COMMITTEE MEMBERSHIP

TITLE: Smart Wall Outlet Design and Implementation for the DC House Project

AUTHOR: Kevin Roy Mendoza

DATE SUBMITTED: June 2014

COMMITTEE CHAIR: Dr. Taufik, Professor of Electrical Engineering

COMMITTEE MEMBER: Dr. Ali Shaban, Professor of Electrical Engineering

COMMITTEE MEMBER: Dr. James Harris, Professor of Electrical Engineering
ABSTRACT

Smart Wall Outlet Design and Implementation for the DC House Project

Kevin Roy Mendoza

Most everyday AC appliances are designed to operate off of 120V coming from the wall outlet in our homes. This voltage is a standard set from our established infrastructure. Unlike AC devices, DC devices do not have any set standard of voltage they all will run off of. This presents a problem for the DC house as the various loads that will be used will have different required input voltages. One set voltage for a wall outlet will not suffice for the DC House. This Smart Wall Outlet is designed with a DC-DC converter that will have its output voltage controlled by an on-board microprocessor. The Smart Wall Outlet detects current going into a device, and will adjust the voltage applied to the device to ensure it operates most efficiently. Proof of concept research has already been performed in the past, and this thesis will look towards implementing this concept on a single circuit board.
ACKNOWLEDGMENTS

I would like to personally thank my thesis advisor, Taufik, for inspiring a passion for power electronics in me, and giving me support throughout this thesis. I would also like to thank my friends and family for believing in me, and supporting me throughout my college career. I would like to thank the folks at Rantec Power Systems Inc. for the guidance they offered me throughout this project. Lastly, I would like to thank my parents, Albert and Rose Mendoza, for their loving support and instilling in me the value of hard work and determination. Without them, none of this would have been possible.
# Table of Contents

List of Tables ............................................................................................. vii
List of Figures .......................................................................................... viii

Chapter 1: Introduction .............................................................................. 1
  1.1 The DC House .................................................................................. 1
  1.2 Why DC? ......................................................................................... 2
  1.3 Previous design Phases of the DC House ......................................... 3

Chapter 2: The Wall Outlet for the DC House ........................................ 5
  2.1 The Need for a Variable Voltage Wall Outlet .................................. 5
  2.2 Previous designs ............................................................................. 6

Chapter 3: Design Specifications ............................................................. 9
  3.1 General Design ............................................................................... 9
  3.2 Output Voltage, Current, and Power Requirements ....................... 10

Chapter 4: Design and Component Selection ........................................ 12
  4.1 Main Converter Switching Controller Selection ......................... 12
  4.2 Bias Converter Switching Controller Selection ........................... 12
  4.3 Digital Potentiometer Selection ..................................................... 13
  4.4 Microprocessor Selection ............................................................... 15
  4.5 Design and Component Selection for the Main Converter .......... 16
  4.6 Overall Power Supply Design ......................................................... 17
  4.7 Software Design ............................................................................ 23

Chapter 5: Experimental Results ............................................................ 25
  5.1 Testing Approach ........................................................................... 25
  5.2 Initial Testing and Troubleshooting ................................................. 26
  5.3 Initial Software Development and Testing with Loads .................. 33
  5.4 Voltage Setting Algorithm Development ....................................... 39
  5.5 Efficiency Measurements ............................................................... 41

Chapter 6: Conclusions and Recommendations .................................... 43

BIBLIOGRAPHY ....................................................................................... 46

Appendix A: Microprocessor Code ............................................................ 48
List of Tables

Table 3-1: Summary of Test Loads .......................................................... 10
Table 3-2: Summary of Design Parameters for the Smart Wall Outlet ..................... 11
Table 4-1: AD5160 Wiper Programming Specifications [9] ........................................... 14
Table 4-2: Connections Made to the LTC3891 from the Other Main Components .................. 17
Table 4-3: Connections Made to the ATmega164P from Other Main Components ............. 18
Table 4-4: LED Flashing Patterns ................................................................. 24
Table 5-1: Experimental Values of Wiper Register Settings and the Resulting Output Voltages . 31
Table 5-2: Input and Output Measurements and Efficiency Data ...................................... 42
List of Figures

Figure 1-1: Block Diagram of the Power Generation/Conversion for the DC House .................. 1
Figure 2-1: Block Diagram of the Connection of the Wall Outlet ........................................ 5
Figure 2-2: Wall Plug Concept for the First Design of the Variable Voltage Wall Outlet [4] ........ 6
Figure 2-3: Block diagram of the first Smart Wall Outlet [6] .................................................... 7
Figure 3-1: Block Diagram of the Smart Wall Outlet Implementation ..................................... 9
Figure 3-2: Test Loads for the Smart Wall Outlet ................................................................. 11
Figure 4-1: Functional Block Diagram for the AD5160 [9] ....................................................... 14
Figure 4-2: ATmega164P Pinout and Port Descriptions [10] .................................................... 15
Figure 4-3: Block Diagram of the INA283 Current Sense Amplifier [11] ............................... 19
Figure 4-4: Smart Wall Outlet Layout .................................................................................... 21
Figure 4-5: Initial Schematic for the Smart Wall Outlet Design .............................................. 22
Figure 4-6: State Diagram for the Smart Wall Outlet .............................................................. 23
Figure 5-1: Bare PCB for the Smart Wall Outlet ..................................................................... 25
Figure 5-2: Final PCB of the Smart Wall Outlet ................................................................. 28
Figure 5-3: Revised Schematic for the Smart Wall Outlet ....................................................... 29
Figure 5-4: Converter V_OUT as a Function of the Digital Potentiometer 8-bit Wiper Setting ...... 32
Figure 5-5: Terminal B to Wiper Resistance as a Function of the Wiper Register Setting ......... 32
Figure 5-6: Input Characteristics for the 6V TV [6] ................................................................. 33
Figure 5-7: Test Setup for the Software Development ............................................................. 35
Figure 5-8: Input Current for the 3V Radio at Full Volume ..................................................... 36
Figure 5-9: Input Characteristics for the 12V Fan Array .......................................................... 37
Figure 5-10: State Diagram for the Voltage Setting Algorithm ............................................ 41
Chapter 1: Introduction

1.1 The DC House

The DC House, much like Cal Poly’s Sustainable Power for Electrical Resources (SuPER) project [16], is a humanitarian effort to provide electrical power to people in third world countries who do not have the established infrastructure to power electrical appliances [1]. The DC House runs off of renewable energy sources such as wind, photovoltaic, hydro-electric, and even human-powered generation. These abundant renewable energy sources will allow for plenty of energy generation for the loads inside of the DC House. These generation sources will provide energy to batteries, and will ultimately power common household appliances and lighting. Figure 1-1 shows a simplified block diagram of the DC House's power generation/distribution stage [2].

![Figure 1-1: Block Diagram of the Power Generation/Conversion for the DC House](image-url)
1.2 Why DC?

Here in America, all of our houses run off of AC power. We have a robust power grid, including massive power plants, substations, and transformers. This entire infrastructure has been developed over the years since we had the means to support its construction. However, in a poor village people typically do not have access or the means to build power plants and transmission lines. They simply do not have the infrastructure. A DC power system when used locally simply does not need this large infrastructure in order to operate. A simplified DC power system could need nothing more than a solar panel, charge controller, and a battery.

Despite our infrastructure being centered on AC, most of the devices we plug into our walls are DC devices. These devices require an AC-DC (rectification) conversion stage, but if the house ran off of DC this stage wouldn’t be necessary. The more stages of power conversion that is necessary, the less efficient the overall system will be. Another example of unnecessary power conversion is the way solar panels are used in America. Solar panels generate energy as a DC device, and they must be converted to AC which will again need to go back to DC if the device requires rectification. On an AC system, tapping into solar power, one of the most abundant renewable energy sources, requires two or three conversion stages, but if the house ran on DC it could be done with just one conversion stage [12].

A main advantage DC has over AC is the fact that batteries are DC devices. Batteries can be considered a bidirectional device because they can both supply, and sink energy from the system. If there is enough energy on the DC bus then the battery will sink energy, and charge itself, and if there is a lack of energy on the bus, then the battery will supply energy to the bus. This is all automatically controlled by the physics behind
energy flow, and will require no automation [13].

The simple fact that batteries can store energy is a unique advantage DC has over AC. Consider the automotive market. DC has been selected to power the electronics in the car because batteries can store energy inside of them while the engine is running, and this allows the driver to access this electricity when the car is not running. In an AC power system you cannot directly store energy in this way, and this would mean you would need constant generation in order to do something as simple as turn on a light. Since the DC House is designed to have PV, wind, and hydro-electric, and if one or all of these were not operating for some reason (no wind or night time) the village will still have power to draw from the battery.

DC also can replace large transformers, the purpose of which is to step up and down the voltage for distribution, with a DC-DC converter. A DC-DC converter is typically much more efficient than a transformer, and this is important because efficiency is a main goal for the DC House project.

1.3 Previous design Phases of the DC House

The DC House project is currently in its Phase 4 stage of work. Phase 1 began on the DC House in academic year 2010-2011 with the house modeling and system design [2; 3]. During this first phase of the project, major specifications of the DC House were considered like the line voltage, efficiency, maximum load size, safety, and the economics associated with the DC House. The different renewable energy sources, as well as the multiple input single output (MISO) DC-DC converter were also being addressed and designed during this first phase of the DC House project. As Phase 2 (AY
2011-2012) began there were improvements on the various sub-projects that were completed in Phase 1. The DC House had its MISO DC-DC converter improved upon, as well as the addition of a 48V light bulb, a 12V wall outlet, and improvements on the human-powered electricity generation. Phase 3 (AY 2012-2013) continued on with more design of the generation sources such as a swing, a merry-go-round, as well as simulations for hydro-electric and wind. Phase 4 (AY 2013-2014) is mostly improvements on previous projects such as the human powered generation, portable DC light bulb, the MISO DC-DC converter, and a bidirectional flyback DC-DC converter. All of these various projects go into making a fully operational DC House.
Chapter 2: The Wall Outlet for the DC House

2.1 The Need for a Variable Voltage Wall Outlet

The DC House has the unique task of supplying power to loads at variable DC input voltages. Any device that runs off of a battery is a DC device, and in everyday life it can be seen that some of them use two AA batteries (3V), 4 AA batteries (6V) or maybe one of the 9V batteries. The fact is that DC devices do not need to follow a standard in contrast to AC because there is no standard for DC power distribution. Figure 2-1 illustrates where the wall outlet will be integrated into the DC House design with three example loads that one might want to plug in.

![Diagram showing the connection of the wall outlet](image)

Figure 2-1: Block Diagram of the Connection of the Wall Outlet

Multiple wall plugs at different output voltages can be implemented, but that would require more converters which would in the end mean more maintenance and more cost. This would also require the user in a third world country to not only know the device's required voltage, but also understand the ramifications of plugging a 3V device into a 24V outlet for example. The need for a variable voltage wall outlet has been recognized since Phase 1 of the DC House design, and there have been previous designs as well as
improvements upon those designs since then. There has also been research looking into variable DC bus voltages in order to maximize efficiency for DC motors which will be another reason to have a variable DC-DC converter [14].

2.2 Previous designs

The first design in Phase 1 aimed to get the design of an adjustable output voltage converter. The design was not intended for the converter to automatically adjust the output voltage whenever you loaded it. For this first iteration, the user had to manually select the output voltage they desired before plugging the device in [4]. While the adjustable DC-DC converter works, the user would still need to know what output voltage each load requires and this can lead to a potentially harmful (to the person or the system) mistake. Figure 2-2 shows how the outlet looked, it has an external resistor $R_{select}$ that needs to be properly adjusted for the output voltage to be suitable for the user's needs [4].

![Figure 2-2: Wall Plug Concept for the First Design of the Variable Voltage Wall Outlet][4]
During Phase 2 a 12V non-adjustable wall outlet was designed and built. The 12V design still has its uses since there are a large number of 12V that go into cars, or RVs. While this converter was successfully implemented, it was obviously not an end solution as it could not change its output voltage [5]. During Phase 2, there was also the first research done on an automatic smart wall outlet. This proof of concept research and testing was done using a microcontroller and a digital potentiometer in order to have the converter automatically adjust the output voltage depending on the load that is connected [6]. Figure 2-3 shows a block diagram of the first prototype of the Smart Wall Outlet.

![Block diagram of the first Smart Wall Outlet](image)

**Figure 2-3: Block diagram of the first Smart Wall Outlet [6]**
This research showed promising results by controlling the output voltage based off of the output current that was read. One of the shortcomings of this design was that it would only be able to detect the minimum input voltage for the test loads that were used. This is not ideal because operating a device at a minimum input voltage will result in higher input current, and will cause the device to run inefficiently. Another shortcoming of this design was that it was not a single power supply. The design consisted of an Arduino test board, a breadboard, and a demo board for the LM5117. This is not something that can be implemented into the design of the DC House because it is three different boards strung together with wires, and it will not fit inside of a wall mounted outlet box very easily.

The goal of this thesis is to improve upon the existing design of the Smart Wall Outlet by making a more compact design. This thesis will combine all three of these boards into one compact PCB that can be put into the DC House easily. This new design will keep the user in mind by having an LED that will signal the state of the Smart Wall Outlet, and will have an overvoltage protection feature. Another goal for this thesis is to improve upon the algorithm in order to ensure the devices operate at each device’s optimal input voltage.
Chapter 3: Design Specifications

3.1 General Design

Since the DC House is running off of a 48V bus voltage, the converter that will be used for this application will need to be a step-down DC-DC converter as there are not very many DC appliances that run above 48V. Since isolation is not a requirement for this design a buck converter was chosen for its low cost and simplicity. There will need to be some way to sense the current, as well as a way to process this information. A current sensing resistor will need to be used at the output of the converter, and then will feed into a current sense amplifier that can give information to a microprocessor. The microprocessor from this point can take this information and will need a way to adjust the output voltage of the converter. Prior research had shown that when using a digital potentiometer as part of the feedback divider for a buck this feature can be achieved [6]. These digital devices cannot run off of 48V, so there will have to be another way for them to get power. This means there will have to be another converter on board to act as a 5V bias supply. A simplified block diagram of the improved Smart Wall Outlet is shown in Figure 3-1 below.

![Block Diagram of the Smart Wall Outlet Implementation](image)

Figure 3-1: Block Diagram of the Smart Wall Outlet Implementation
The Smart Wall Outlet will be an all in one board, unlike the previous design. The processor, main DC-DC converter and the digital potentiometer will all be on the same board. The Smart Wall Outlet will have an LED on it as well that will indicate certain conditions like that the processor has shut down the main converter due to a fault, or that the processor has stopped adjusting the output voltage. The dimensions of this final PCB will be 3.93”x 3.14”, which will be much more compact than the previous design. The final Smart Wall Outlet will have both banana jacks as well as turrets for both input and output. Ultimately the Smart Wall Outlet is to be mounted on a wall of the DC house and will make use of a regular AC wall outlet for the simple reason that they are readily available.

3.2 Output Voltage, Current, and Power Requirements

Most converters are designed with a particular fixed output voltage and maximum output current. Since this converter is designed to have a variable output voltage, the approach taken was to design it for three devices that were obtained for testing. Table 3-1 shows the three devices that will be used for testing, and their input parameters.

<table>
<thead>
<tr>
<th>Device</th>
<th>Vin Minimum</th>
<th>Voltage Rating</th>
<th>Power Rating</th>
<th>Current Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>3V Radio</td>
<td>1.5V</td>
<td>3V</td>
<td>.066W</td>
<td>.022A</td>
</tr>
<tr>
<td>6V TV</td>
<td>2.5V</td>
<td>6V</td>
<td>3.2W</td>
<td>.533A</td>
</tr>
<tr>
<td>12V Fan Array</td>
<td>5V</td>
<td>12V</td>
<td>3.2W</td>
<td>.268A</td>
</tr>
</tbody>
</table>
The converter will need to be designed to handle a maximum load of 48W as shown in equation 3-1 below.

\[ \text{Power (Watts)} = \text{Current (Amps)} \times \text{Voltage (Volts)} \]  

(3-1)

In summary, the converter will be designed for the specifications listed in Table 3-2 below.

<table>
<thead>
<tr>
<th>Table 3-2: Summary of Design Parameters for the Smart Wall Outlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage</td>
</tr>
<tr>
<td>Output Voltage</td>
</tr>
<tr>
<td>Output Current</td>
</tr>
<tr>
<td>Maximum Output Power</td>
</tr>
</tbody>
</table>
Chapter 4: Design and Component Selection

4.1 Main Converter Switching Controller Selection

Since the Smart Wall Outlet requires the output to be lower than the input of 48V, a step down converter is required. A buck converter was selected in order to accomplish this task. The buck converter would need to be able to handle a relatively high voltage as well. For improved efficiency, a buck converter with synchronous rectification was sought after. After considering a few choices the LTC 3891 from Linear Technologies was selected. This chip is a low I_Q, 60V synchronous step-down controller. It was chosen because of its slightly better efficiency than the other controllers. The low quiescent current will also help it be more efficient as the Smart Wall Outlet will probably spend some time idling, as there might not always be a load connected to it. The LTC 3891 also has current mode control, which will be useful as a hardware over-current protection that will complement a software over-current protection implemented on the processor [7].

4.2 Bias Converter Switching Controller Selection

The Bias Converter is needed for a few functions. First, it is needed to supply 5V for the microprocessor, digital potentiometer, current sense amplifier, as well as an optional VCC supply for the LTC 3891. This V_{CC} is optional because the chip will generate its own V_{CC} from the 48V input, but this would be horribly inefficient as it uses a linear regulator internal to the chip. A linear regulator's efficiency is shown by Equation 4-1.

\[ \eta_{LR}(\%) = \frac{V_{out}}{V_{in}} \]  

(4-1)
With a high input voltage of 48V, and a low output voltage of 5.1V (internal to the chip), this would cause the LTC 3891 to have a very inefficiently generated bias voltage. That is why the 5V will be supplied by the bias converter.

Between these various loads for the bias converter, an estimated 100-200mA will be required from it. Another requirement that the bias converter shares with the main converter is the ability to operate at 48V\textsubscript{in}. The LM2597HVM was selected in order to fill the needs for this bias converter as it can run up to 60V\textsubscript{in} and can supply up to 500mA. This controller has an internal switch, and requires minimal external circuitry. This is useful because it will take up less board space, and will be simpler to implement [8].

4.3 Digital Potentiometer Selection

After some research there were a couple different types of digital potentiometers in regards to how the wiper is set. The choices came down to serial peripheral interface (SPI) or Inter-Integrated Circuit (I\textsuperscript{2}C) based digital potentiometers. A SPI based digital potentiometer was chosen because SPI is generally less complex, and for this task, the advantages I\textsuperscript{2}C has over SPI are not needed. The AD5160 256 position digital potentiometer was selected. The functional block diagram for this digital potentiometer is shown below in Figure 4-1.
The AD5160 is a 256 position potentiometer, meaning the wiper register is an 8-bit register which can range from 0-255. When the wiper register is loaded with all of its bits to be 1 (0xFF) then the wiper will go up to pin A, and when the register is loaded to all 0's (0x00), the wiper will go all the way down to pin B. The wiper settings can be seen in Table 4-1 below showing an example for the 10K\(\Omega\) digital potentiometer.

### Table 4-1: AD5160 Wiper Programming Specifications [9]

<table>
<thead>
<tr>
<th>(D) (Dec.)</th>
<th>(R_{WB}) ((\Omega))</th>
<th>Output State</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>9961</td>
<td>Full Scale ((R_{AB} - 1 \text{ LSB} + R_{W}))</td>
</tr>
<tr>
<td>128</td>
<td>5060</td>
<td>Midscale</td>
</tr>
<tr>
<td>1</td>
<td>99</td>
<td>1 \text{ LSB}</td>
</tr>
<tr>
<td>0</td>
<td>60</td>
<td>Zero Scale (Wiper Contact Resistance)</td>
</tr>
</tbody>
</table>
4.4 Microprocessor Selection

The microprocessor is obviously an important part to the Smart Wall Outlet. It is required for important tasks such as sensing output current, sensing output voltage, enabling/disabling the main converter, and providing the ability to set the digital potentiometer. An additional feature was the ability to be programmed via a Joint Test Action Group (JTAG) device. This will allow the processor to be soldered to the printed circuit board (PCB), and still have the ability to program the processor in-circuit.

The ATmega164P is a processor that can handle all of these tasks without being oversized and therefore being overly expensive. The ATmega164P is an 8-bit 16KB Flash with on-board Analog to Digital converters (ADC), 32 Input/Output pins, is SPI compatible, and can be programmed via JTAG. Figure 4-2 depicts a pinout of the ATmega164P and a general view of the different ports, and what these ports will be used for [10].

![Figure 4-2: ATmega164P Pinout and Port Descriptions [10]](image-url)
4.5 Design and Component Selection for the Main Converter

The LTC3891 has application notes and demo boards that were used as design aids for sizing many of the smaller components, but there were still some decisions to be made about a few components. The primary switching MOSFET, as well as the synchronous MOSFET needed to be sized for the harshest operating conditions of 48Vin at 12Vout and 4A. The necessary equations for this are shown below in Equations 4-2 through 4-5.

Voltage Rating $> V_{in}$ \hspace{1cm} (4-2)

Duty Cycle for the primary MOSFET (in CCM) $D = \frac{V_{out}}{V_{in}}$ \hspace{1cm} (4-3)

Current Rating for Primary MOSFET $> I_{out} \times D$ \hspace{1cm} (4-4)

Current Rating for Synchronous MOSFET $> I_{out} \times (1-D)$ \hspace{1cm} (4-5)

The equations above result in the demand for MOSFETS with a voltage rating of greater than 48V, and current ratings greater than 1A for the primary MOSFET, and greater than 3A for the synchronous MOSFET.

The next component that needed to be sized was the inductor. The sizing of the inductor is critical to ensure a small output voltage/current ripple. Equation 4-6 below is from the application notes on the LTC3891 Datasheet which refers to sizing the inductor for ripple current [7].

$$\Delta I_L = \frac{1}{(f)(L)} V_{OUT} (1 - \frac{V_{OUT}}{V_{IN}})$$ \hspace{1cm} (4-6)

$$\Delta I_L = \frac{1}{(200kHz)(68\mu H)} 3V \left(1 - \frac{3V}{48V}\right) = .206A$$
To avoid having a large ripple current, either a higher switching frequency or a larger inductance value will be needed. Because a higher switching frequency would cause more switching losses, a larger inductance value was decided on. The inductance value of 68µH calculated from Equation 4-6, is the largest inductance readily available that could handle 4A without saturating the core [7].

4.6 Overall Power Supply Design

Figure 3-1 in the previous chapter shows a high level block diagram, with four main blocks: the main converter, the 5V bias converter, the microprocessor, and the digital potentiometer. These four blocks need to all have crucial connections. Table 4-2 and Table 4-3 below show these connections.

| Table 4-2: Connections Made to the LTC3891 from the Other Main Components |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| Connections to the LTC3891  | Connections Made to the LTC3891 |
| LTC3891 | ATmega164 | AD5160 | LM2597HVM | Purpose |
| Pin # | Pin Name | Pin # | Pin Name | Pin # | Pin Name | Bias Converter | Purpose |
| 6 | Run | 18 | PortD4 | - | - | - | Allows the Processor to have control of enabling or disabling the Main Converter |
| 9 | VFB | - | - | 1 | W (Wiper) | - | Makes feedback connection to the Digital Potentiometer |
| 17 | EXTVCC | - | - | - | - | 5V output | Makes the LTC3891 more efficient by avoiding the internal generation of Vcc |
Table 4-3: Connections Made to the ATmega164P from Other Main Components

<table>
<thead>
<tr>
<th>Connections to the ATmega164P</th>
<th>ATmega164</th>
<th>AD5160</th>
<th>LM2597HVM</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin #</td>
<td>Pin Name</td>
<td>Pin #</td>
<td>Pin Name</td>
<td>Bias Converter</td>
</tr>
<tr>
<td>5</td>
<td>SS_PortB4</td>
<td>6</td>
<td>/CS</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>MOSI_PortB5</td>
<td>5</td>
<td>SDI</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>SCK_PortB7</td>
<td>4</td>
<td>CLK</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>VCC</td>
<td>-</td>
<td>-</td>
<td>5V Output</td>
</tr>
</tbody>
</table>

These connections are crucial to ensure the whole power supply works together as a system. Another decision that had to be made was how to read the output current and transfer this information to the processor. The INA283 is a current sense amplifier with a gain of 200V/V [11]. It can read the differential voltage across a sense resistor, and translate this into an analog voltage which will then be fed to an input to an ADC on the ATmega164P. A block diagram for this device is shown in Figure 4-3 below.
Selection of the sense resistor is shown in Equation 4-7. A 4.096V reference is used with the ATmega164P at pin 32 AREF. Since the ADC on the processor is a 10-bit ADC, this means the top end digital value it can read is 1023, which results from an analog value of 4.096. In order for the ADC to have resolution and accurately measure the current reading, the maximum anticipated amount of current should result in the INA283 reporting an analog value of somewhere between 3-4V. Since the highest test load being used on this board is a .533A device. The top end current reading was set around .8A, but the current rating of the sensing resistor will be chosen for the 4A specification.
\[ V_{out} = I \times R_{sense} \times G_{INA283} + \frac{V_{Ref}}{2} \quad (4-7) \]

Gain is 200V/V and \( V_{Ref} \) for the INA283 is 2.048V

\[ V_{out} = 0.768 \times 20\text{m}\Omega \times 200 + \frac{2.048}{2} = 4.096V \]

A 20mΩ sense resistor was chosen, but the power rating must also be considered.

Equation 4-8 shows how the power rating for this resistor was selected.

\[ P = I^2 \times R \quad (4-8) \]

\[ P = 4^2 \times 20\text{m}\Omega = 320\text{mW} \]

Since the converter is designed for 4A the power rating of \( R_{sense} \) will have to be at least 320mW. Therefore, a 500mW rated resistor was selected for \( R_{sense} \).
For the schematic and layout design of the overall power supply, CADSOFT's Easily Applicable Graphical Layout Editor (EAGLE) was used as the primary tool. EAGLE was used to make the schematic and pick the parts based on their package sizes to ensure they would all fit when laid out on a custom two layer PCB. The layout for the Smart Wall Outlet is shown in Figure 4-4. The red traces are the top layer and the blue traces are the bottom layer. The schematic for the Smart Wall Outlet is shown in Figure 4-5.

Figure 4-4: Smart Wall Outlet Layout
Figure 4-5: Initial Schematic for the Smart Wall Outlet Design
4.7 Software Design

The algorithm from the first Smart Wall Outlet will be applied and improved upon if possible. The algorithm relied on sensing a sudden increase (\(\Delta I_{\text{out}}\)) in output current to determine the device was running. When the Smart Wall Outlet is first applied power, the digital potentiometer will be set for the output voltage around 1.5V before the converter is even enabled, and will start to increase from the detection of current. The processor will keep increasing the output voltage of the converter until a large \(\Delta I_{\text{out}}\) is detected, and then will try to fine tune the output voltage. In the event the load is unplugged, the converter will reinitialize itself. In the event the voltage keeps rising uncontrollably, the processor will be monitoring the output voltage and will shut down the converter and go into a fault state. The software for this new converter will have a few basic states as shown in Figure 4-6.

Figure 4-6: State Diagram for the Smart Wall Outlet
The software for this new Smart Wall Outlet will also flash the on board LED in a few distinct patterns in order to determine visually what state the converter is in. The LED will flash quickly to signal that the converter has stopped adjusting its output voltage, it will flash the LED slowly to signal it is standing by for a load, it will stay lit to signal the converter is adjusting the output voltage, and it will flash the LED in quick 3 flash burst in order to signal either a short circuit or an overvoltage has been detected. Table 4-4 summarizes the LED patterns in order for a user to determine what state the converter is in.

**Table 4-4: LED Flashing Patterns**

<table>
<thead>
<tr>
<th>Condition of Smart Wall Outlet</th>
<th>LED Flashing Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing by and unloaded</td>
<td>Periodically and slowly flashing</td>
</tr>
<tr>
<td>Loaded and Setting $V_{out}$</td>
<td>Always On</td>
</tr>
<tr>
<td>Loaded and $V_{out}$ is set</td>
<td>Periodically and quickly flashing</td>
</tr>
<tr>
<td>Fault Condition</td>
<td>Quick 3 Flash Burst</td>
</tr>
</tbody>
</table>
Chapter 5: Experimental Results

5.1 Testing Approach

Figure 5-1 shows the bare PCB which was then populated by the selected components. The components for the 5V bias supply were first soldered to the board, as the processor cannot run without it, and the main converter depends upon its supply voltage as well. Verifying this 5V bias works properly was the first step taken in testing the board. The next step was getting the components for the main converter soldered to the board, and verifying its operation using a static resistor instead of the digital potentiometer. Lastly, the digital control circuitry including the processor, digital potentiometer, and current sensing amplifier will be soldered to the board and tested. Once these components are up and running, the testing and tweaking of the control program can begin.

Figure 5-1: Bare PCB for the Smart Wall Outlet
5.2 Initial Testing and Troubleshooting

There were a few errors made when laying out and designing the PCB that needed to be corrected and resulted in a revised schematic. As stated in the previous section the construction and testing of the Smart Wall Outlet began with the 5V bias supply. Once the LM2597HVM and its components necessary for its operation were soldered to the board, input voltage was applied and the output of the bias converter was measured. The bias voltage was not the desired 5V, but rather was following the input voltage. The cause of this turned out to be a crucial missing trace connecting the 5V output to the feedback inside of the LM2597HVM. A jumper wire had to be placed in the circuit in order to make this connection.

Once the 5V converter was fixed, the testing moved towards getting the main converter running. The main converter was set for a 12V output using the two static resistors to make the feedback divider. When the main converter was turned on for the first time, it was apparent there was an instability because the output voltage was not steady, and there was an audible high pitched ringing. This problem was remedied by adding more capacitance on the output, but there was still a problem when current was drawn. After investigation, another layout/design error was made in that the current sense leads on the LTC3891 were implemented in reverse. Traces were cut and the leads were swapped in order to remedy this problem.

Once the main converter was running satisfactorily, the focus switched over to the digital control circuitry, including the processor, digital potentiometer, and the current sense amplifier. One of the main problems discovered with this circuitry was that the 10 pin header used, was not the standard .1” spacing header but rather a 2mm spacing header. Besides this fact pull-up resistor R12 was misplaced and should have been
placed to pull up the TDI pin of the header. These problems had to be made before communication to the processor could begin. A .1” header was soldered to the board with wires running from pin to pin, as the solder pads were too small to accept the larger header, and R12 was removed, and a new pull-up resistor was implemented for the TDI pin.

Once communication is established to the processor, the digital potentiometer could be set ultimately giving control over the output voltage of the converter. A problem soon became apparent however, since the digital potentiometer was used as the top resistor in the divider, this means that the voltage on the device can be up to 12V. According to the datasheet, the maximum voltage on any pin can be 7V [9]. This would mean that the digital potentiometer would have to be switched to the bottom resistor so that the maximum voltage that will be applied across the resistor terminals will be 0.8V as it is the feedback voltage of the LTC3891. This was done by removing R5, cutting the trace at pin 7 (on the AD5160), in order to ground pin 7, and connect another resistor from V_{out} to pin 1. The populated PCB with these changes is shown below in Figure 5-2 and a corresponding revised schematic is shown in Figure 5-3.
Figure 5-2: Final PCB of the Smart Wall Outlet
Figure 5-3: Revised Schematic for the Smart Wall Outlet
There are a few things to point out when moving the digital potentiometer to the bottom resistor of the feedback divider. Equation 5-1 will help illustrate the difference between having the digital potentiometer on the bottom of the feedback divider as opposed to the top resistor.

\[ V_{\text{out}} = \frac{V_{\text{ref}}}{R_{\text{bottom}}} \times R_{\text{top}} + V_{\text{ref}} \]  

(5-1)

Equation 5-1 relates the output voltage of the converter to the value of the top and bottom resistors as it is divided down into the reference voltage of the LTC3891 which is 0.8V [7]. One thing that can be noticed from this equation is that the output voltage follows a linear relationship with respect to the value of \( R_{\text{top}} \). This means that if the digital potentiometer was used as the top resistor, the wiper register could be incremented, and the output voltage would follow in a linear fashion. However, if the digital potentiometer is selected as \( R_{\text{bottom}} \) then there would be a \( \frac{1}{x} \) type of relationship between the output voltage and the digital potentiometer setting. This means as the digital potentiometer is lowered in constant increments, the output voltage will increase more and more each time.

Another implication of placing the digital potentiometer as the top resistor is the maximum voltage the converter will see can be set based off of the maximum value of \( R_{\text{pot}} \). This would help prevent an overvoltage condition for the converter if the processor accidentally sets its value incorrectly. In contrast, placing the digital potentiometer as the bottom resistor allows the potential for the digital potentiometer to be set near 0Ω and
causing the output voltage of the converter to be set dangerously high.

Table 5-1 shows the experimental results as the wiper register was loaded, and the output voltage was set. A 4.12KΩ resistor was used as $R_{top}$ in the feedback divider as shown in Figure 5-3 above.

Table 5-1: Experimental Values of Wiper Register Settings and the Resulting Output Voltages

<table>
<thead>
<tr>
<th>Wiper Setting</th>
<th>Decimal</th>
<th>Hex</th>
<th>Output Voltage (V)</th>
<th>$R_{pot}$ Resistance (Ω)</th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>0xFF</td>
<td>1.46</td>
<td>4994</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>0x8E</td>
<td>2</td>
<td>2747</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>0x60</td>
<td>2.5</td>
<td>1939</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>0x4B</td>
<td>3</td>
<td>1498</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>0x3B</td>
<td>3.5</td>
<td>1221</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>0x32</td>
<td>4</td>
<td>1030</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>0x2A</td>
<td>4.5</td>
<td>891</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>0x25</td>
<td>5</td>
<td>785</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>0x21</td>
<td>5.5</td>
<td>701</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0x1D</td>
<td>6</td>
<td>634</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0x1A</td>
<td>6.5</td>
<td>578</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0x18</td>
<td>7</td>
<td>532</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0x16</td>
<td>7.5</td>
<td>492</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0x14</td>
<td>8</td>
<td>458</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0x13</td>
<td>8.4</td>
<td>434</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0x11</td>
<td>9</td>
<td>402</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0x10</td>
<td>9.6</td>
<td>375</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0x0E</td>
<td>10.1</td>
<td>354</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0x0D</td>
<td>10.7</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0x0C</td>
<td>11.3</td>
<td>314</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0x0B</td>
<td>12</td>
<td>294</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0x0A</td>
<td>12.91</td>
<td>272</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0x09</td>
<td>13.87</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0x08</td>
<td>15</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0x07</td>
<td>16.4</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0x06</td>
<td>18.1</td>
<td>191</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-4 shows this experimental data in the form of a graph relating the wiper register to the respective output voltage. Figure 5-5 shows the performance of the
AD5160 in that its resistance increases linearly as the wiper register is increased.

![Graph showing the relationship between output voltage and wiper register setting](image1)

Figure 5-4: Converter $V_{OUT}$ as a Function of the Digital Potentiometer 8-bit Wiper Setting

As discussed earlier, the output voltage does not follow a linear relationship with the digital potentiometer setting, but rather an asymptotical $\frac{1}{x}$ function.

![Graph showing the relationship between terminal B to wiper resistance and wiper register setting](image2)

Figure 5-5: Terminal B to Wiper Resistance as a Function of the Wiper Register Setting
5.3 Initial Software Development and Testing with Loads

The first phase of programming that was done was just to get the peripheral features working, while a later design phase will focus on the voltage setting algorithm. This phase of the software development was to get the correct LED flash patterns, the overvoltage detection sequence, and the re-initialization of the converter when a load is unplugged. The proper voltage setting algorithm will be looked into later.

The first load that was tested was the 6V TV. The input current vs. input voltage graph for the 6V TV is shown below in Figure 5-6. It can be seen that current already exists at the initial output voltage of $1.5V_{in}$. Because of this fact, the algorithm for setting the output was simply setting the output to 6V (as this was the expected load) at the existence of current.

![Figure 5-6: Input Characteristics for the 6V TV [6]](image-url)
During this first design phase the algorithm for setting the output was simply setting the output to 6V (as this was the expected load) at the existence of current. The pseudo-code below helps describe this initial software design.

```
While (forever)
{
    If (the output voltage is not set, and there is current detected)
    {Set the output to 6V and declare converter as “set”}

    If (the output voltage is too high)
    {
        Disable the converter
        While (forever)
            {Flash LED in a burst fashion until power cycle}
    }

    If (output voltage was previously set, and now there is no current)
    {Reinitialize the converter to 1.46V and declare converter as “not set”}

    Set LED to flash quickly or slowly whether the converter is marked as “set” or “not set”
}
```

This initial software implementation seemed to work well. The converter at turn on would idle at 1.46V with a slow LED flash, and when the 6V TV was plugged in it would automatically set itself to 6V and the LED would flash quickly. When the 6V TV was unplugged again, the output voltage would reset to 1.46, and the LED would begin to flash slowly again. To test the overvoltage protection, the limit was set around 7V, and the converter simply set the output voltage to 7V instead of 6V. When this would happen the output voltage would be brought down to 0V, and the LED would flash in its designed burst fashion until the power was cycled. Figure 5-7 shows the test setup that was used in order for this initial software development.
Since the initial testing for the 6V TV was successful, a similar approach was used in order to run the 3V radio. The only difference between this approach and the 6V TV was that the output voltage was set to 3V at the detection of current. A first look at the implementation showed that the output voltage was set to 3V when the 3V radio was plugged in, but the processor seemed to have a hard time telling the difference between no current, and the small amount of current drawn by the 3V radio. The output voltage would go back and forth from 3V to 1.46V even though the 3V radio was not disconnected. This may have been due to the lack resolution on the ADC for the processor to tell the difference between 0A and less than .05A. Upon further investigation the 3V radio, while drawing a steady 22mA with the volume very low as
listed in Table 3-1 in Chapter 3, has a very erratic input current when the volume is increased. A scope plot using a current probe measuring the input current for the 3V radio at 3V input is shown below in Figure 5-8.

![Scope Plot](image)

**Figure 5-8: Input Current for the 3V Radio at Full Volume**

Figure 5-8 shows that for this particular load, reading the output current of the converter may not be the best solution in order to set the output voltage automatically for it. As an attempt to improve this, an algorithm for averaging this output current will be looked into and discussed in a following section.

The 12V fan array will be the last load looked into for this project. This fan array had to be characterized in order to know whether or not it will draw current at 1.46V\text{in}. Figure 5-9 shows the input characteristics of the 12V fan array.
Figure 5-9: Input Characteristics for the 12V Fan Array

The fans first start moving, at around 4.5 \( V_{in} \), but they stay consistently on at 5 \( V_{in} \). It can be seen from the graph that there is no indication according to the input current to the fan array that 12V has been reached. There is just a steady slope ramping up as more voltage is applied. It can also be observed from Figure 5-9, that there is no input current at the converter’s 1.46V set point. This means that the same initial software design cannot be applied to the 12V fan array without some modifications. The two choices that were considered were either having the output voltage while idling set itself to 5V for a small window of time and if it does not detect current then reset to 1.46V, or have a button in order to tell the processor that a load has been connected and to start setting the output voltage. An obvious disadvantage of the first method is if there is a low voltage and low current device connected, the processor could think there is no load and try to poll for a load. This could damage a small load, such as the 3V radio if it is suddenly set to 5V. The second method, while requiring a small amount of user input, will keep any of the loads safe from an unanticipated overvoltage. The test point that is
on the board, will be used to model this switch. When the test point is grounded, this will signal to the processor that a load is connected. Pseudo-code below will help illustrate how this switch will fit into the code.

While (forever)
{
  If (the output voltage is not set, and the test point is grounded)
  {Set the output voltage and declare converter as “set”}

  If (the output voltage is too high)
  {
    Disable the converter
    While (forever)
    {Flash LED in a burst fashion until power cycle}
  }

  If (output voltage was previously set, and now there is no current and the test point is not grounded)
  {Reinitialize the converter to 1.46V and declare converter as “not set”}

  Set LED to flash quickly or slowly whether the converter is marked as “set” or “not set”
}

This minor adjustment to the code seemed to work as expected. When the fan array was plugged in to the 1.46V starting point nothing happened, but when the test point was grounded the converter went to 12V and the fans were running. When the fans were unhooked, the converter reinitialized as long as the test point was no longer grounded as well. Ideally a button would be used rather than a switch, this would help prevent the test point from being held to ground. If the test point was held to ground the converter will not reinitialize, and this could lead to plugging in a device straight into an overvoltage condition.
### 5.4 Voltage Setting Algorithm Development

Now that the peripheral functions of the software are functioning, there needs to be a more in depth look at the output voltage setting algorithm. Unlike the previous design, which relied on $\Delta I$ of the device, this algorithm will rely on the existence of current to step the voltage to discrete levels, and then make an intelligent decision on what to ultimately set as the output voltage. If the previous method was used, the DC fan array will have an over voltage condition because the way the current ramps up. It does not show any indication that the fan array is at 12V, but keeps increasing its current as the voltage is increased.

The first problem that was looked into was that the processor could not tell the difference between no load, and the when the 3V radio was connected. The erratic input current of the 3V radio shown in Figure 5-8 above could definitely cause the processor to misread the input current into the 3V radio. A first attempt to solve this problem was to loop through 100 samples of the current into the 3V radio, and record the maximum reading. The idea behind this being that the maximum reading would be well above 0A, and the processor can decide that there is a device connected. This solution seemed to work reasonably well, but upon further investigation it became apparent that even reading the maximum value would change significantly from reading to reading. The next approach was an attempt to average the values that were read over a large amount of samples. When using about 1000 samples the average reading was steady enough to get a reliable reading, and allowed the processor to tell the difference between no load and when the small current of the 3V radio was being drawn. This averaging method was then used in the code every time a current reading was performed.

Now that a reliable current reading algorithm had been established, the 3V radio
can be properly detected, and run at its rated voltage. Since all three of the test loads can now be detected and run off of the Smart Wall Outlet, a closer look at a voltage setting algorithm is in order. As discussed previously the use of the on-board test point will be used to help signal that the converter is loaded with a device. The processor will begin stepping up the voltage when the test point is grounded while monitoring the current. In the event there is nothing attached and the test point is grounded, the processor will step up the output voltage up to 12V, and if there is still no current detected it will reset to 1.46V. This will help the converter from being falsely set to a voltage even when there is nothing attached at the output. The fact that the 12V fan array will not draw current until an input voltage of 5V, will be used in order to help the processor identify that this particular load is connected. Figure 5-10 helps illustrate this voltage setting algorithm.
This new software development stage still has all of the peripheral software design included. The LED flashing patterns will remain the same as listed in Table 4-4 in the previous chapter, the over-voltage is still in effect, as well as the re-initialization of the output voltage to 1.46V when a load is disconnected. With all of these peripheral features still intact, the overall algorithm and program seemed to work well. The Smart Wall Outlet could properly set its voltage in accordance to each of the three loads when the load was connected, and the test point was momentarily grounded.

5.5 Efficiency Measurements

As with any DC-DC converter, efficiency is a primary concern. With that in mind some efficiency measurements and calculations for the Smart Wall Outlet were taken. Of the three loads that were used in the development of the Smart Wall Outlet, only the 6V
TV, and the 12V fan array were used for efficiency measurements. The 3V radio current was difficult to measure due to its sporadic nature. Equation 5-2 shows the calculation used to find the efficiency and Table 5-2 below shows the input and output characteristics needed to calculate efficiency for the Smart Wall Outlet being loaded with the 6V TV and the 12V fan array.

\[
\eta(\%) = \frac{P_{\text{out}}}{P_{\text{in}}} = \frac{V_{\text{out}} \cdot I_{\text{out}}}{V_{\text{in}} \cdot I_{\text{in}}}
\] (5-2)

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage (V)</td>
<td>Current (A)</td>
<td>Voltage (V)</td>
</tr>
<tr>
<td>6V TV</td>
<td>48</td>
<td>0.064</td>
</tr>
<tr>
<td>12V fan array</td>
<td>48</td>
<td>0.0815</td>
</tr>
<tr>
<td>Unloaded</td>
<td>48</td>
<td>0.0076</td>
</tr>
</tbody>
</table>

At 82% efficiency the converter is not running as efficiently as it should. It is not uncommon to have a DC-DC converter running above 90% efficiency. One cause of the low efficiency could be the low loading conditions for the main converter. The converter was designed for 48W, but the loads that were used were around 3W. It is common that when a converter is lightly loaded the efficiency will be at its lowest. Another potential cause of the low efficiency might be the choice of MOSFETs. The MOSFETs that were used were probably not the most efficient choice. Package size was reduced by choosing a part that had a higher $R_{\text{DSon}}$ which will increase conduction losses from switching and result in a lower efficiency. Another factor resulting in a high $R_{\text{DSon}}$ was the fact that they are relatively high voltage and current MOSFETs. At a 60V/4A rating, they had a higher $R_{\text{DSon}}$ than would be desirable for a switching MOSFET at above 30mΩ [15].
Chapter 6: Conclusions and Recommendations

The main goal of this thesis was to improve upon the previous Smart Wall Outlet. This was done in a few different ways. The original Smart Wall Outlet had three different boards included in its design, and this new design has only one compact board. This allowed for a more compact package that can be mounted inside of a wall outlet. The software design was also an improvement. This new design allows for control of the main converter such as enabling or disabling the converter, an on-board LED to signal to the user in what state the converter is in, the ability to protect the converter from an overvoltage condition, and the re-initialization of the converter when the load is unplugged. The actual voltage setting algorithm took a different approach compared to the previous design, but seemed to work well at least for the three loads that were designated for this project. This new algorithm relied on a few assumptions. One being that if the input current existed at the 1.46V starting point, and it was really small, it was probably a 3V device. The next was that if this current wasn’t very small, then it would be a 6V device, with the assumption that it probably has four 1.5V AA batteries (6V) instead of two. The last assumption was that if there was no current existing up until above 5V, then it must be a 12V device. This last one could probably be a more safe assumption because there are a lot of 12V devices that could be plugged in, and at 1.46V they probably will not be drawing current. If there were more devices, then this design could be extended to accommodate more loads.

There were a few problems observed with the testing of this converter. As previously discussed the resistance of the potentiometer would ramp up as a $\frac{1}{x}$ function.
This made for inaccuracy at the higher output voltages. The smallest incrementation of the digital potentiometer (1-LSB) would jump the voltage up by over 1V. As shown in Table 5-1, the output voltage cannot be set to 14V or 16V in this fashion because of the way the curve in Figure 5-4 ramps up at lower R\textsubscript{pot} values. This means if there was a device that ran nominally off of 16V for example, the closest the output voltage could be in this configuration, would be 16.4V.

Another problem noted during testing, was that the previous method of using the ΔI\textsubscript{out} is invalid for some loads. The 12V fan array would keep ramping up its current as the input voltage was increased in a very linear fashion, which does not allow for the processor to detect a ΔI\textsubscript{out} that would signal that it has reached its nominal V\textsubscript{in}. A solution for this problem would be to possibly find a digital potentiometer that can accept higher voltages on its resistor pins. This way the R\textsubscript{pot} can be used as the top resistor in the divider, but this might cause problems with the converter’s stability.

Another observation that could have turned into a problem if there had been a heavier load, came from using one current sense amplifier to read in current. In this design, the maximum 4.096V (ADC reference) input into the ADC would result in .768A being drawn. This means if there was anything above this current being drawn, the processor would still think there was .768A being drawn. So if there was a higher load that was needed, the sense resistor/current sense amplifier designed in Section 4.6 would have to be altered. This would increase the maximum reading the processor could read, but at the cost of losing resolution at lower output current loads like the 3V radio. This could be remedied by implementing more current sense amplifiers in series with their own individual sense resistors. The idea behind this being that one amplifier would be
for smaller currents, one for mid-range currents, and the last for higher currents. This way the processor can still read in higher values of current, without losing resolution to the smaller currents. This solution would come at an efficiency and cost penalty however because there would be multiple sense resistors in series with the output current, and multiple current sense amplifiers.

The next possible steps to improving this design include characterizing more loads, and adjusting the software to work with them, particularly a higher load near the design specification of 50W. This algorithm that was used can be developed further with the use of these extra loads. Hardware improvements could also be made by the overall packaging of the unit. Ideally the unit should fit inside of an outlet box that will be mounted on the wall of the DC house the button to signal a load is connected, as well as the LED would have to be accessible to the front panel of the outlet. Another change would be to find a digital potentiometer that can handle a higher voltage across it so it can be used as the $R_{top}$ in equation 5-1 and thus avoiding the $\frac{1}{x}$ relationship between the digital potentiometer setting and the output voltage.

Overall, basing the design off of reading the output current of the converter is complicated and has its pitfalls. This method, while being successful in this Thesis for the three loads that were used, might not work as planned if there was another load with unpredicted input characteristics. If more loads were tested, and the software was adjusted accordingly, then the present method could operate more successfully, although true universality will still be a challenging trait to achieve.
BIBLIOGRAPHY


doi: 10.1109/ISGTEurope.2012.6465882

doi: 10.1109/ICHQP.2008.4668827

doi: 10.1109/TIA.2013.2265873

<http://www.vishay.com/docs/65351/sq3426eev.pdf>

<https://courseware.ee.calpoly.edu/~jharris/research/super_project/super_table_of_contents.htm>
Appendix A: Microprocessor Code

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

uint16_t V_read;
uint16_t I_read;
uint16_t I_max;
uint32_t I_avg;

int Vset = 0;
int i = 0;
int high_V = 0;
float V_out = 1.5;

void SPI_init(void) //initialize SPI registers
{
    DDRB = ((1<<DDB4)|(1<<DDB5)|(1<<DDB7));
    SPCR = ((1<<SPE)|(1<<MSTR)|(1<<SPR0)|(1<<CPOL)|(1<<CPHA));
}

void Indicate_fault(void) //Fault Detected
{
    PORTD &= ~(1<<PD4); //disables the converter
    while (1)
    {
        _delay_ms(1000); //Burst Flash for LED
        for (i=0; i<3; i++)
        {
            PORTD &= ~(1<<PD6);
            _delay_ms(90);
            PORTD |= (1<<PD6);
            _delay_ms(90);
        }
    }
}

void ADC_init(void) // enable ADC and divide clock factor =128
{
    ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}

uint16_t adc_read(uint8_t ch) //Reads ADC on channel that is Selected
{
    I_read = 0;
    ch &= 0b00000011;
    ADMUX = (ADMUX & 0xF8)|ch;
    ADCSRA |= (1<<ADSC);
    while(ADCSRA & (1<<ADSC));
    return (ADC);
}

char SPI_Transmit(char cData) //Transmits 8-bit number through SPI
{
    PORTB &= ~(1<<PB4);
    SPDR = cData;
    while(!(SPSR & (1<<SPIF)));
PORTB |= (1<<PB4);
return SPDR;
}

void check_Fault(void)
{
    V_read = adc_read(1); //read ADC from channel 1 (voltage sense)
    if (V_read > 811 || V_read <50) //if fault is flagged need to cycle power
    {
        Indicate_fault();
    }
}

void check_current(void) //Obtains max current and average current based
//on 1000 samples
{
    I_max=0;
    I_avg=0;
    for(i=0; i<1000; i++)
    {
        I_read = adc_read(0);
        I_avg = I_avg +I_read;
        if (I_read > I_max)
        {
            I_max=I_read;
        }
    }
    I_avg=I_avg/1000;
}

void Set_Voltage(float Set_command) //sets voltage and checks for a fault
{
    if (Set_command==1.5)
    {
        SPI_Transmit(0xFF);
        V_out=1.5;
    }
    else if (Set_command==3)
    {
        SPI_Transmit(0x4B); // set to 3V
        V_out=3;
    }
    else if (Set_command ==5)
    {
        SPI_Transmit(0x25); // set to 5V
        V_out=5;
    }
    else if (Set_command ==6)
    {
        SPI_Transmit(0x1D); // set to 5V
        V_out=6;
    }
    else if (Set_command ==7)
    {
        SPI_Transmit(0x18); // set to 7V
        V_out=7;
    }
    else if (Set_command==9)
    {
        SPI_Transmit(0x11); // set to 7V
        V_out=9;
    }
    else if (Set_command==12)
    {
        SPI_Transmit(0x0B); //set to 12V
        V_out=12;
int main(void)
{
    SPI_init();
    ADC_init();
    DDRD = (1<<DDD4)|(1<<DDD6);  // sets portD for Enable, LED and test point
    PORTD = 0x20;
    SPI_Transmit(0xFF);  // initializes the digital pot/output voltage
    PORTD |= (1<<PD4);  // Pulls PortD pin4 high to enable the main converter
    PORTD |= (1<<PD6);  // Keeps LED off for initialization
    _delay_ms(100);
    while (1)
    {
        check_current();
        if ((Vset > 0) && (I_avg < 262))  // if previously set to a voltage,
        {
            Set_Voltage(1.5);  // Reinitialize converter
            high_V = 0;
            Vset = 0;
        }

        if (!(PIND & (1<<PD5)))  // if test point is grounded.
        {
            check_current();
            while (Vset == 0)  // code for setting the output voltage
            {
                PORTD &= ~(1<<PD6);  // keep LED on while searching for
                // correct voltage
                while (I_avg < 258)  // Routine when no current detected,
                {  // and test point is grounded
                    Set_Voltage(5);
                    check_current();
                    if (I_avg > 262)
                    {
                        high_V = 1;
                        break;
                    }
                    _delay_ms(1000);
                    Set_Voltage(7);
                    check_current();
                    if (I_avg > 262)
                    {
                        high_V = 1;
                        break;
                    }
                    _delay_ms(1000);
                    Set_Voltage(9);
                    check_current();
                    if (I_avg > 262)
                    {
                        high_V = 1;
                        break;
                    }
                    _delay_ms(1000);
                }
            }
        }
    }
}
Set_Voltage(12);
check_current();
if (I_avg > 262)
{
    high_V = 1;
    break;
}
_delay_ms(1000);
Set_Voltage(1.5);
break;
}

while (I_avg >262 && V_out <3)//routine for when the Test point
    //grounded, but there was already
    //Current detected
    
    Set_Voltage(3);
    check_current();
    if (I_avg > 400)
    {
        Set_Voltage(6);
        check_Fault();
        break;
    }
    _delay_ms(1000);
    break;
}

while (I_avg >262 && high_V==1) //when a "high Voltage load"
    //is detected, set to 12V
    
    Set_Voltage(12);
    check_current();
    break;
}
check_Fault();
if (V_out==1.5)
{
    Vset = 2;
}
else
{
    Vset = 1;
}

if (Vset == 0)
{
    _delay_ms(700);
    PORTD ^= (1<<PD6); //slow LED flash indicating Vout is not set
    //and unloaded
}
else if (Vset == 1)
{
    _delay_ms(15);
    PORTD ^= (1<<PD6); //Fast LED flashing to indicate Vout is set
    //and a device is attached
}

return 0;