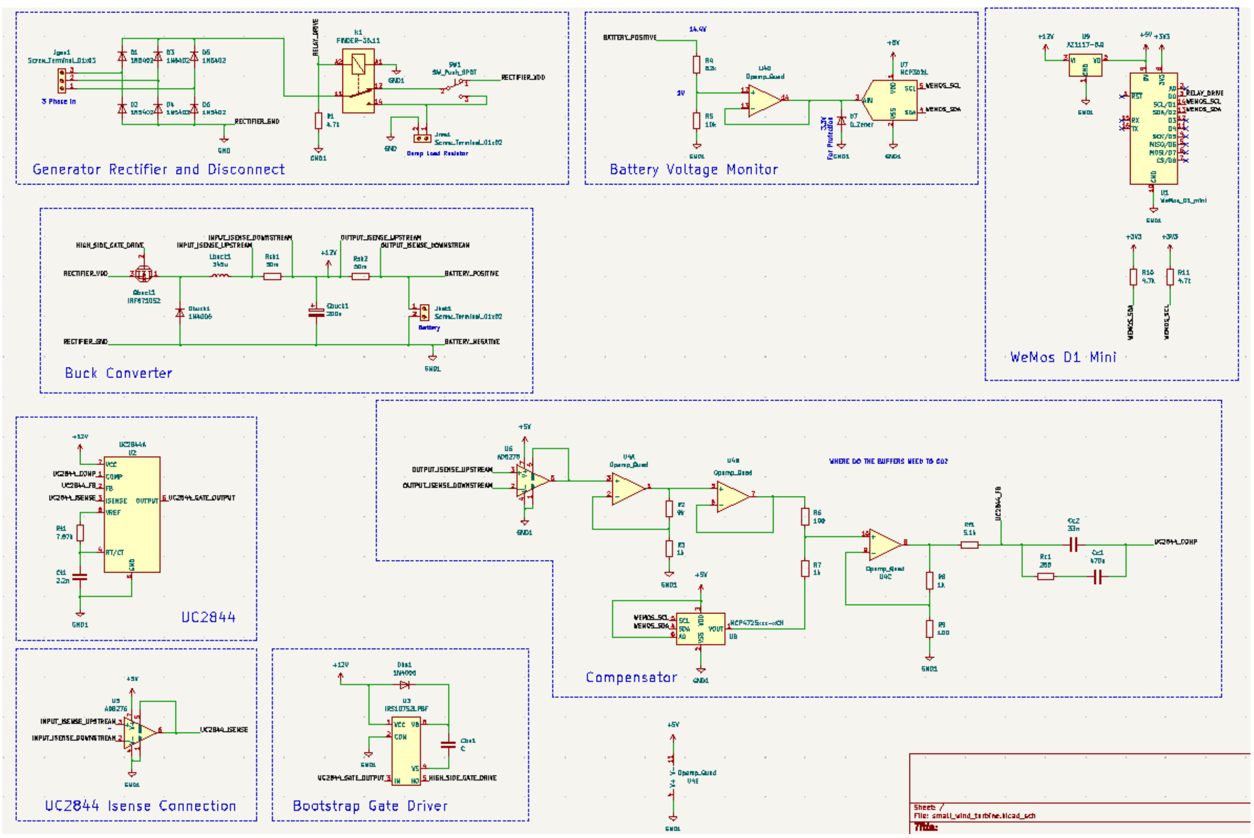
Small Turbine Power Electronics Design

Author: Ian Eykamp
Date: June 26, 2023

Purpose:

This document is a detailed design description for the power electronics for the small wind turbine. My work so far consists of a PLECS model, a KiCAD schematic and layout (coming soon), and a couple LTspice simulations. After review, this design will be manufactured on a PCB and validated by testing on the small wind turbine. Ideally, the same circuit can be scaled up for use on the large wind turbine, with bulkier components.

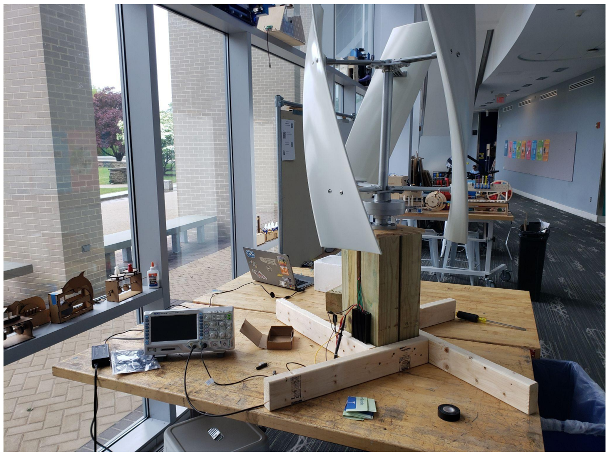
Overview:

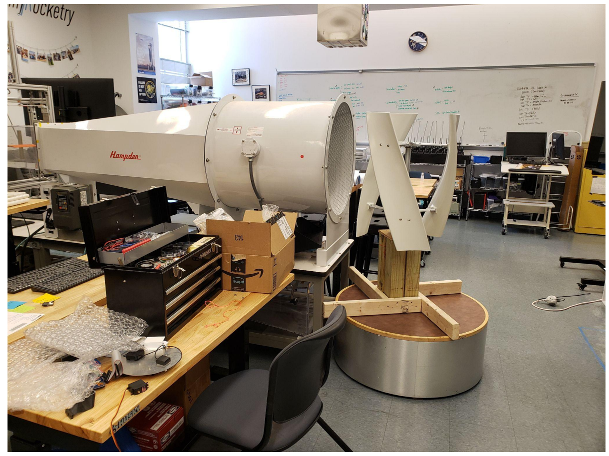


The circuit consists of a 3-phase rectifier with disconnect circuitry, the buck converter circuit, compensator circuit with adjustable current control, UC2844 with auxiliary current sense and gate driver ICs, and a microcontroller for monitoring the battery level and specifying the charge current.

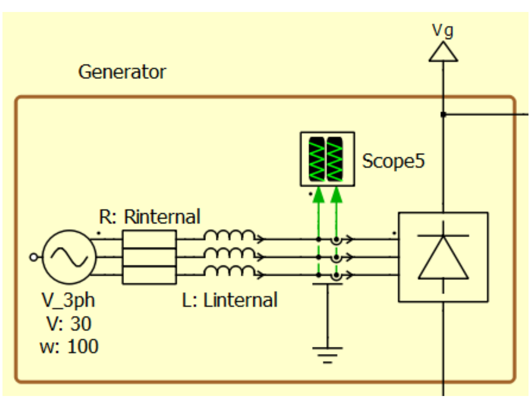
Generator

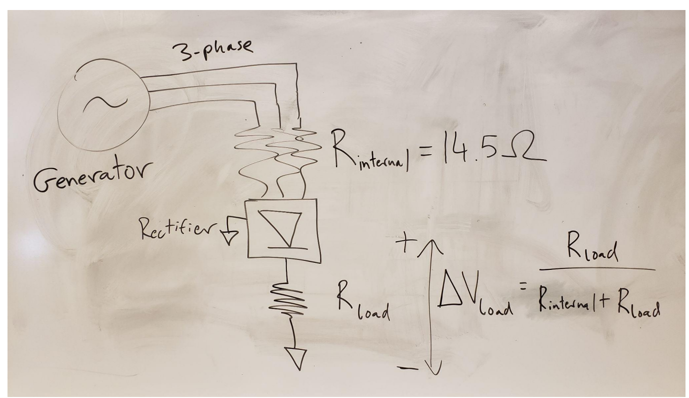
The generator sits at the base of the small vertical axis wind turbine. It outputs 3-phase power when the turbine is spun. See [this slideshow]() on how 3-phase power works. The three wires from the generator connect into the Jgen1 terminal on the PCB.





The generator has an internal resistance of 14.5Ω, which can be modeled with a resistor in series with each of the three phases. In larger generators, the inductance of the coils is also important, but here I have assumed the inductance is zero. It is important to know the internal resistance, because it determines how much current we are able to draw from the generator at a given voltage, as follows.





No matter what its form, everything to the right of the rectifier can be treated as a load resistor with an impedance equal to R = V / I. This is true for our circuit, where we are charging a battery at a certain voltage (12V) with a certain current (0.25A). Our output impedance is 48Ω. The load impedance and the internal resistance form a voltage divider, such that we only experience $V_{gen} \cdot \frac{R_{load}}{R_{internal} + R_{load}}$ across the output.

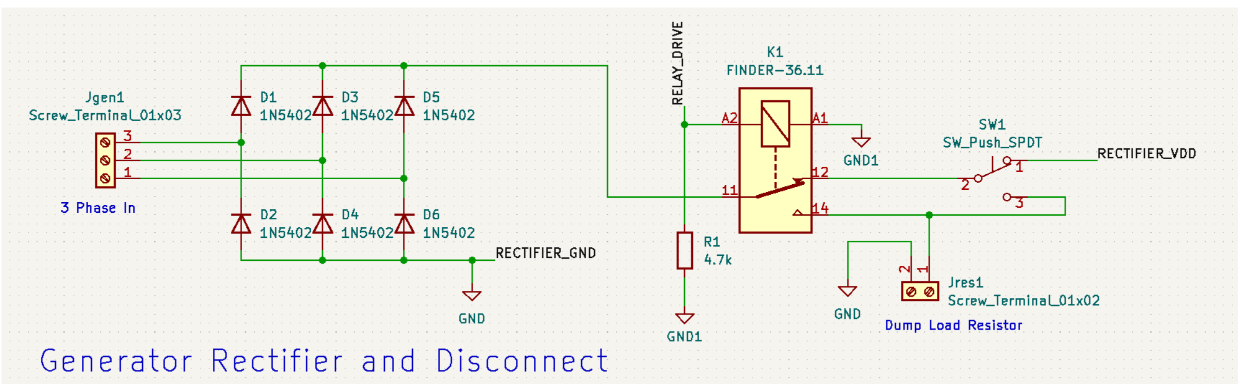
We need to charge a 12V battery (really, when it's done charging, it will be at 14.4V) using our buck converter. Because of the chip we are using, our buck converter can have no more than a

50% duty cycle, which means our input voltage must be at least 2 * 14.4V = 28.8V at a bare minimum. Round up to 30V.

To spin the generator and get power out of it, I will place it in front of the 4th floor wind tunnel in the Rocketry lab. Without getting to a point where I felt it was dangerous, I can get it to 30V with some load resistance. I am confident I can spin it faster if we need a higher voltage.

For the circuit design, I assume the generator voltage is nominally 30V on each wire, but that means the rectified voltage is $30V \cdot \sqrt{3} \approx 50V$, because the rectifier takes the maximum difference between any two wires. Again, see [this slideshow](). So our voltage divider has to give us at least 30V from the 50V available. That limits the output impedance we are allowed to have to somewhere around 25Ω (after conservative rounding). We know our output voltage will be >12V, so by I = V / R = 12V / 25Ω, our output current should not greatly exceed 0.5A. Excellent. I designed with a quarter of an amp in mind.

Rectifier and shut-off switches



The rectifier is also explained in the [slideshow]() mentioned previously. I use Schottky diodes with the lowest forward voltage drop I could find for these diodes, to avoid dropping the voltage too much during rectification. A smaller voltage drop also means less power will be dissipated, since all of the current passes through these diodes.

Next is a relay in series with a manual switch. The relay is essentially a switch that is controlled by the microcontroller to shut off current when the battery is charged or in case something goes wrong. The manual switch serves the same purpose. It can be seen from the schematic that if either of these switches is opened, the current will flow through a dump resistor instead, electronically braking the turbine. The dump resistor is separate from the PCB and will be connected via a plug-in terminal.

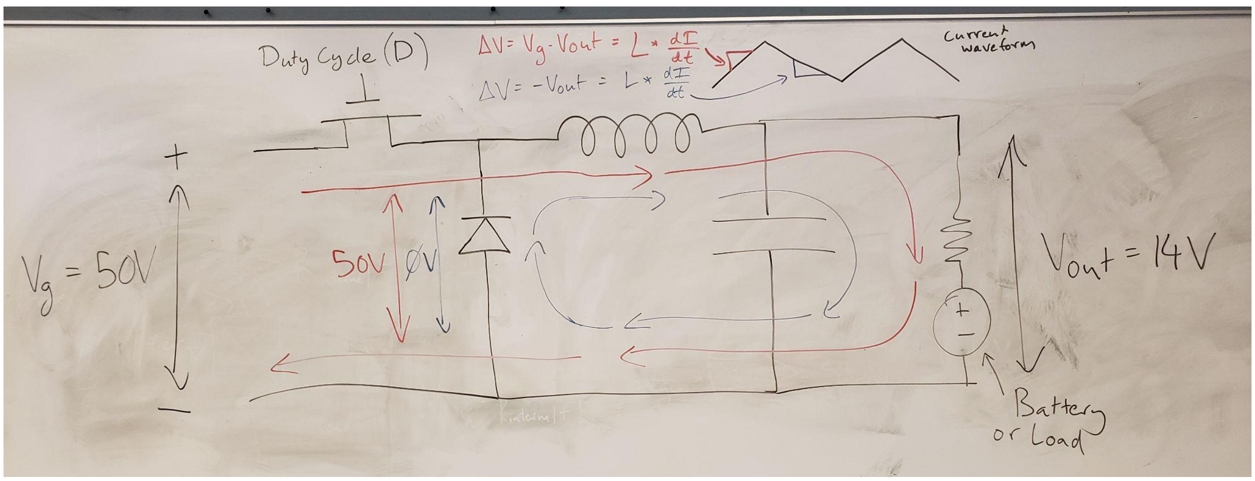
The dump resistor will be 16Ω to match the generator internal resistance and rated for 300W, with its own heat sink. That should be more than enough for a 50V input, even if all the power is dissipated through the dump load and not through the internal resistance. We have

$P = \frac{V^2}{R_{internal}} = \frac{(50V)^2}{16\Omega} \approx 156W$. At 70V, we have 306W, but again some of the power will be dissipated by the internal resistance.

A relay and manual switch are chosen which can easily handle the high voltages and currents.

Buck Converter

The best way to understand the buck converter is to take Power Electronics with Beat. Failing that, here is a crude explanation.

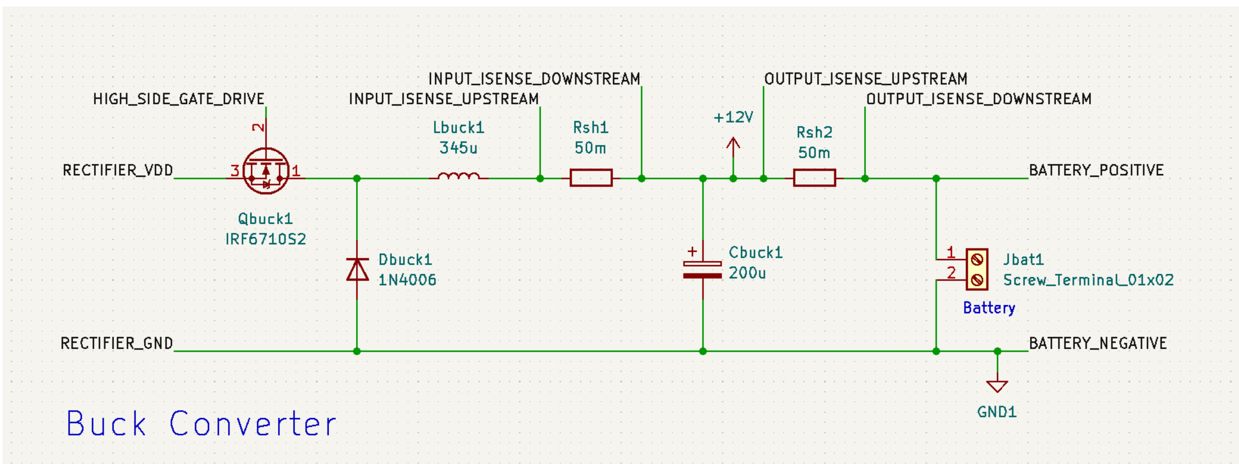


A buck converter consists of four primary components: a MOSFET which acts like a switch, an inductor, a diode, and a capacitor. The MOSFET is driven with a certain duty cycle that determines the voltage conversion ratio (Vout / Vg). It's easiest to analyze if we assume Vout is fixed at the desired voltage; in fact, this is the primary role of the capacitor. We choose a large enough capacitor (in our case, 200uF) so it can smooth out any bumps or discontinuities in the output voltage, of which there will be plenty.

During the time when the switch is on, current flows from the source through the inductor to the output (red arrows), ignoring the diode. Assuming the output is at constant voltage and the MOSFET is a perfect conductor, there must be a voltage drop across the inductor of $V_L = V_g - V_{out}$. The inductor obeys the law V_L = L dI/dt; V_L is constant, so the current increases linearly. It does so until the switch is turned off.

When the switch is off, no current flows from or to the source. An inductor cannot instantaneously change its current, so current continues to flow through the inductor and into the load (blue arrows). Now current flows through the diode to complete the loop. Assuming the output voltage is still constant and no voltage drop across the diode, the voltage across the inductor is now -Vout (it's negative because the left side of the inductor is now at a lower voltage than the right hand side). As it discharges, the current through the inductor decreases by V_L = L dI/dt, where V_L = -Vout. In our case, we turn the switch on again before the inductor gets to zero current (continuous conduction mode, or CCM), but there are plenty of times you want the current to fall all the way to zero and stay there a while before the cycle repeats (discontinuous conduction mode, or DCM). In either case, the capacitor is large enough to ensure the output voltage stays at the average value the entire time.

Another way to think about the buck converter is just a switch turning on and off really fast, the way you can control the brightness of an LED with PWM. The ratio of input to output voltages is controlled by the duty cycle of the switch. Then the inductor and capacitor act as a massive low-pass filter on the output.



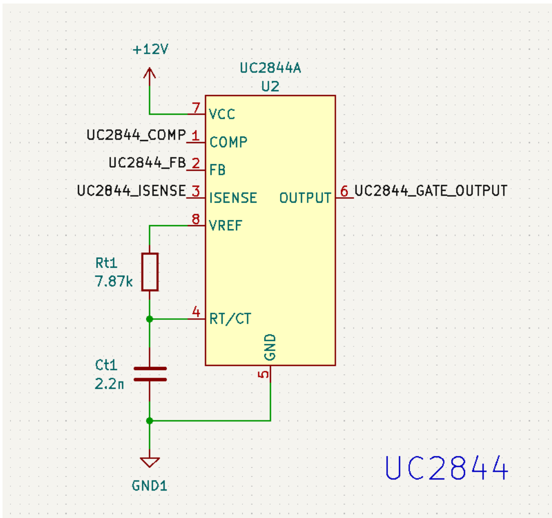
UC2844

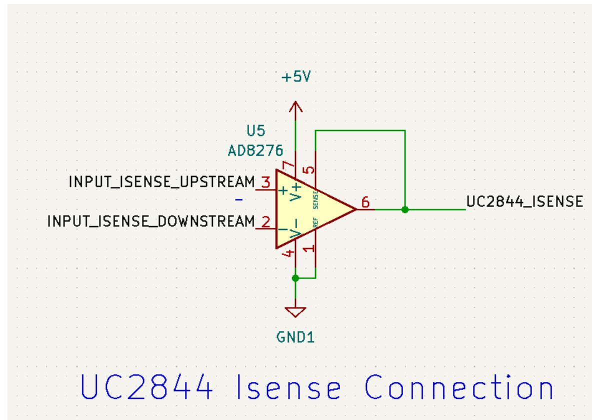
The UC2844 is an integrated circuit for controlling the duty cycle of power converters like this one. We gained a lot of familiarity with it in Power Electronics. Here is the datasheet for the newest version, the UC2844A.

The way it works is it it takes in a voltage at the I_{SENSE} pin corresponding with the amount of current through the inductor—typically, the voltage across a known shunt resistance is used. In my case, since my shunt resistor is not connected to ground on one side, I have to take the voltage difference using a differential amplifier.

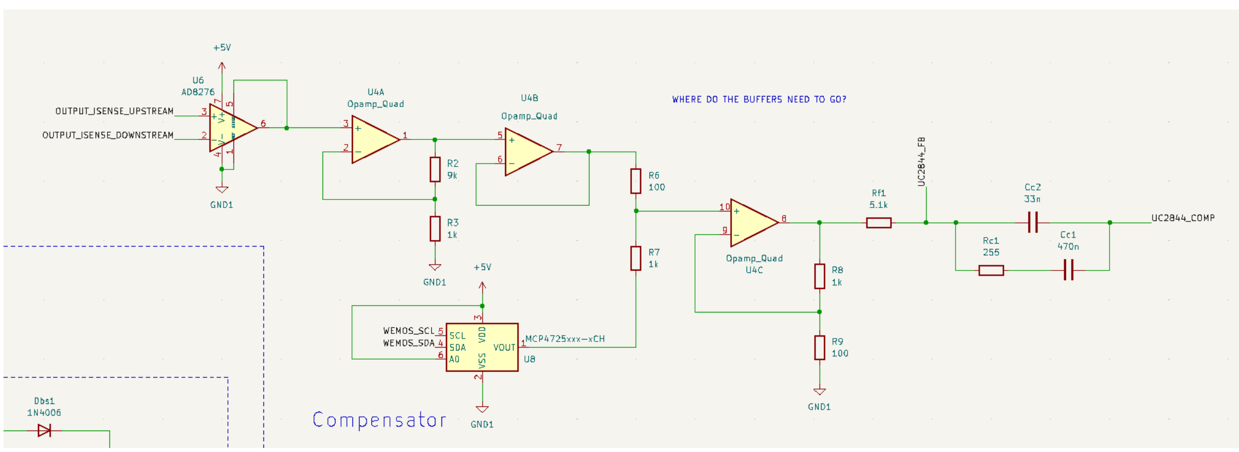
In the last section, I described how the voltage through the inductor increases linearly while the switch is on. Once the I_{SENSE} pin reaches the threshold voltage, the UC22844 turns the switch off and waits a specified amount of time before it turns it back on again. The amount of time it waits is determined by Rt1 and Ct1, and I chose values that will make it turn on every 0.00002 seconds (50kHz). By choosing the threshold for the ISENSE pin at which it turns the switch off, we can set the duty cycle of the switch and thus the voltage conversion ratio.

Instead of setting the threshold voltage directly, I am using a feedback loop to adjust the threshold until it reaches a desired output current into the battery that is specified by the microcontroller. This feedback loop is made stable by the compensator described in the next section.





Compensator



The actual compensator that we learned how to design in Power Electronics is the rightmost part of this circuit, comprising Rf1, Rc1, Cc1, and Cc2. Its job is to ensure that the feedback loop is stable and has a unity gain at steady-state; that is, if we give it a constant target value, then we want it to settle on exactly that value. The rest of this circuit is essentially used to specify the target value.

Specifically, based on the internal design of the UC2844, the target value is defined as the input quantity that makes the value of the UC2844_FB pin equal to 2.5V. If the input quantity is such that the UC2844_FB pin is higher than 2.5V, then it will decrease the duty cycle; and if the UC2844_FB pin is lower than 2.5V, it will increase the duty cycle. I'm using the term 'input quantity' because it really can be anything—in Power Electronics, the input quantity was the load voltage after going through a 4:1 voltage divider; thus, the target value was a 10V load voltage, such that the UC2844_FB pin was 2.5V.

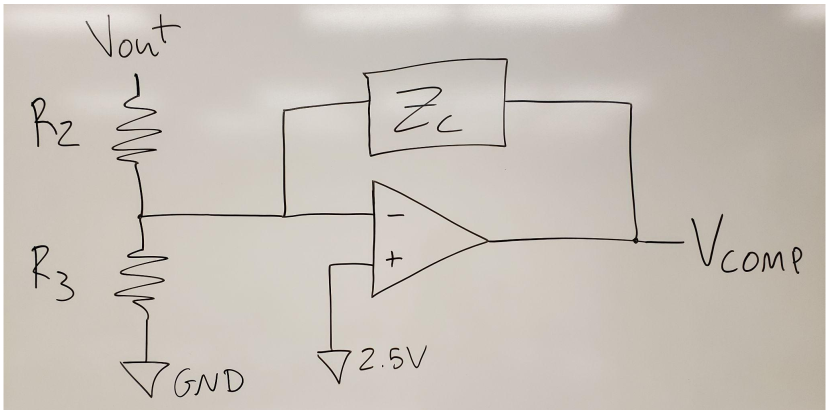
In our case, the target value is a load current of between 0 and 0.5A, as specified by the microcontroller. Here's how:

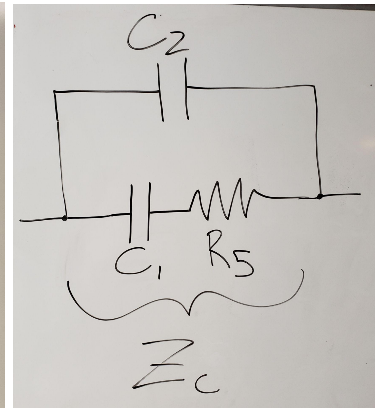
Suppose we have a 0.5A current going into the load, which in this case is our battery. The current passes through a shunt resistor Rsh2, creating a voltage drop between nodes OUTPUT_ISENSE_UPSTREAM and OUTPUT_ISENSE_DOWNSTREAM. Rsh2 is 50mΩ, which means the voltage difference is V = 50mΩ * 0.5A = 25mV. This is picked up by the diff amp and amplified 10-fold by the first op-amp. R5 and R7 form a linear voltage combiner (aka a voltage adder), whereby the top signal is multiplied by a further 10-fold and added to the bottom signal. Assume the bottom signal is 0V for now. So we just had a 100-fold gain on our 25mV, which indeed yields 2.5V for a current of 0.5A.

The bottom signal comes from a digital to analog converter (DAC) controlled by the microcontroller. Suppose we set this to 1.25V. Then the voltage adder will add up to 2.5V if and only if the top signal is also equal to 1.25V. This corresponds to an input voltage of 0.25A. By setting the DAC value to any value between 0V and 2.5V, we can set the target current to be anywhere between 0A and 0.5A. This is useful, because as the battery gets closer to full charge, we will want to reduce the current flowing into it and eventually stop charging it altogether.

Extra notes: I used two stages of amplification each with a gain of 10 instead of one stage with a gain of 100, because I had extra op-amps (I'm using an op-amp quad package), and I figured it might be more accurate. I do not know if all the buffering is necessary or whether I need another buffer on the output before the compensator components. Also, all the op-amps and the diff amp are powered by 5V, which is taken by a linear regulator from the battery. Yet I am using the diff amp to measure voltages which are nominally 12V-14.5V or so. Fortunately, the particular [diff amps] I'm using are rated to measure voltages beyond the rails and significantly above 15V.

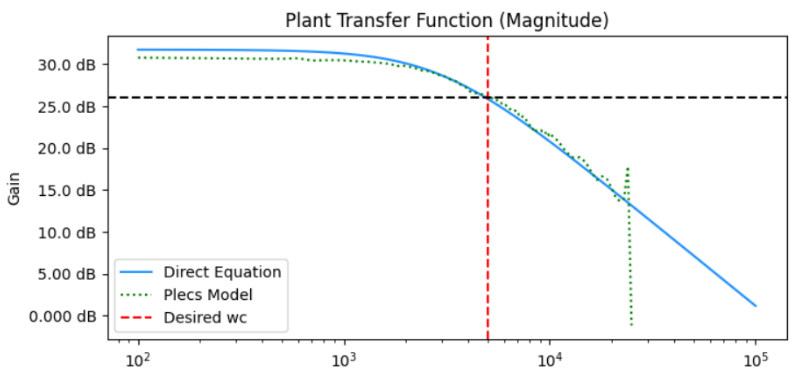
When analyzing the compensator circuit, it helps to look at the functional block diagram of the UC2844. In particular, there is an op amp with UC2844_FB leading to the inverting input, UC2844_COMP leading to the output, and the noninverting input internally tied to 2.5V. It's like the image below but without resistor R3 (this image was taken from a Power Electronics lab). UC2844_FB is the node between the resistors.



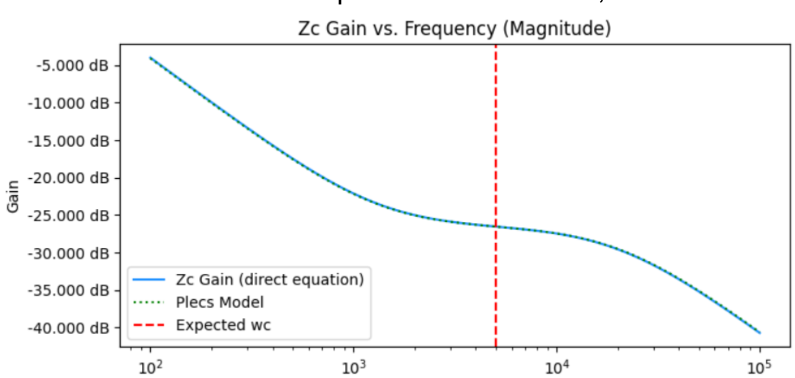


Choosing Compensator Values

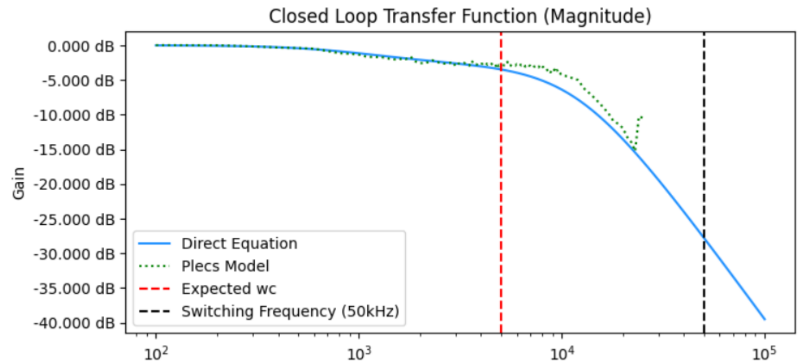
This is the subject of an extensive analysis documented in this jupyter notebook, using this PLECS model and other resources in this folder. The idea is to analyze the transfer function of the plant, which looks like this…



…and make it suitable for stable feedback with no DC offset. This is done by adding another transfer function—the compensator— in series, whose transfer function looks like this…



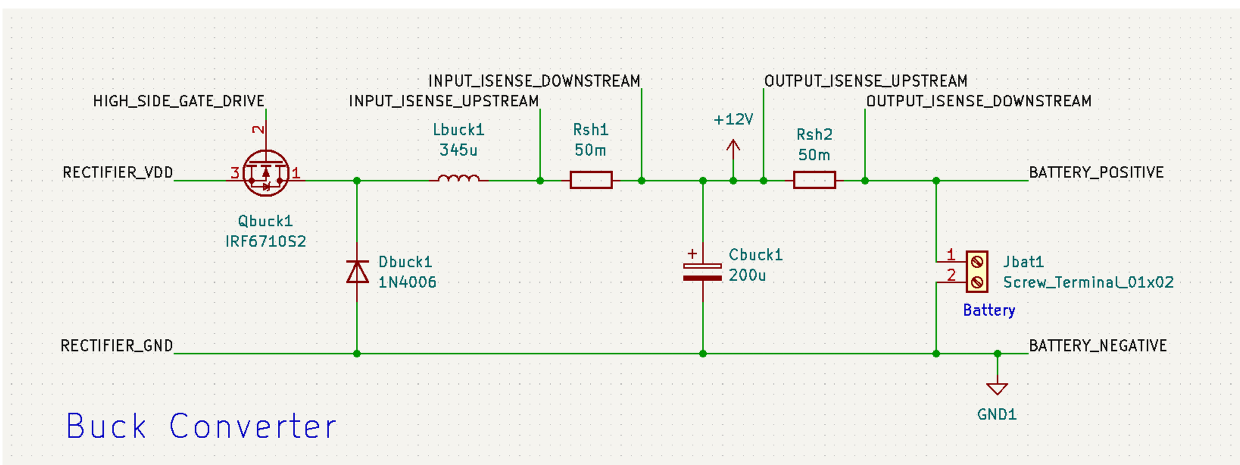
…to produce a net transfer function with infinite gain at DC, zero gain at the red dashed line, and strong attenuation near the switching frequency, such that any fluctuations due to the switching behavior are ignored. After closing the feedback loop, the transfer function becomes…



…which has all the properties described. Here, the black dashed line is the switching frequency, and it has attenuation of around -30dB, whereas at steady-state (0Hz), there is unity gain, meaning the system will reach the desired state if we feed it a steady target.

The PLECS model verifies that the target current can be arbitrarily specified by the microcontroller between 0A and 0.5A, and the output current will settle on that value within half a millisecond with no overshoot.

Placement of the current sense resistors



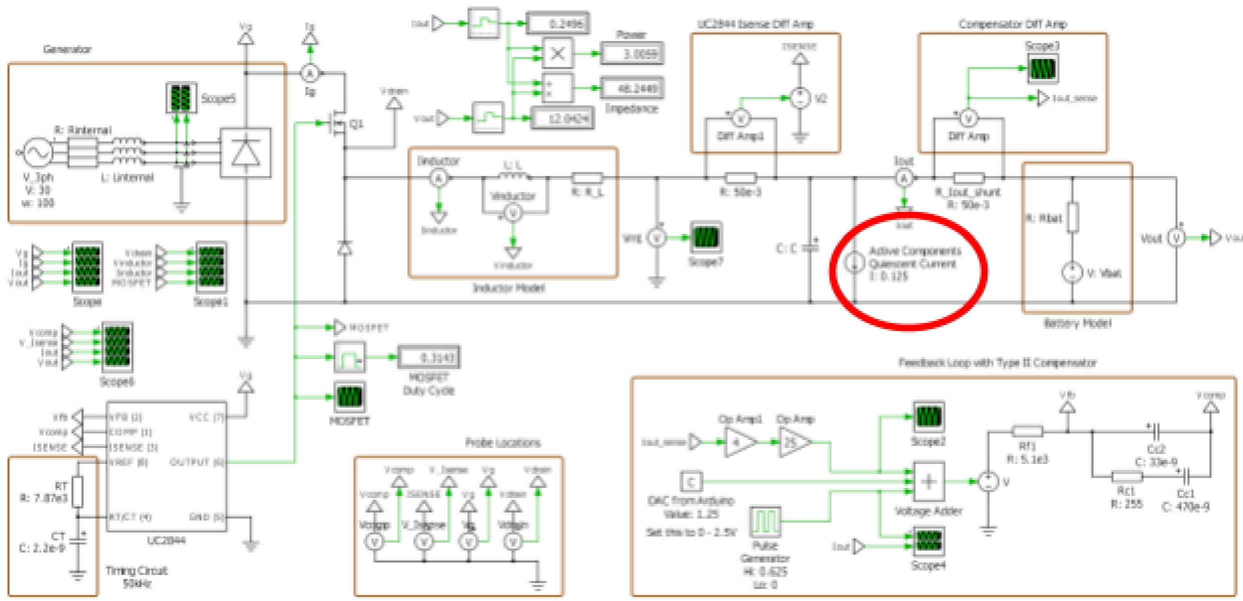
A current sense resistor is essentially a shunt resistor whose resistance is small and well known, such that when current passes through it, the current can be inferred from the voltage drop. In Power Electronics, we placed the current sense resistor downstream of the load, with its negative lead at ground. The MOSFET was also tied through the shunt resistor to ground. This worked because it was easy to measure the shunt voltage, as it was already referenced to ground, and it was easy to drive the MOSFET. Contrarily, my design establishes a common ground between the load (the battery) and the source (the negative rectifier terminal).

I already established that we can use a diff amp to measure the voltage difference across the shunt resistors, even though the nominal voltage is on the order of 12-15V. But why are there two of them, and why these specific locations?

The first shunt resistor, Rsh1, acts as the current sense pin tied to UC2844_ISENSE, which is directly responsible for making the UC2844 turn off the MOSFET when the duty cycle is reached. It is critical that the resistor used for this purpose be placed to the left of the 200uF capacitor. If it is to the right, then some of the current will flow through the capacitor, and it will not reach the current threshold at the proper time. PLECS simulations show that the output is unstable and sporadic when the UC2844_ISENSE resistor is to the right of the capacitor.

The second shunt resistor, Rsh2, is directly in series with the battery, such that the resistor current is equal to the battery current. This is the current sense resistor that is used to feed the compensator and feedback loop. I specify that I want a certain current, say 0.25A, going into the battery, and I want the feedback loop to adjust until I get that amount of current into the battery, regardless of how much current is drawn by the auxiliary components or dissipated by anything along the way.

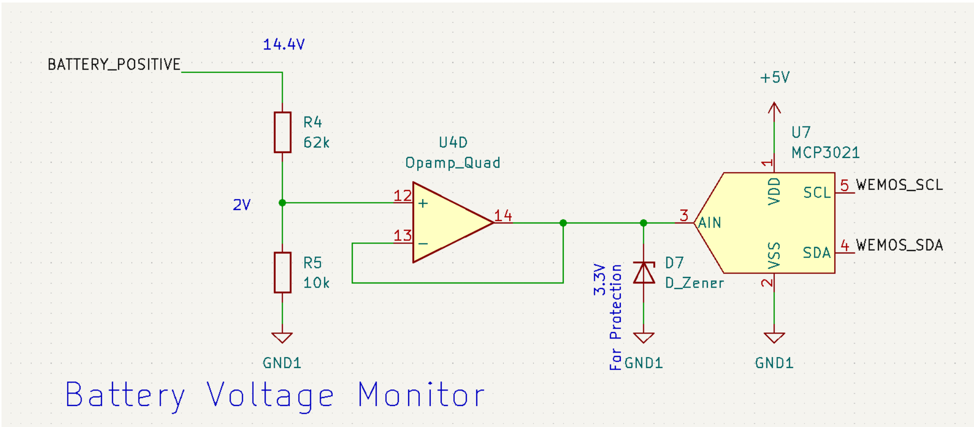
The +12V rail is used for powering the UC2844 and gate driver IC directly, as well as passing through a linear regulator to supply 5V to the microcontroller, op-amps, diff amps, and other components. Notably, the 12V rail is supplied upstream of the second shunt resistor, such that as much current can be siphoned off as needed without affecting the current to the battery controlled by the feedback loop. This is modeled by a 125mA current source upstream of the second shunt resistor in my PLECS model.



The +12V supply rail

It should be noted that the +12V rail symbol used in KiCAD is not actually 12V. Instead, it is the battery voltage which is dependent on its state of charge, and it will be closer to 14.4V or 15V when the battery is nearly charged and there is still some current through Rsh2.

Battery voltage monitor



The state of charge of the battery can be monitored based on its voltage. [Sources] recommend charging the battery at a high current up to 14.4V or 14.6V, and then reducing the current for the remainder of the charging cycle. Thus, it is important to know the battery voltage accurately and precisely.

For this, a simple resistive divider is used to divide the 12V-14.6V voltage down to the 0-3.3V range so it can be accessed by an analog to digital converter (ADC). The ADC is sensitive to overvoltage at the input, and the voltage should never need to go above 3.3V, so I included a zener diode for safety. It is probably not necessary.

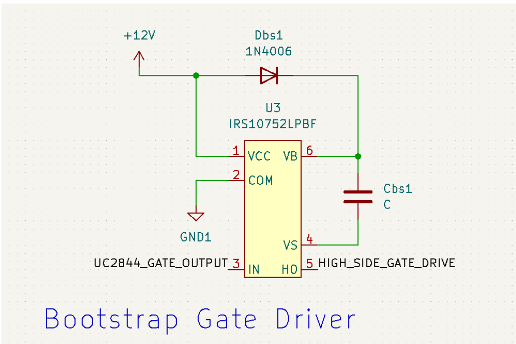
I am using an external ADC and DAC. These are usually available onboard a typical microcontroller using the analog pins, but I don't trust those to be accurate all the time, and the microcontroller I am using only has one analog pin anyway. The ADC and DAC are accessible to the microcontroller via I^2C communication. I included 4.7kΩ pull-up resistors on the I^2C lines.

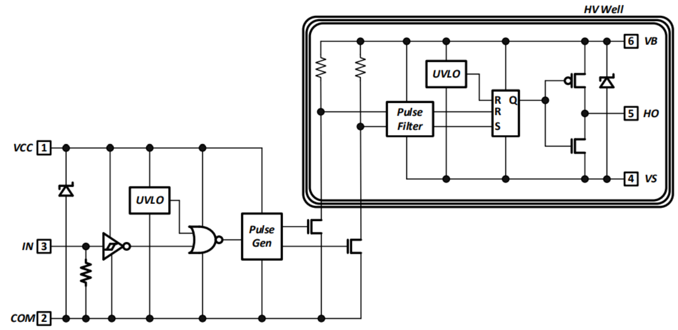
Bootstrap Gate Driver

I mentioned earlier that I moved the MOSFET of the buck converter circuit from the low side (downstream of the load and closest to ground) to the high side (upstream of the inductor) for the purposes of a shared ground between the battery and the rectifier. The low side MOSFET could be easily switched using the UC2844 gate drive output, which in this case would be nominally 12V. Since the source was tied close to ground, the gate-to-source voltage could easily be made high or low.

Using a high-side MOSFET presents the tricky problem that when the MOSFET is conducting, its source is nearly at Vg, which is close to 50V. In order to drive the gate, we need a gate driver capable of delivering at least 60V to the gate; or rather, 10V between the gate and the source.

Fortunately, we have the bootstrap circuit which accomplishes exactly this, along with a purpose-built IC to serve as the high-side gate driver.

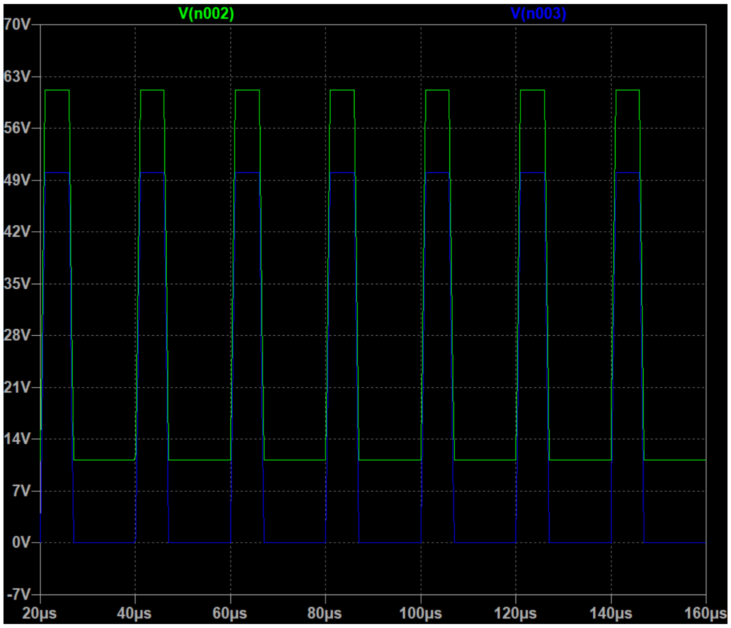


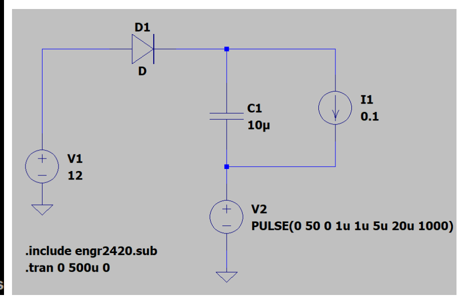


It works by maintaining an almost constant voltage across the capacitor due to the diode, with the negative terminal of the capacitor connected to the source of the MOSFET. Consider the relation I = C dV/dt, where I can only flow forward through the diode. The capacitor cannot decrease in voltage except through parasitics or by discharging into the gate capacitor on the MOSFET. As the voltage degrades over time, it will be replenished every time its voltage drops below 12V, which happens every cycle of the buck converter when the switch is turned off and the source of the MOSFET is tied close to 0V. When the MOSFET is turned on and its source rises close to 50V, the voltage across the capacitor means the gate drive pin is around 60V as desired.

The image on the right shows the block diagram of the gate driver IC and how it converts a low logic level voltage into a high voltage output.

I have verified in LTspice the behavior of the diode-capacitor circuit and confirmed that it can boost a high voltage level even higher for short periods of time, even when a small parasitic current is constantly being pulled from that node.





Microcontroller

The PCB will have two rows of female sockets where an ESP8266 board can be plugged in. The ESP8266 will be powered via the Vin pin using the 5V linear regulator on the PCB. It will also have breakout pins for attaching other sensors. I am very comfortable using the ESP8266 and have found it to be very reliable. It also has WiFi connectivity, which could be useful.

The main connections to the microcontroller are the I^2C pins for the DAC and ADC, as well as one digital pin for controlling the relay. It will be programmed to monitor the battery voltage using the ADC and to follow a specified charging current curve for the battery. If dangerous conditions are detected, it will shut off power to the circuit using the relay at the buck converter's input.

When the battery is attached, it should be able to feed the 5V regulator and continue to power the microcontroller, allowing it to reconnect the relay if it is given the signal to resume charging.