Element</td>
<td></td>
</tr>
<tr>
<td>• Converted Voltage</td>
<td></td>
</tr>
<tr>
<td>Functionality</td>
<td></td>
</tr>
<tr>
<td>• Buck and Boost voltage</td>
<td></td>
</tr>
<tr>
<td>• Runs programmed code with DC-DC converter such as MPPT and battery charging.</td>
<td></td>
</tr>
</tbody>
</table>
The goal of this project was to build an easy-to-use DC-DC converter that anyone with a minimal understanding of electronics could implement for quick prototyping. It needs to balance both ease of use and cost to be something that a customer would realistically choose. To prevent cost the PCB was made as small as possible with readily available components, many of which were optional depending on the budget of the end user. Ease of use was achieved using free software that could run on any computer with USB ports, not necessarily just a Raspberry Pi.

Table 3-2: Customer requirements #1

<table>
<thead>
<tr>
<th>Customer Requirements</th>
<th>Engineering Specification</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>B,C,D</td>
<td>At max power, converter needs to be &gt;98% efficient</td>
<td>MPPT can be used to maximize the efficiency of the DC-DC converter [4]</td>
</tr>
<tr>
<td>A,B,C,D</td>
<td>Source code and hardware schematics open to public</td>
<td>Open-Source software allows small business to implement their products more easily [5]</td>
</tr>
</tbody>
</table>

Table 3-3: Customer Requirements #2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Target</th>
<th>Tolerance</th>
<th>Risk(H,M,L)</th>
<th>Compliance(A,T,S,I)</th>
<th>Test Equipment Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of different applications designed for</td>
<td>4</td>
<td>Min</td>
<td>H</td>
<td>I</td>
<td>Oscilloscope Solar Panel Test Load</td>
</tr>
<tr>
<td>Time to Learn</td>
<td>15 Min</td>
<td>Max</td>
<td>M</td>
<td>-----</td>
<td>Poll</td>
</tr>
<tr>
<td>Bandwidth of Primary Control Loop</td>
<td>1kHz</td>
<td>Min</td>
<td>M</td>
<td>A</td>
<td>Oscilloscope</td>
</tr>
<tr>
<td>Power</td>
<td>1kW</td>
<td>Min</td>
<td>M</td>
<td>T</td>
<td>Test Load</td>
</tr>
<tr>
<td>Overall Cost of Parts</td>
<td>$35</td>
<td>Max</td>
<td>L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 4. Design and Analytical Results

As shown in figure 3 and 4 below the topologies of a buck and boost converter are the same but backwards. By switching the input and output of the two you can turn one topology into the other, which is why the Inverter is able to support two topologies with one design (in modern circuits the diode is replaced with a switch).

As can be seen in the buck and boost topology the duty cycle of the switch is what changes the output voltage as shown in equations below.

\[ \frac{V_o}{V_i} = D \quad (eq \ 4 - 1) \]
Boost: \( \frac{V_o}{V_i} = \frac{1}{1-D} \) (eq 4-2)

These equations create an inversely proportional relationship between the buck and boost topologies. This means that for a buck topology, the duty cycle must increase to create a larger decrease in voltage, which in a boost topology, the duty cycle decreases to create a larger increase in voltage. As a result, we only must multiply the control loop by -1 to switch the topology and pass the voltage into the equations at the end to get the correct duty cycle.

**PI Control**

For the buck and boost control for the voltage and current control a digital PI controller was used to stabilize the system. The proportional part of the control system uses error in the system to see how much is needed to fix the output error as shown in (eq 3). Then in (eq 4) the discrete integral is calculated to find the total error to eventually fix the steady state error of the system.

\[
Proportional = kp \times error \quad (eq \ 4-3)
\]

\[
Integral = \sum ki \times error \times Ts \quad (eq \ 4-4)
\]

The above calculations though only give the “proportional” and “integral” part of the voltage/current, but for the voltage and current control to work properly, this error must be converted into percent error as duty cycle error and voltage/current error is proportional to each other.

\[
Proportional_{dutyCycle} = kp \times \left( \frac{Voltage_{error}}{Voltage_{expected}} \right) \quad (eq \ 4-5)
\]

\[
Integral_{dutyCycle} = \sum ki \times \left( \frac{Voltage_{error}}{Voltage_{expected}} \right) \times Ts \quad (eq \ 4-6)
\]

\[
dutyCycle = 1 + \left( Proportional_{dutyCycle} + Integral_{dutyCycle} \right) \quad (eq \ 4-7)
\]

Thus using (eq 7) allows one to find the correct duty cycle based on the voltage/current percent error.
Figure 4-2: DC-DC Converter Schematic
Hardware Design

Figures 4-2 and 4-3 show the design's schematics and footprints. In the center of the schematic the MOSFETS, Inductors, and capacitors that make up the buck/boost converter circuit can be found. For these we chose high power components so as not to have them be the limiting factor in the design. They were rated to a max of 5A and 100v which is above the requirements specified in chapter 3. When testing the maximum voltages of the converter, the buck topology was able to handle the specified voltage, but the boost topology could not go above 40V. This is due to the measured transients on the 5v line. Essentially, setting the converter to boost above 40v would cause the 5v line to drop below the limits of the AT mega, which would restart the chip and cause
the software to crash. This is due in part to the amplitude and frequency of the current flowing through the inductor, since it is the largest and most noisy component on the board.

When testing the output voltage within the tolerances of the converter we found that it gave a critically damped response to both the buck and boost converter topologies. Depending on how big the jump was between the current and new set points, the transient would change proportionally, meaning that slowing down the converter would create a more underdamped response.

To maintain an accurate and constant steady state output the PI controller discussed above was implemented, but this requires the ability of the Atverter to sense the input and output voltage/current accurately and precisely. To ensure the accuracy of the sensors, extensive testing was down on them to better tune the sensors output values to be closer to the actual values. Shown in figures 4, 5, and 6 below are the results of the tuning process for the first Atverter and is how the other Atverter sensors were also tuned. Once the sensor value and actual value were graphed on the x and y axis for a suitable amount of data points, the equation could be implemented in the code so that the sensor value was more accurate to the real value.

Figure 4-4: Atverter 1 Low Side Voltage Sensor Tuning

\[ y = 1.0344x + 36.027 \]
Figure 4-4: DC-DC Converter Layout

Figure 4-5: DC-DC Converter Layout

y = 1.0211x - 92.411

y = 1.0471x + 23.672
Software Design

The above figure is the block diagram of the computer-side code. This runs on either a raspberry pi, or windows computer hosting a node-red server. Node-Red is a program that allows for easy setup and manipulation of network capable devices and runs using JavaScript code. Specifically, the code in the diagram continuously sends a "Request" command to the DC-DC converter over serial. The ATmega on the converter listens to the command and responds with the data it’s collecting high side voltage and current, low side voltage and current, and PWM duty cycle. The computer then parses that data and displays it on a graph for the end user to view. Moreover, this code implements numeric inputs and dropdowns so that the user can change the topology, and setpoints of the converter on the fly. Also, because this code relies on a serial port to communicate with the Atmega, multiple converters can be shown and manipulated simultaneously, the only limit being the number of serial ports the computer has. As a result, this satisfies the requirement of needing to have the converter be easy to use, since the user has access to a menu where they can select their desired implementation in a matter of minutes.
The ATmega328p communicates with the computer over serial and runs the control system for the DC-DC converter. The file it runs contains states for boost, buck, and current control as well as the necessary variables and libraries. As seen in figure 4-2 Digital pin 1 on the microcontroller sends a pulse-width-modulation signal to a gate driver which then switches the MOSFETS. Analog pins zero through three are dedicated to collecting sensor data for programming. The Atverter libraries read the ADC values from these analog pins and convert them into millivolt and milliamp values. Moreover, the Atverter initialization sets two timers, one to program the output of the duty cycle, and the other to determine the speed of the control loop. The former is controlled by the topology’s control loop and is initialized in the Atverter class’s constructor. As a result, little is needed to be done in the main program to set up the PWM signal other than to initialize the Atverter class itself. The second timer is initialized in a method contained in the Atverter class. Both the timer count and a function address are taken as arguments so that the microcontroller knows what to run and how fast. These two timers take up two-thirds of the timers on the ATmega328p and allow for code to run in parallel. The main loop of the program only calls the serial control function. Depending on if there is anything on the serial buffer, the ATmega will either send back sensor data, or change which control loop the topology interrupt calls, or changes the setpoint that the control loops are trying to achieve. These commands are set to be seven characters long and are easily readable so that they can be controlled either on a terminal, or through the abstracted Node-Red server.
Chapter 5. Testing and Results

Final Demonstration Procedure

The demonstration consists of attaching three “Atverter’s” to a laptop so that Boost, Buck, and Current mode can be run at the same time while the laptop reads desired data from the Atverter’s

1. Turn on 2 power supplies and set one to 5V and the other to 20V but make sure the power supplies are not “active”.
2. Attach the power supply cables to the Atverter’s depending on the topology and purpose (boost mode, buck, etc.).
3. For the boost Atverter attach the DC fan, buck Atverter attach lead acid battery, and current control Atverter attach the LED’s.
4. Now plug in the USB serial cables so that the Atverter’s are attached/connected to the laptop.
5. Go to “http://localhost:1880/” on your laptop and make sure to run node-red on your command center.
6. Deploy the flow for the node-red UI and set each Atverter to the proper mode depending on how the hardware was setup, example shown in figure 4 below.
7. Change the voltage or current values as necessary.

![Figure 5-1: Final Demonstration Block Diagram](image-url)
Figures 5-2 and 5-3 shown above are the buck and boost reactions to open load and load attached to the end of the Atverter. The reactions time it takes to get to steady state was about the same at 113ms, and since a steady state filter was used, the steady state error was never more than 500mv.
Figure 5-4 shown above is the UI for the server that runs the Atverter’s when attached to the laptop. As shown the user can set the state, and output value, while the server automatically takes the high/low side voltage, high/low size current and the PWM, to graph them in real time. The resolution of the UI is only 10hz, which provides a balance between not interrupting the Atverter’s control loop while also updating frequently enough for the end user to see the effects their inputs have on the hardware.
Chapter 6. Conclusion

Thus, the paper shows how the project was able to introduce and create an open-source dc-dc converter. From the ground up given the buck and boost converter, the students were able to create the necessary equations to calculate the correct duty cycle of the PWM. From there the students implemented a control system to monitor and adjust the output depending on the state of the input. An abstraction layer was also implemented so that the end user has an easy way to interact with the converter, and so that the topology can be switched easily.

Currently there are voltages that the equations do not quite match and that causes the controls system to oscillate the output around the specified voltage. To improve the design an ADC with more bits would be used so that the resolution of the controls signal would be more precise. Furthermore, higher quality voltage and current sensors could replace the current sensors as the ones on the board were not very accurate. This would allow the buck converter to match the user’s set output more closely. Lastly, the board could be redesigned to keep high voltage signals further away from low voltage signals which would lower transients on the 5v line and reduce noise on the data lines. Overall, the system specifications for this project were met, and it was pushed to a public repository on GitHub so that anyone else can pick up the project.
References


## Appendices

### Table 8-4: Block Diagram Table

<table>
<thead>
<tr>
<th>Index</th>
<th>Quantity</th>
<th>Part #</th>
<th>Manufacturer</th>
<th>Description</th>
<th>Unit Price</th>
<th>Extended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>IR2302PBF-ND</td>
<td>IR2302PBF</td>
<td>IC GATE DRVR HALF-BRIDGE</td>
<td>7.6</td>
<td>15.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8DIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>BAT46CT-ND</td>
<td>BAT46-TR</td>
<td>DIODE SCHOTTKY 100V 150MA</td>
<td>0.331</td>
<td>3.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DO35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>IRF540NPBF-ND</td>
<td>IRF540NPBF</td>
<td>MOSFET N-CH 100V 33A</td>
<td>1.35</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TO220AB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>296-TMCS1108A4BQDTCT-ND</td>
<td>TMCS1108A4BQDT</td>
<td>SENSOR CURRENT HALL 5.6A 8SOIC</td>
<td>2.83</td>
<td>11.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>M8843-ND</td>
<td>2319-H-RC</td>
<td>FIXED IND 390UH 5A 72 MOHM TH</td>
<td>4.66</td>
<td>9.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>493-15918-ND</td>
<td>URS2A331MHD</td>
<td>CAP ALUM 330UF 20% 100V RADIAL</td>
<td>1.78</td>
<td>7.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>PPC10.0KXCT-ND</td>
<td>SFR16S001002FR500</td>
<td>RES 10K OHM 1/2W 1% AXIAL</td>
<td>0.25</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ED2580-ND</td>
<td>OSTTA024163</td>
<td>TERM BLK 2P SIDE ENT 5.08MM PCB</td>
<td>0.76</td>
<td>3.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>2057-PH1-20-UA-ND</td>
<td>PH1-20-UA</td>
<td>CONN HEADER VERT 20POS 2.54MM</td>
<td>0.32</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1528-1785-ND</td>
<td>2222</td>
<td>GPIO FEMALE HEADER RASPBERRY PI</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>160-1939-ND</td>
<td>LTL1BEKVJNN</td>
<td>LED GREEN/RED DIFFUSED T/H</td>
<td>0.37</td>
<td>1.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>296-49723-ND</td>
<td>LM393PE3</td>
<td>IC COMPARATOR DIFF DUAL 8DIP</td>
<td>0.42</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>TC74HC02APF-ND</td>
<td>TC74HC02APF</td>
<td>IC GATE NOR 4CH 2-INP 14DIP</td>
<td>0.46</td>
<td>0.92</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>2151-AS-16.000-18-ND</td>
<td>AS-16.000-18</td>
<td>CRystal 16.0000MHZ 18PF TH</td>
<td>0.18</td>
<td>0.36</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>399-4220-ND</td>
<td>C317C220J2G5TA</td>
<td>CAP CER 22PF 200V NP0 RADIAL</td>
<td>0.19</td>
<td>0.76</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>2725-K78U05-500R3-ND</td>
<td>K78U05-500R3</td>
<td>DC/DC CONVERTER 5V</td>
<td>2.89</td>
<td>5.78</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>ATMEGA328P-PU-ND</td>
<td>ATMEGA328P-PU</td>
<td>IC MCU 8BIT 32KB FLASH 28DIP</td>
<td>3.11</td>
<td>6.22</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>2057-ICS-328-T-ND</td>
<td>ICS-328-T</td>
<td>IC SOCKET, DIP, 28P 2.54MM PITCH</td>
<td>0.27</td>
<td>0.54</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>4608X-2-101LF-ND</td>
<td>4608X-102-101LF</td>
<td>RES ARRAY 4 RES 100 OHM 8SIP</td>
<td>0.45</td>
<td>0.9</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>BC3456CT-ND</td>
<td>MBA02040C1203FC100</td>
<td>RES 120K OHM 0.4W 1% AXIAL</td>
<td>0.46</td>
<td>2.76</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>4606X-1-103LF-ND</td>
<td>4606X-101-103LF</td>
<td></td>
<td>0.44</td>
<td>1.76</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>1568-PRT-16763-ND</td>
<td>PRT-16763</td>
<td></td>
<td>2.25</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Table 8-5: Gantt Chart EE 460 Schedule
The Arduino took a little longer than originally thought as learning the digital control systems necessary to implement the code as well as turning the control system was found to be harder than originally expected and take much longer. This led the MPPT not being fully completed in time for the final demonstration but the building blocks for MPPT (Current control and Voltage control) were completed.

The Raspberry Pi during use was deemed too slow to support the entire UI so it was necessary to instead use the laptop to support the UI instead.
Table 8-4: Gantt Chart Interfacing

Since the Raspberry Pi was deemed too slow to use with the UI code, it was necessary to interface with the laptop instead which is why interfacing is almost completely done except for the fact it is done on the laptop instead of the Raspberry Pi.

Table 8-6: Gantt Chart Final Demo and Presentation
Analysis of Senior Project Design
Joshua Hutchinson
Jason Poon

Summary of Functional Requirements
The product is an Open-Source DC-DC Converter (OSDDC) with the capability to power manage multiple projects at the same time. Unlike conventional dc-dc converters that are proprietary and require specialist knowledge, and are usually designed for a single function, the proposed dc-dc converter will comprise of a programmable MCU and a Raspberry Pi (RPi) interface to allow less-skilled consumers to monitor and modify a power converting system. An open-source library will be developed that contains voltage control, current control, maximum power point tracking, and battery charge control profiles. Each library will be easy to implement through a GUI on the Raspberry Pi and will be controlled using an Atmega328 located on the power conversion unit. C++ and the Arduino IDE will be used for testing and will retain functionality in the finished project for more knowledgeable customers to edit the pre-set profiles. Moreover, the Pi will communicate with multiple converters and monitor their set points in applications where more than one dc-dc converter is necessary. The supporting hardware around the microcontrollers is a dc-dc converter, while the connection with the RPi and any external hardware will be open-source and custom designed to accommodate multiple converters on a single system. As a result, the integration of the dc-dc converter will provide a way to easily set up a microgrid system without the use of proprietary voltage converting hardware.

Primary Constraints:
As previously mentioned, the OSDDC has multiple inputs, a rechargeable battery, DC-DC Converter, as well as high- and low-level control. These are product requirements that will be challenging to implement as the whole system builds off each other to succeed. Specifically high- and low-level control systems limit signal/data transfer and increase power consumption. Whereas data bandwidth and software development rely on the high and low control hierarchy to split the worker properly. Outside of those some personal limitations were my knowledge of digital control systems. I had been exposed to control systems in the past but the digital side and implementation of said control systems were lacking. So, it definitely took some time research and figure out the proper equations for the digital PI controller.

Economic:
In terms of human capital, the OSDDC provides a range of jobs in manufacturing, hardware, and software development. Development of the product requires the team to design and test prototypes to manufacture a final product to sell. By far the development phase was the most expensive stage of the product life cycle, as the labor costs are estimated to be about 5000$.

In terms of financial capital, one of the product's requirements is affordability, as the project is designed for smaller companies and hobbyists. Thus, the uses prebuilt microcontrollers and self-made hardware to decrease costs and improve customizability. Component costs come out to about
38\$ per product with the ability to customize the product more which would add to the total cost a bit.

**If Manufactured on a Commercial Basis:**
- Estimated Manufacturing Cost per Product: 38\$
- Estimated Wholesale Price per Product: 38\$
- Estimated Retail Price per Product: 38\$
- Estimated Annual Cost for User to Operate Product: 0\$
- Estimated Annual Profit: 0\$

This project is designed to be open source, so the idea is to not make any money but more so provide a new idea free to the community. The consumer will still have to pay for the devices that are used but the code will be provided for free.

**Environmental:**
This device is a power manager so in terms of environmental impacts it is on the lower end as it tries to increase the efficiency of other products. That said currently most electrical energy is produced by nonrenewable means which contributes to the ever increasing of carbon dioxide in the environment in a negative way.

A critical component of the product is silicon-based transistors in the microcontrollers and other electrical hardware which may not be completely recyclable/renewable. Plastics, metals, and rubber insulation are all materials that will be used in the making of the device. If disposed of unethically, the waste from the product would lead to an environmental risk. To combat this, simple instructions could be given to ensure users know how to properly dispose of the product.

**Manufacturability:**
To create a final product, a Raspberry Pi with a PCB Buck-Boost Converter (for testing), with components will be added, assembled, and tested. Since the product is composed of standard electrical components no setbacks should occur, though with the current supply chain issues, some components, especially Raspberry Pi will be harder to come by for the average consumer. But as the main product is programmed onto the chip manufacturing results will not matter directly to the consumer but to the 3\textsuperscript{rd} party creators of the Raspberry Pi and AtMega328P chip.

**Sustainability:**
The overall product design, manufacturing, and project cycle take into account all aspects of sustainability: environmental, economic, and societal. The product is designed to be sustainable as it is open source to be cheaper for economic reasons; power manager which helps increase power efficiency of products and push forward renewable energy for the environmental impact; and being open source, it is accessible to a lot of people. No extra profit is gained by the designers, the device code is free to all, but the necessary hardware is the only part that needs to be purchased. In the long run this device will have a minimal impact on the environment and hopefully lead to more uses in renewable energy.

**Ethical:**
While the product is meant to be used to power manage smaller projects and increase favorability of renewable energy, since it is open source, it may also lead to many people buying the product and immediately discarding of it. This would end up creating more waste than necessary so to combat this dilemma instructions or a “README” on the code is necessary, so the user understand how to properly dispose of the Raspberry Pi and other hardware components they so choose to use/buy.

Considering the IEEE code of ethics #2, “the improvement of understanding individual and society of the capabilities and societal implications of conventional and emerging technologies” resonates the most with this product. The product is similar to introducing better ways people can power manage different designs they have and introduce people to more renewable methods. It is important to educate users on the use of power electronics in our current world and how it may help.

**Health and Safety:**
The device is able to handle up to 120W of power and while the device itself is low power and relatively safe the hardware that one may work with combined with this product can be dangerous. Power systems usually work with high voltage and current systems which always pose a danger to people if the proper safety is not considered. On our part we must ensure proper testing of the device is undertaken for electrical safety and instructions for proper assembly of the product.

The health and safety of the user is very important, it is every crucial to let the user understand the dangers this can pose, and we must ensure the instruction state how to properly use the device so no mistakes can occur.

**Social & Political:**
DC devices have always had the problem of efficiently converting and boosting voltage levels until recently. Even now proper devices are very constrained and thus hard to fit all types of projects out there, which is why our OSDDC is a necessary step in the right direction. This problem has led to much inefficiency in how our power is managed and thus the ever increasing problem with our reliance on fossil fuels.

One of our project goals is to put out a quality product, low in cost, and open to all who want to use it. No extra profit is gained by the designers, the device code is free to all, but the necessary hardware is the only part that needs to be purchased. In the long run this device will have a minimal impact on the environment and hopefully lead to more uses in renewable energy.

**Development**
Node-red was a new tool learned for this project as it is an easy way to develop and create an UI server ([Node-RED (nodered.org)](http://nodered.org)). Along with creating the server something else new I learned was the use of digital control systems. I had been exposed to control systems in past classes but was never taught how to implement them in the real world so a learning experience was understanding how control systems not only can be transitioned into the digital world but also how to implement and tune them for stability.
Project Title: Open-Source DC-DC converter

Student’s Name: Student’s Signature: Elijah Gordon

Advisor’s Name: Advisor’s Initials: Date: Jason Poon

• Summary of Functional Requirements
The product is an Open-Source DC-DC Converter (OSDDC) with the capability to power manage multiple projects at the same time. Unlike conventional dc-dc converters that are proprietary and require specialist knowledge, and are usually designed for a single function, the proposed dc-dc converter will comprise of a programmable MCU and a Raspberry Pi (RPi) interface to allow less-skilled consumers to monitor and modify a power converting system. An open-source library will be developed that contains voltage control, current control, maximum power point tracking, and battery charge control profiles. Each library will be easy to implement through a GUI on the Raspberry Pi and will be controlled using an Atmega328 located on the power conversion unit. C++ and the Arduino IDE will be used for testing and will retain functionality in the finished project for more knowledgeable customers to edit the pre-set profiles. Moreover, the Pi will communicate with multiple converters and monitor their set points in applications where more than one dc-dc converter is necessary. The supporting hardware around the microcontrollers is a dc-dc converter, while the connection with the RPi and any external hardware will be open-source and custom designed to accommodate multiple converters on a single system. As a result, the integration of the dc-dc converter will provide a way to easily set up a microgrid system without the use of proprietary voltage converting hardware.

• Primary Constraints
As previously mentioned, the OSDDC has multiple inputs, a rechargeable battery, DC-DC Converter, as well as high- and low-level control. These are product requirements that will be challenging to implement as the whole system builds off each other to succeed. Specifically high- and low-level control systems limit signal/data transfer and increase power consumption. Whereas data bandwidth and software development rely on the high and low control hierarchy to split the worker properly.

• Economic
In terms of human capital, the OSDDC provides a range of jobs in manufacturing, hardware, and software development. Development of the product requires the team to design and test prototypes to manufacture a final product to sell. By far the development phase was the most expensive stage of the product life cycle, as the labor costs are estimated to be about 5000$.

In terms of financial capital, one of the product’s requirements is affordability, as the project is designed for smaller companies and hobbyists. Thus, the use prebuilt microcontrollers and self-made hardware to decrease costs and improve customizability. Component costs come out to about 38$ per product with the ability to customize the product more which would add to the total cost a bit.
If Manufactured on a Commercial Basis:
- Estimated Manufacturing Cost per Product: 38$
- Estimated Wholesale Price per Product: 38$
- Estimated Retail Price per Product: 38$
- Estimated Annual Cost for User to Operate Product: 0$
- Estimated Annual Profit: 0$

This project is designed to be open source, so the idea is to not make any money but more so provide a new idea free to the community. The consumer will still have to pay for the devices that are used but the code will be provided for free.

Environmental:
This device is a power manager so in terms of environmental impacts it is on the lower end as it tries to increase the efficiency of other products. That said currently most electrical energy is produced by nonrenewable means which contributes to the ever increasing of carbon dioxide in the environment in a negative way.

A critical component of the product is silicon-based transistors in the microcontrollers and other electrical hardware which may not be completely recyclable/renewable. Plastics, metals, and rubber insulation are all materials that will be used in the making of the device. If disposed of unethically, the waste from the product would lead to an environmental risk. To combat this, simple instructions could be given to ensure users know how to properly dispose of the product.

Manufacturability:
To create a final product, a Raspberry Pi with a PCB Buck-Boost Converter (for testing), with components will be added, assembled, and tested. Since the product is composed of standard electrical components no setbacks should occur, though with the current supply chain issues, some components, especially Raspberry Pi will be harder to come by for the average consumer. But as the main product is programmed onto the chip manufacturing results will not matter directly to the consumer but to the 3rd party creators of the Raspberry Pi and AtMega328P chip.

Sustainability:
The overall product design, manufacturing, and project cycle consider all aspects of sustainability: environmental, economic, and societal. The product is designed to be sustainable as it is open source to be cheaper for economic reasons; power manager which helps increase power efficiency of products and push forward renewable energy for the environmental impact; and being open source, it is accessible to a lot of people. No extra profit is gained by the designers, the device code is free to all, but the necessary hardware is the only part that needs to be purchased. In the long run this device will have a minimal impact on the environment and hopefully lead to more uses in renewable energy.

Ethical:
While the product is meant to be used to power manage smaller projects and increase favorability of renewable energy, since it is open source, it may also lead to many people buying the product and immediately discarding of it. This would end up creating more waste than necessary so to
combat this dilemma instructions or a “README” on the code is necessary, so the user understand how to properly dispose of the Raspberry Pi and other hardware components they so choose to use/buy.

Considering the IEEE code of ethics #2, “the improvement of understanding individual and society of the capabilities and societal implications of conventional and emerging technologies” resonates the most with this product. The product is introducing better ways people can power manage different designs they have and introduce people to more renewable methods. It is important to educate users on the use of power electronics in our current world and how it may help.

**Health and Safety:**
The device can handle up to 120W of power and while the device itself is low power and relatively safe the hardware that one may work with combined with this product can be dangerous. Power systems usually work with high voltage and current systems which always pose a danger to people if the proper safety is not considered. On our part we must ensure proper testing of the device is undertaken for electrical safety and instructions for proper assembly of the product.

The health and safety of the user is very important, it is very crucial to let the user understand the dangers this can pose, and we must ensure the instruction state how to properly use the device so no mistakes can occur.

**Social & Political:**
DC devices have always had the problem of efficiently converting and boosting voltage levels until recently. Even now proper devices are very constrained and thus hard to fit all types of projects out there, which is why our OSDDC is a necessary step in the right direction. This problem has led to much inefficiency in how our power is managed and thus the ever-increasing problem with our reliance on fossil fuels.

One of our project goals is to put out a quality product, low in cost, and open to all who want to use it. No extra profit is gained by the designers, the device code is free to all, but the necessary hardware is the only part that needs to be purchased. In the long run this device will have a minimal impact on the environment and hopefully lead to more uses in renewable energy.

**Development:**
To develop the design for our OSDCC, the team researched on pass types of DC-DC converters. The basic components used in these devices and similar devices that exist for this project. Basic electrical components, and a user interface created properly collect data from the Raspberry Pi using SPI so that the ATmega328 and the Raspberry can communicate with each other. Proper coding practices will be implemented, and electronic design skills obtained through courses at Cal Poly.