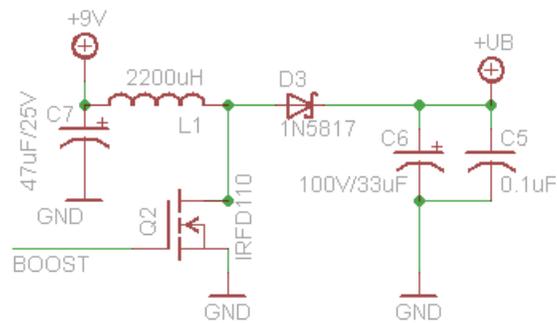


DIY DC/DC Boost Calculator

Created by Tyler Cooper

Tube 30-60VDC boost supply



Last updated on 2018-08-22 03:36:04 PM UTC

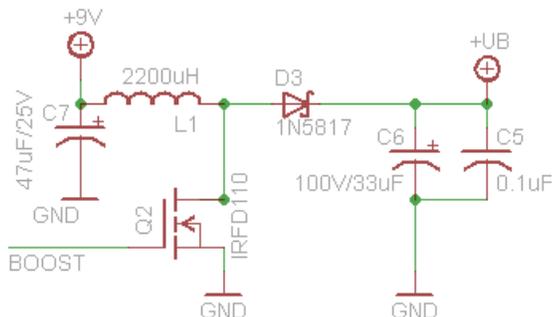
Guide Contents

Guide Contents	2
Overview	3
The Calculator	4

Overview

For many small projects, its cheaper and easier to DIY a boost converter than to buy a specialty chip. DIY converters are usually not as efficient but they're quick & cheap!

Tube 30-60VDC boost supply



The above schematic section shows how I designed a 30-60V vacuum fluorescent tube display driven from a microcontroller pin.

Tubes such as VFDs, Nixies, Decatrons, etc require high voltage to light the gas in the tube. In order to reduce cost, we use a microcontroller to make a boost converter and avoid paying \$5 for a separate chip. We can do this because we don't need a precision output and the current draw is mostly constant. The boost regulator is run open-loop there is no feedback resistor divider as it isn't necessary as long as the input voltage is within a reasonable range

The microcontroller runs at 8MHz so the 8-bit PWM output is 31250 Hz. The inductor and output capacitor is calculated below. The diode is a standard Schottkey type, but make sure you specify one that can handle the full voltage difference and peak current. The switch just has to be able to handle the max voltage plus some for safety. Note that this design is meant for 'static' output currents, not for variable current draw designs. There is no feedback and its very approximate! This is not for precision electronics!

The boost circuit works by connecting the power inductor **L1** to ground that current can flow through it by turning on **Q2**. After a little bit of time, we disconnect the inductor from ground (by turning off **Q2**) this means that there is no longer a path for the current in **L1** to flow to ground. When this happens, the voltage across the inductor increases (this is the electric property of inductors) and charges up **C6**. When the voltage increases to the level we want it to be (30V+) we turn on **Q2** again which allows the current in **L1** to flow back to ground. If we do this fast enough, and **C6** large enough, the voltage on **C6** is smoothed out and we get a nice steady high voltage

The timing of turning off/on **Q2** allows us to modify the output voltage. Normally there is a feedback resistor to the microcontroller but it is not here because we are running it open-loop. To drive **Q2** we use the PWM output from the microcontroller and adjust the duty cycle to vary brightness.

These sorts of designs can be easily made with a 555, once you have the PWM output, connect it up to **Q2**!

The Calculator

For this simple calculator, enter in the frequency, voltage ranges and current ranges and the duty cycle, inductor and current requirements will be displayed!

Frequency	<input type="text" value="31250"/>	Hz	This is the boost converter frequency. For microcontrollers its often the CPU clock / 256
Min Vin	<input type="text" value="9"/>	V	The lowest expected input voltage
Max Vin	<input type="text" value="12"/>	V	The highest expected input voltage
Min Vout	<input type="text" value="30"/>	V	The lowest desired output voltage
Max Vout	<input type="text" value="60"/>	V	The highest desired output voltage
Iout	<input type="text" value="0.020"/>	Amps	Output current draw
Vripple	<input type="text" value="0.1"/>	V	Maximum allowable voltage ripple

Calculate!

Min. Duty Cycle	$D_{min} = 1 - (V_{imax}/V_{omin})$	<input type="text"/>	<input type="text"/>	%
Max. duty cycle	$D_{max} = 1 - (V_{imin}/V_{omax})$	<input type="text"/>	<input type="text"/>	%
Min. Inductor size	$L > D * V_{in} * (1-D) / (freq * 2 * I_{out})$	<input type="text"/>	<input type="text"/>	uH
Peak inductor current	$I_{pk} = (V_{inmax} * D)/(f * L)$	<input type="text"/>	<input type="text"/>	A
Minimum capacitor	$Cap > I_{out} / (V_{ripple} * freq)$	<input type="text"/>	<input type="text"/>	uF
Minimum Schottky diode	$V_{breakdown} \geq V_{outmax} \ \& \ I_{diode} \geq I_{pk}$	<input type="text"/>	<input type="text"/>	V A