Pocket Tutor

Build a handheld version Of the Morse Code Tutor

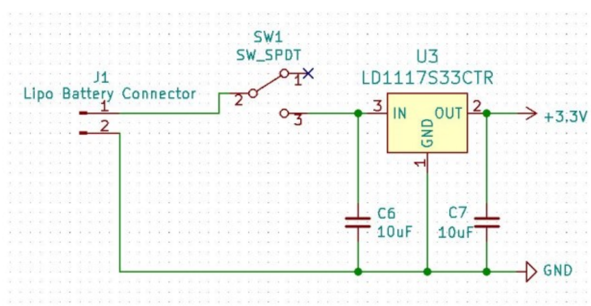
Part 2: How not to design A power supply.

Bruce E. Hall, W8BH

Almost all electronic projects require a power supply. In the Morse Code Tutor, we used a pre-assembled step-down module for the main power supply, with additional regulation on the microcontroller module. It worked well with 9VDC input, but not with lower-voltage batteries. In our Pocket Tutor we need a smaller circuit that optimizes battery use and provides battery recharging. I will show you schematics for each revision of the power supply that I built, starting with my first and simplest approach. Continue reading if you want to read about ~~my mistakes~~ what I learned.

Power supply woes.

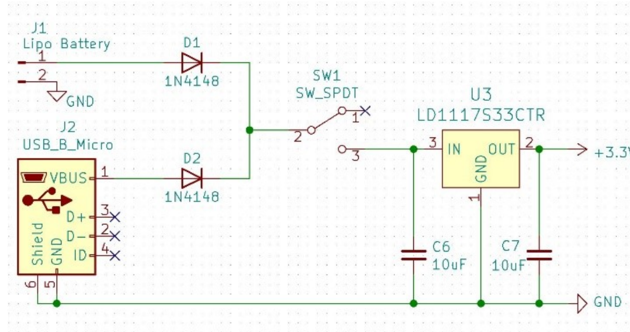
The power source has to be small and light, so I settled on a single LiPoly 3.7V cell. From my battery experiments, I knew this type of battery would give hours of operation on a single charge. Unfortunately, these cells have up to 4.2V when fully charged, so directly powering a 3.3V circuit does not work. My initial design was simple: use a 3.3V regulator, and add input and output capacitors for stability.



I made it, and it worked. Occasionally the output voltage drifted below 3.3 V, but not by much, so I ignored it.

What if the battery dies? It would be a good idea to allow DC input from another source, like a USB cellphone charger. So, I added a USB input using two diodes. The diodes isolate the power sources from each other. Without diode isolation, current flows from the 5V USB source into the 3.7V Lipo battery, potentially overcharging and destroying it.

I made the modifications, confident of success. I was only adding two diodes. But It didn't work at all. The output of the regulator was far below 3.3V. What went wrong? Answer: too many voltage drops. To make a long story short, each diode drops the voltage across it by 0.7V. So, if the battery is at 3.8V, the voltage at the other end of the diode will be 3.8V – 0.7V = 3.1 V. The voltage regulator will reduce the voltage even further. Fortunately, there is a special type of diode, called a Schottky diode, which reduces the forward voltage drop from 0.7V to 0.3V. I replaced my diodes with Schottky's, and my voltage output improved. Once again, though, I ignored the occasional downward voltage drift. Here is a photo of my version 1 board, sporting a beefy LD1117 at the top, flanked by the C6 and C7 bypass caps.

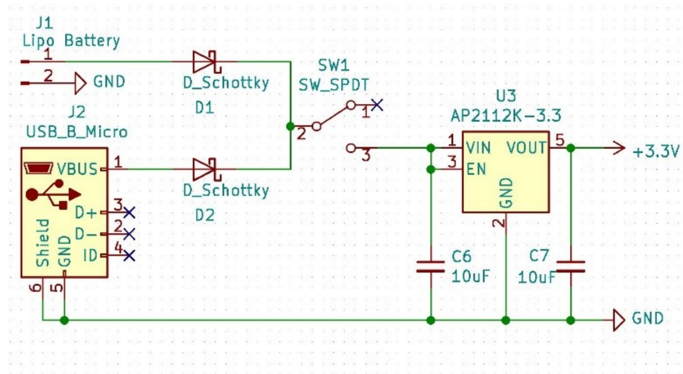


Not all regulators are created equal. I was using the LD1117, a standard, go-to regulator like the 7805. The chart below compares several 3.3V regulators and some of their important characteristics.

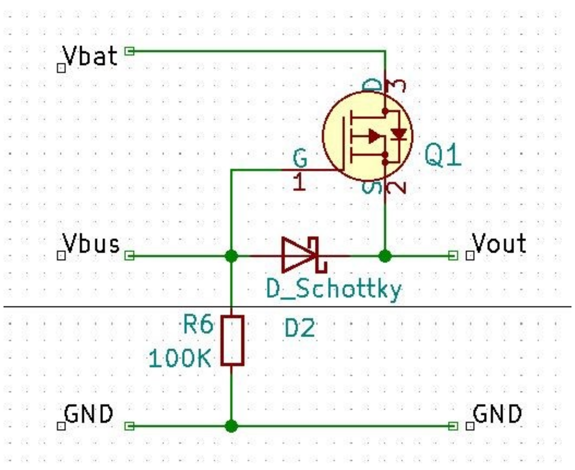
Regulator	LD1117S33	AP2112-3.3	LM3940	TC1262-3.3VDB	TPS73633
Vin Min	no spec	2.5V	4.5V	2.7V	1.7V
Vin Max	15.0V	6.0V	5.5V	6.0V	5.5V
Output Current	950mA	600mA	1000mA	500mA	400mA
Vdo @ 200mA	1000mV	125mV	150mV	150mV	75mV
Digikey	LD1117S33CTR	AP2112K-3.3TRG1	LM3940IMPX-3.3	TC1262-3.3VDB	TPS73633DBVR (Mouser)
Price	$0.41	$0.47	$1.60	$0.51	$1.95

Look at the line for voltage drop-out (Vdo). Vdo specifies the smallest difference between the input and regulated output voltages. Therefore, the *minimum* input voltage is 3.3V + Vdo. For the LD1117, you must supply 3.3V + 1.0V = 4.3 V in order to a regulated 3.3V output. Anything less will result in a lower, non-regulated output. For our application, where there is little voltage difference between the battery and the 3.3V output, having a very low Vdo is important. The LD1117 was a poor choice! Time for a new regulator. I chose the AP2112, a low-cost device with very low Vdo that still has decent (600ma) current output.

The updated design swaps the LD1117 for the AP2112 regulator, and uses Schottky diodes in place of standard silicon ones. With this design we get regulated 3.3V from as little as 3.3 + 0.3 + 0.125 = 3.725 volts battery input. After 3 tries, I finally have a usable supply. Unfortunately, we lose regulation below 3.725 volts, and there is still plenty of battery life left at that level. Can we make it be better?

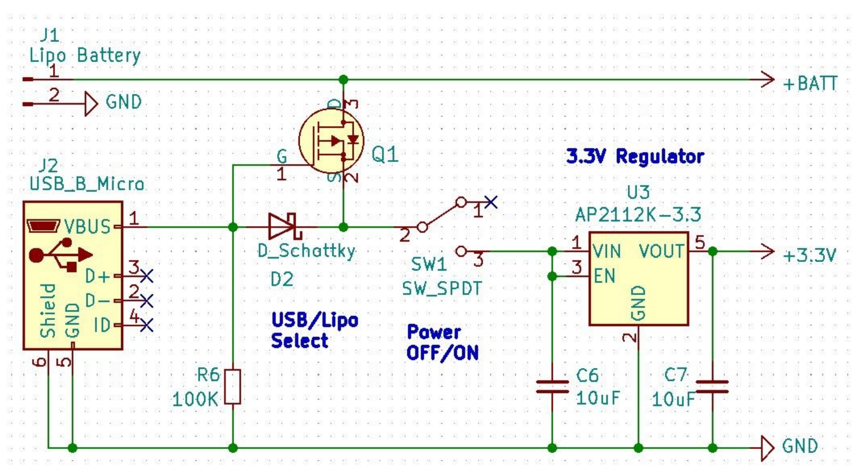


Surely others have encountered a similar situation, trying to regulate a single 3.7V Lipo battery to 3.3V out. It didn't take long to find plenty of examples and learn a few more tricks. For example, look at this slick use of a MOSFET, adapted from Adafruit's feather board:



This is a P-channel MOSFET, which turns on when its gate goes low. If there is no Vbus (USB) input, the gate is pulled low, the MOSFET switches on, and Vbat (the battery) connects with very low resistance to the regulator. The 0.3V penalty of using a Schottky diode on the battery input is completely eliminated! Note that we still need a Schottky (D2) for the Vbus input, but for the 5V supply this is not a concern: 5.0V – 0.3V = 4.7V is more than enough input voltage for the regulator. When USB Power is present, Vbus > Vbat, the MOSFET switches off, and backflow to VBat is prohibited by the MOSFET's internal diode. Clever! I added the P-channel MOSFET to my power supply.

Now we have a power supply capable of 3.3V regulated output with battery input as low as 3.425V. And, since our battery is nearly exhausted at this level, we are wasting very little battery capacity. Here is the updated schematic.



It works very well.

The final addition to this power supply is a battery charging circuit.   There are thousands of choices: to see these choices, go to your favorite electronic distributor and search for PMIC (power management integrated circuit).   Since we are using a cell with build-in protection, we only need one with a charge management controller.   Microchip Technology makes the MCP73831, which is ideally suited for charging via USB power. The chip provides a status output line to show charging status.   The rate of charging is set by a programming resistor.

The charging rate should reasonably match the size of the battery.   In our application, batteries of 300-1500 mAh are small in size and are sufficient for hours of operation.   These batteries can be safely charged as fast as 1C. For example, using a rate of 1C, a 1000 mAh battery is charged at a current of 1000 mA.   This charging rate is well above the capacity of our small charger IC, however.   I chose a charging rate of 100 mA, which is 0.1C for a 1000 mAh battery and 0.3C for a 300 mAh battery. 100 mA is a slow rate of charge for larger batteries, but safe for tiny lipo cells down to 100 mAh.

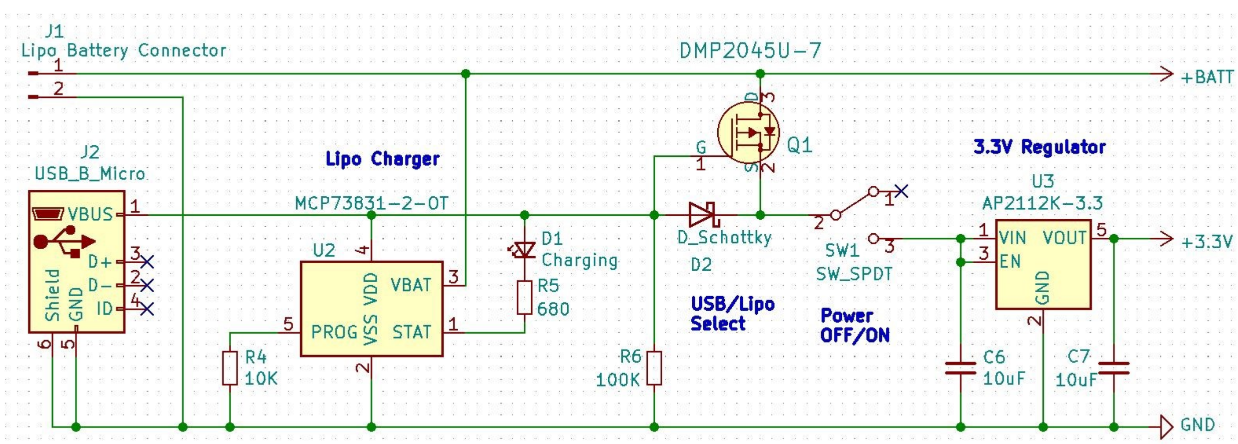
The programming resistor needed is determined by the following equation:

R (in K-ohms) = 1000 V / I(mA).

Therefore, we need a 1000 V/100 mA = 10 K resistor.

Pin 1 of the MCP73831 is the status line. This pin goes low when the battery is charging, and can sink up to 25mA of current.   Let's add an LED and current limiting resistor to visually show when the battery is charging.   What resistor value should we use? It is helpful to know how bright your LED is at different current levels. For example, the Chanzon 0805 LED I am using, has a forward voltage drop of about 2.8V and is bright at 3mA. If the voltage drop across the LED is 2.8V, the voltage drop across the resistor is 5.0 – 2.8 = 2.2V. Using Ohms law, to get 3mA of current you should use a resistor of 2200 mV/3mA = 733 ohms.   I chose 680 ohms, the closest standard value in the E12 series. A 510-ohm resistor will make the LED very bright, and a 1K resistor less bright. See my article on LEDs for more information. Side note: you could also tie the status pin to a microcontroller input, so that the application also knows when the battery is charging.

Here is the finalized power supply schematic, including the battery charging circuit. The programming resistor is R4.



While experimenting with this circuit, I've managed to let the magic smoke out of several parts. The first part to go is usually the MCP73831. You should suspect the MCP73831 if you notice significant (~20mA) battery drain when the switch is off. Normal current drain at switch-off should be less than 20 uA.

The next part of the project is the microcontroller circuit. Stay tuned. 73, Bruce.