DC-DC Converter

John Poirier, Elie Chauvin
UVM Department of Electrical Engineering

Table of Contents

Description:

This project required the construction of a DC-DC converter using a buck circuit, and HV switch circuit, and an Arduino Metro Mini. The Circuit would be powered by a 20-volt source, either a benchtop source or a solar panel, and would output a desired DC output, in our case a 1.2-volt output, to a load, which for us was a 1.2-volt DC motor. The Arduino would provide a pulsed source from the 20-volt source, the HV switch would turn the 20-volt DC source into a 20-volt pulsed source, and the buck circuit would convert the 20-volt pulsed source into a 1.2-volt DC source.

Solution:

a. System

Our approach for this project was to break it into sections that can then be integrated. First, we built the buck circuit, which gave us a stepped-down DC output voltage using the pulse generator source. Then, we created the HV switch circuit, which takes the 20-volt DC source and a pulsed source and converts it into a 20-volt pulsed source. Finally, we coded the Arduino, powered by the 20-volt DC source being stepped down to 5 volts by a voltage regulator, to provide a pulsed source that we tuned to a specific value. This Arduino acts as the pulse source for the HV switch. Once all the circuits were built separately, we combined them into one circuit that powered a 1.2-volt DC motor.

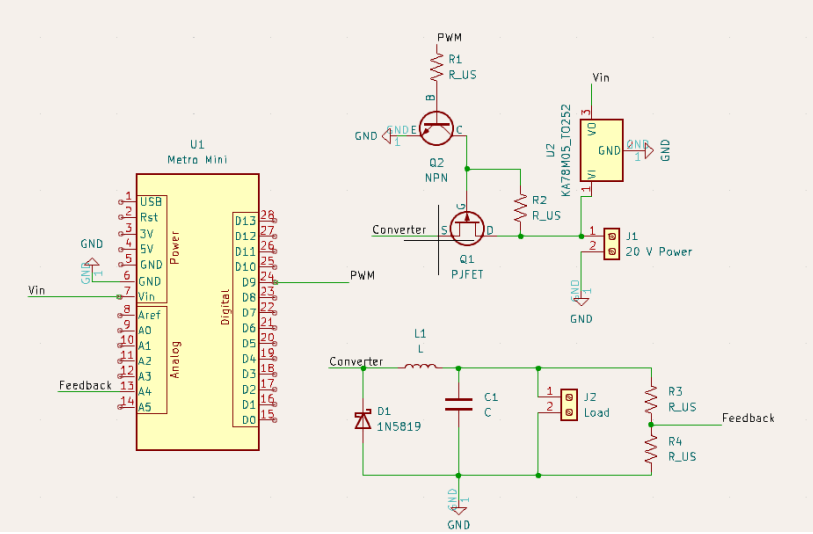


Figure 1: Circuit Schematic

b. Key Materials

The key materials for this project are the Arduino, NPN and PFET transistors, the Schottky diode, and the voltage regulator.

Component	Cost
NPN Transistor	$0.42 (DigiKey)
PFET Transistor	$0.52 (DigiKey)
5V Voltage Regulator	$0.94 (DigiKey)
1N5819 Diode	$0.50 (DigiKey)
Arduino Metro Mini	$14.95 (DigiKey)

The full Bill of Materials can be found in the Appendix.

c. Cost Analysis

This project was relatively inexpensive, as the transistors, inductors, and capacitors were all under a dollar, and the Arduino was roughly 15 dollars. Prototyping this is significantly more expensive, as buying components in bulk is much more cost-effective. The total cost of our prototype was $19.66, as calculated using the Bill of Materials.

Schedule:

Initially, we created a Gantt chart to have a timeline of when certain tasks would be accomplished. We were able to follow the timeline relatively well, with one major caveat. Due to time constraints, we were uncertain if there would be enough time to order a prototype PCB, and in the end, we were unable to get a PCB shipped in time. Other than this hurdle, we were able to follow the timeline relatively well, with some issues along the way.

Below is our Gantt chart at the time of mid-project status reports, followed by our updated Gantt chart.

Table 1: Gantt Chart at time of Status Reports

DC-DC Converter	03/18 - 03/22	03/25 - 03/29	04/01 - 04/05	04/08 - 04/12	04/15 - 04/19	04/22 - 04/26	04/29 - 05/3
LTSpice Simulate							
Build Converter							
Build Switch							
Combine Circuit							
Integrate MCU							
Design Schematic							
Design PCB							
Order PCB?							
Solder PCB?							
Final Report							

Table 2: Updated Gantt Chart

DC-DC Converter	03/18 - 03/22	03/25 - 03/29	04/01 - 04/05	04/08 - 04/12	04/15 - 04/19	04/22 - 04/26	04/29 - 05/3
LTSpice Simulate							
Build Converter							
Build Switch							
Combine Circuit							
Integrate MCU							
Design Schematic							
Design PCB							
Order PCB?							
Solder PCB?							
Final Report							

Results

a) Buck Circuit Design

For this circuit design, we followed the example of a basic buck circuit given during the lecture period. The schematic for this circuit is shown below in Figure 2. We built this circuit on a breadboard using a function generator as our power source to create a prototype and validate the circuit. The circuit pulls down the voltage from your source to a value specified by the parameters of the pulse and the values of the components used. Initially, we used a resistor as the load for ease of measurement, then moved on to using a DC motor as the load. Then we integrated the HV Switch circuit and the Arduino

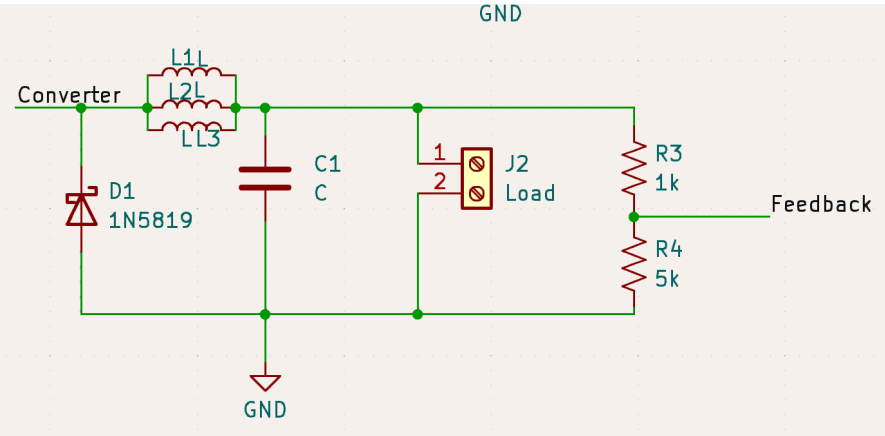


Figure 2: Buck Circuit

b) HV Switch Design

For this circuit design, we followed the example of an HV switch provided by Dr. Gallagher. The schematic for this circuit is shown below in Figure 3. We built this circuit on a breadboard using the benchtop source to create a prototype and validate the circuit. This circuit takes a DC source and a pulse source and turns on and off the circuit periodically, outputting a pulse source with the DC source's voltage value as the on voltage. In this case, we are taking a 20-volt DC source and turning it into a 20-volt pulse source. Then, we integrated the buck circuit and the Arduino with this switch circuit.

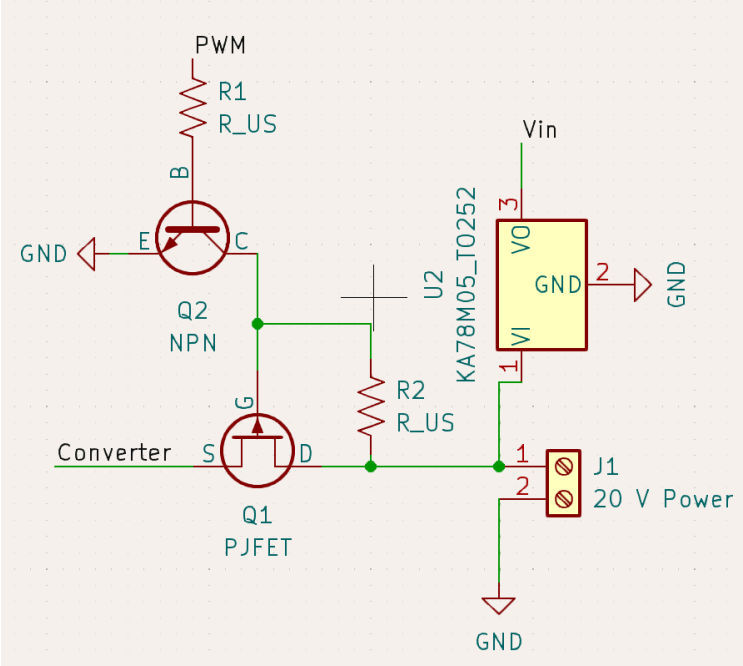


Figure 3: HV Switch

c) Programming/Integrating MCU

To code our Microcontroller Unit (MCU), we based our foundation on the example code provided by Dr. Gallagher. Then, we edited the values for our initial pulse width, our period, and our target output voltage. Through testing, we found values that gave us our desired output. Notably, we ended up using an initial pulse width of 0, resulting in a delayed output. This code is shown in the Appendix.

d) PCB Schematic and Layout

When designing our PCB schematic, we used our prototype circuit and copied that into KiCad. We used labels to connect our circuits so that we could keep our three circuits visually separate for ease of troubleshooting. The full schematic can be found in the Appendix.

When designing the PCB layout, we first found the appropriate footprints for each of our components. We had to measure the components to ensure accuracy for our dimensions. We added two screw terminals to act as connecting points for our 20-volt DC source and our load motor to the circuit. Our goal when designing the PCB layout was to maintain a compact form while also keeping important components, like the Arduino and screw terminals, easily accessible. For our copper pours, we had one layer connecting the 20-volt net to all appropriate nodes and the other layer connecting the ground net to all appropriate nodes. This resulted in the least number of traces. The PCB Layout can be found in the Appendix.

e) Tuning to Motor

Initially, our prototype circuit was designed to have a DC output of 5 volts going across a load resistor. However, the motor we used for our load is rated at 1.2 volts. To account for this, we tuned our component values to provide higher power ratings to avoid burning out any components. We also had to edit our code to achieve an output voltage of 1.2 volts instead of 5 volts.

f) Simulations

A critical portion of this project was simulations. We created an initial simulation before creating any of our physical circuits to make sure the behavior was what we were looking for. Then, as we built our prototype, we simulated it to ensure our measurements made sense. Then, with our final circuit design, we restimulated with our new component values and pulse parameters and visualized the output across the load motor. Simulation results can be found in the Appendix.

Validation

With our fully functioning DC-DC converter, the final task was to take measurements to find power efficiency and voltage ripple.

a) Efficiency Assessment

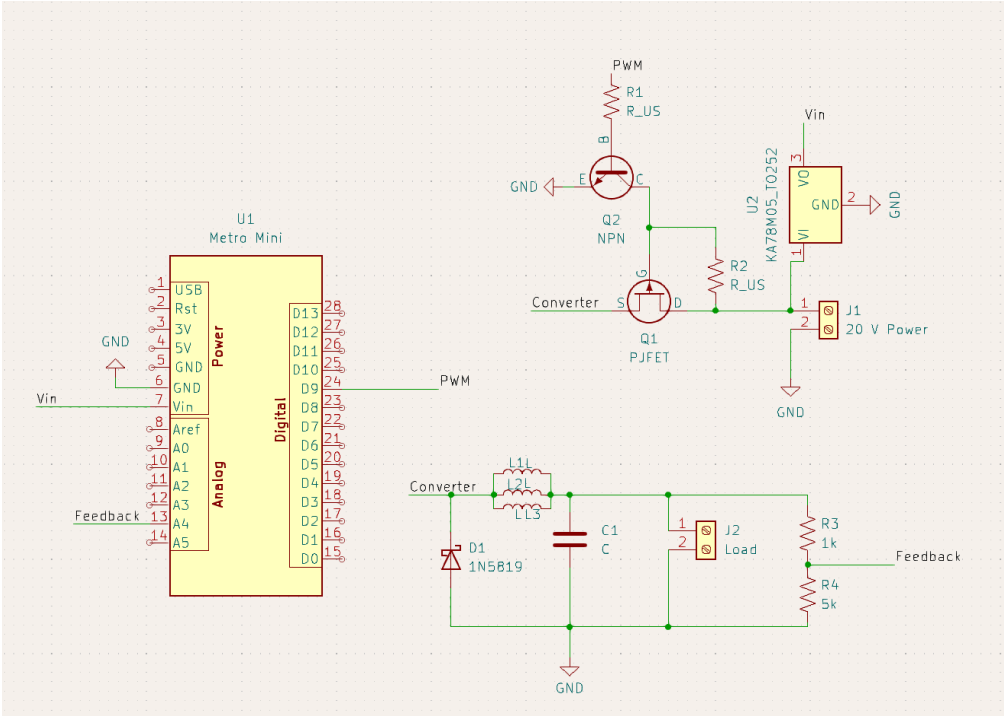
To evaluate the efficiency of the circuit, we take our input power and compare it to our output power. Due to the nature of this circuit, converting a high-voltage source to a low-voltage source, we expect the efficiency to be low. Our circuit pulls 20 volts and 0.302 amps from the source, and outputs about 1.2 volts at 0.487 amps. Calculating power, we find our input power is 6.04 watts and our output power is 0.5844 watts. Using the equation efficiency = (output power/input power) *100%, we get an efficiency of 9.6%, which is exceptionally low. This makes sense, as all of the components get relatively hot during operation, signifying very high power losses.

Appendix

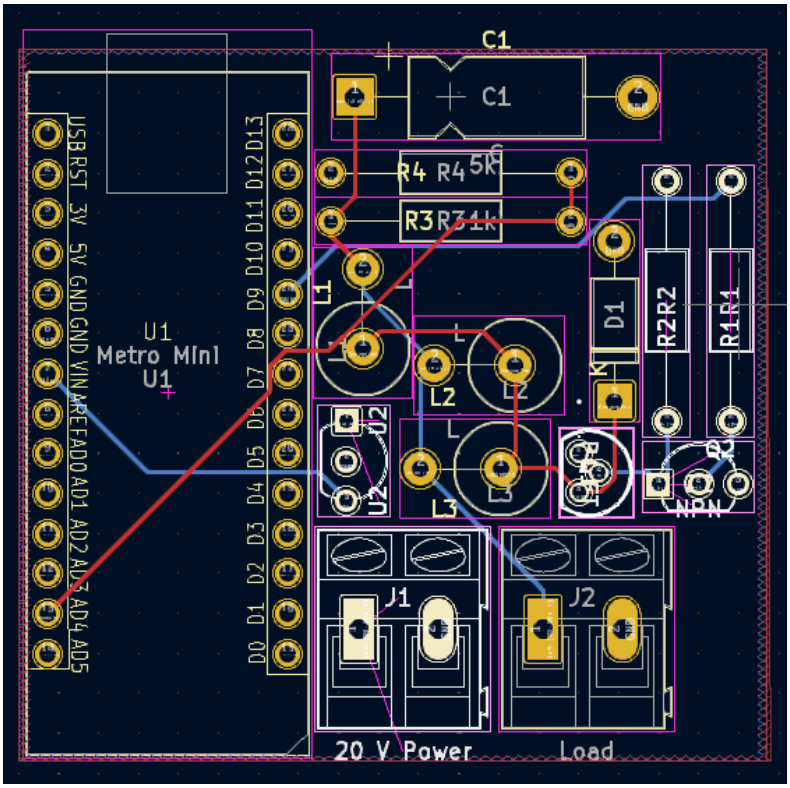
Full Bill of Materials

Component	Cost
NPN Transistor	$0.42 (DigiKey)
PFET Transistor	$0.52 (DigiKey)
5V Voltage Regulator	$0.94 (DigiKey)
Screw Terminal (2)	$0.70 (DigiKey)
Resistors (Varying) (4)	$0.10 (DigiKey)
47 uF Capacitor	$0.26 (DigiKey)
1mH Inductor (3)	$0.27 (DigiKey)
1N5819 Diode	$0.50 (DigiKey)
Arduino Metro Mini	$14.95 (DigiKey)

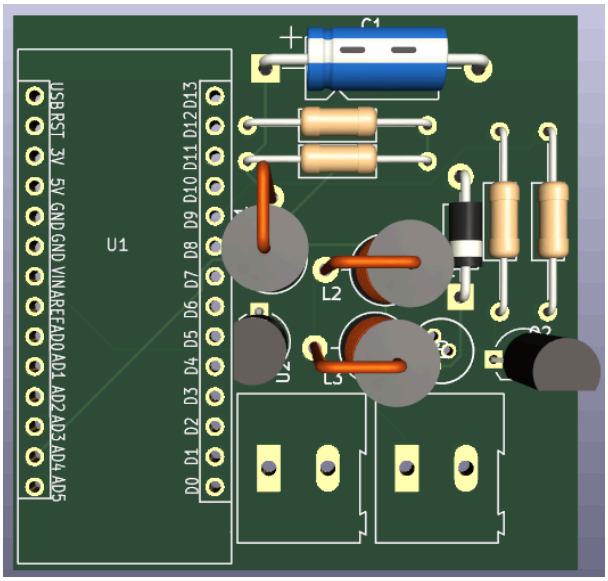
Full Schematics and Layout



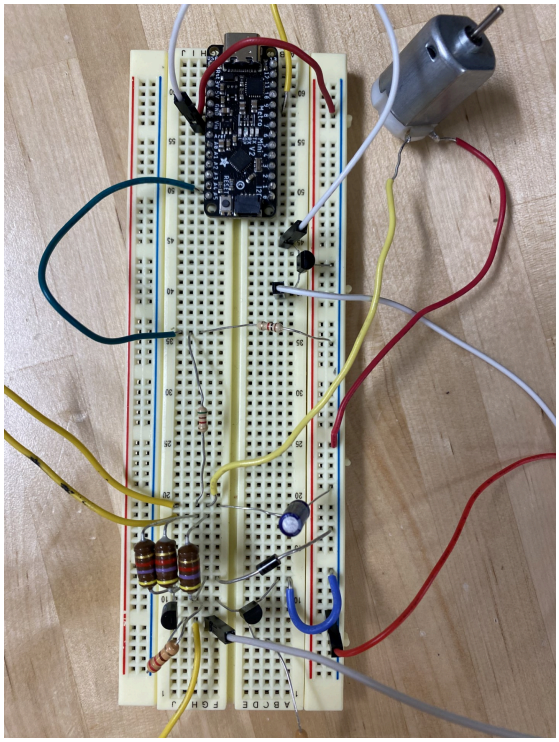
Schematic



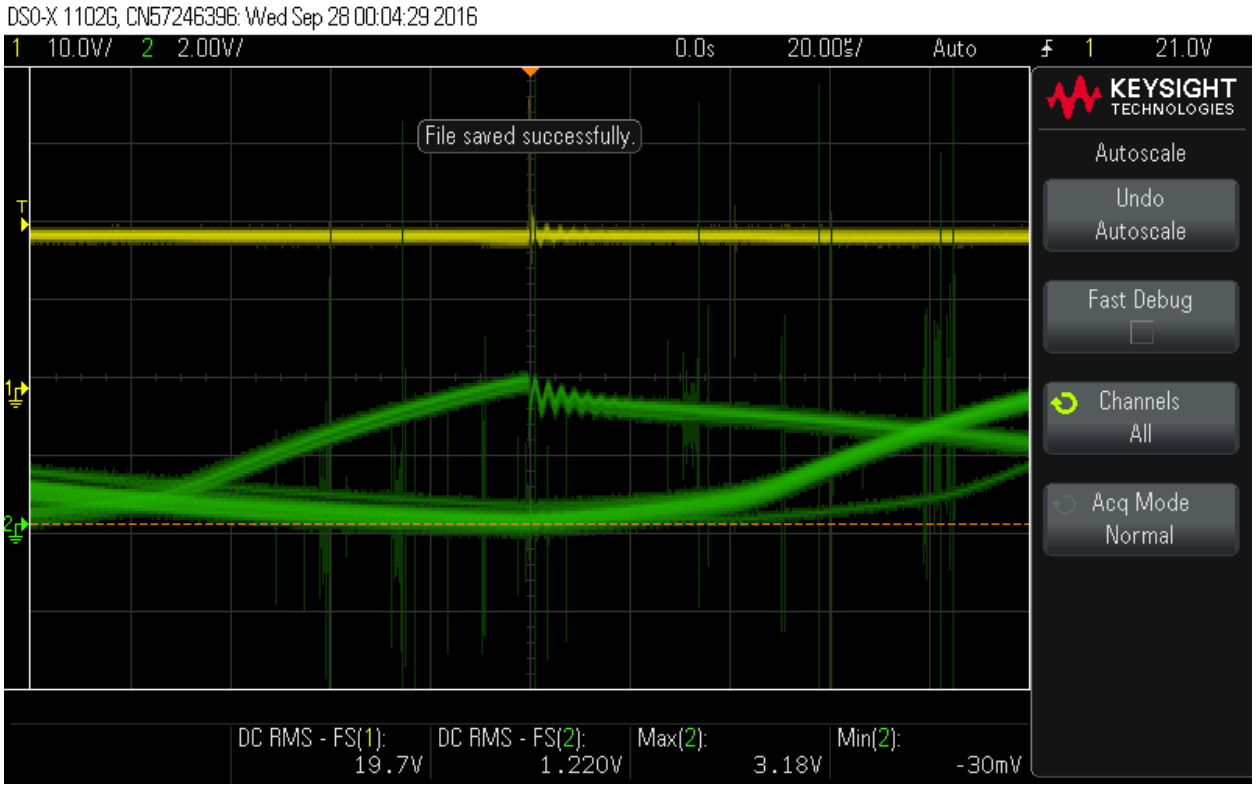
PCB Layout



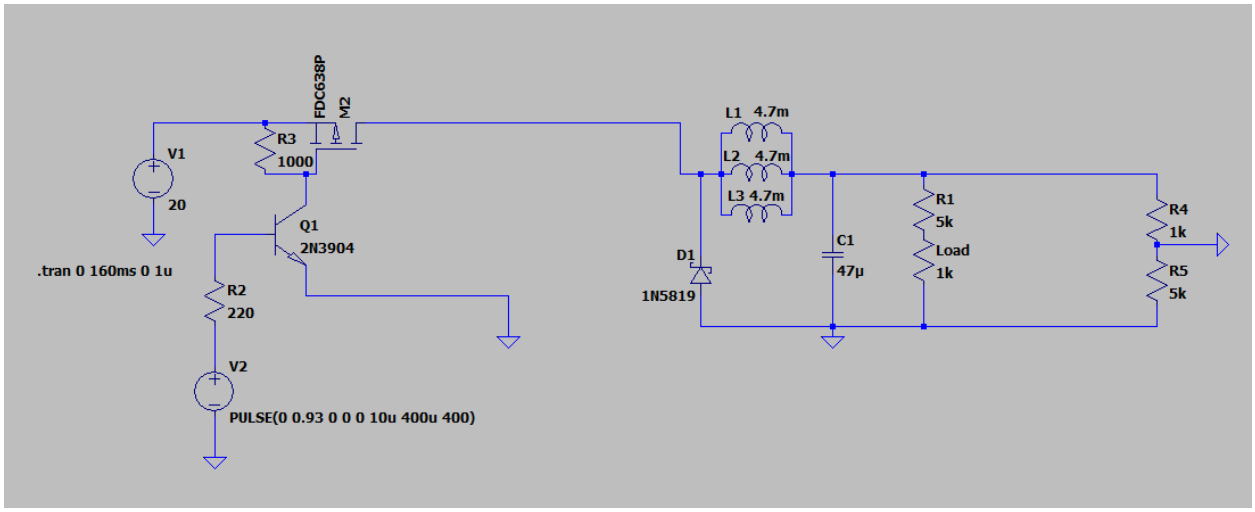
3D View of PCB Layout



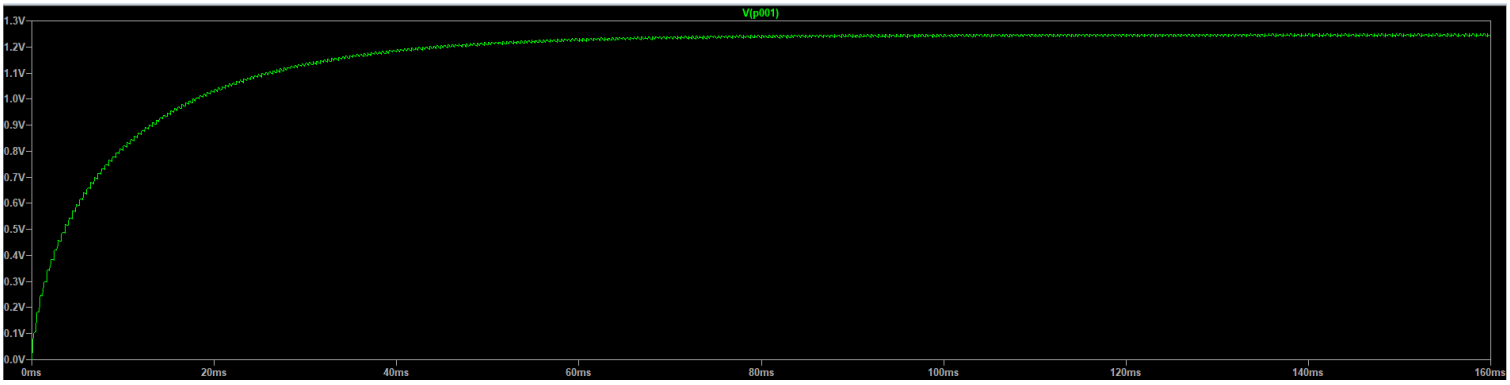
Final Prototype

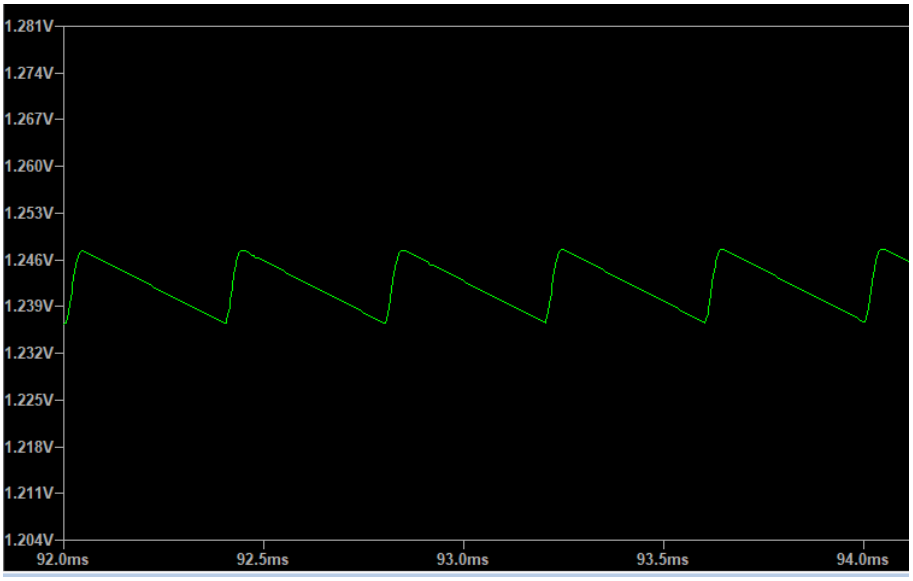


Oscilloscope Output Measurement (Green-Output, Yellow-Input)



Simulation Schematic





Simulation Results Measuring Voltage Across Load

Source Code

```
/*********************************************************/
/*************Defines Here**********************/
#include <myTimer.h>    // Make sure myLibs is installed

#define PWinit 0  //  In micro seconds.   Max 4000 us = 4ms
#define PER 400   //  In micro seconds.  Max 4000 us = 4ms
#define VOUT  0.93    // this is your target voltage out.  It can be floating point!!

#define PULSE_PIN 9
#define ANALOGINPIN 4

/*********************************************************/
/******* Prototype Function declarations ******/
//*********************************************************/

/*********************************************************/
/*************Global Variables Here**********************/
/*********************************************************/
   int senseValue;
   int senseTarget=VOUT*54;  //52.4  home system1  34.2 system 2
   int i= PWinit*16;

void setup() {

  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  initPWM_fast(PER*16);    // Pin 9

   // Let system relax for 1s

   pwmPW_PER_fast(i,PER*16);
   delay(4000);
}


void loop(){
    senseValue = analogRead(ANALOGINPIN);

    if(senseValue>(senseTarget+2))i-=1;
    else if(senseValue<(senseTarget-2))i+=1;
    if(i<0)i=0;

    pwmPW_PER_fast(i,PER*16);

    //Serial.print(" \n data =");
     //Serial.print(i);
   // Serial.print(senseTarget);

    delay(2);
}
```

Sources

Dr. Matthew Gallagher, https://brightspace.uvm.edu/d2l/home/63830