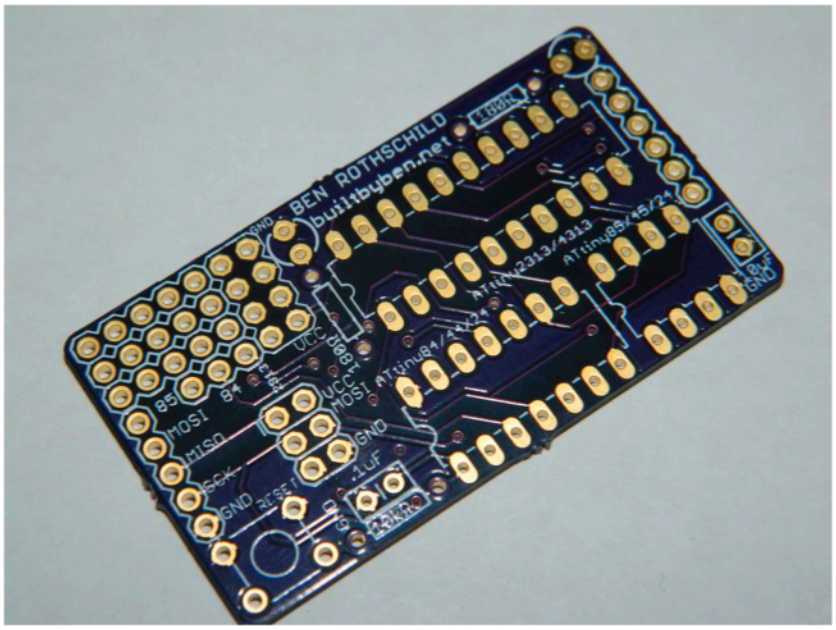
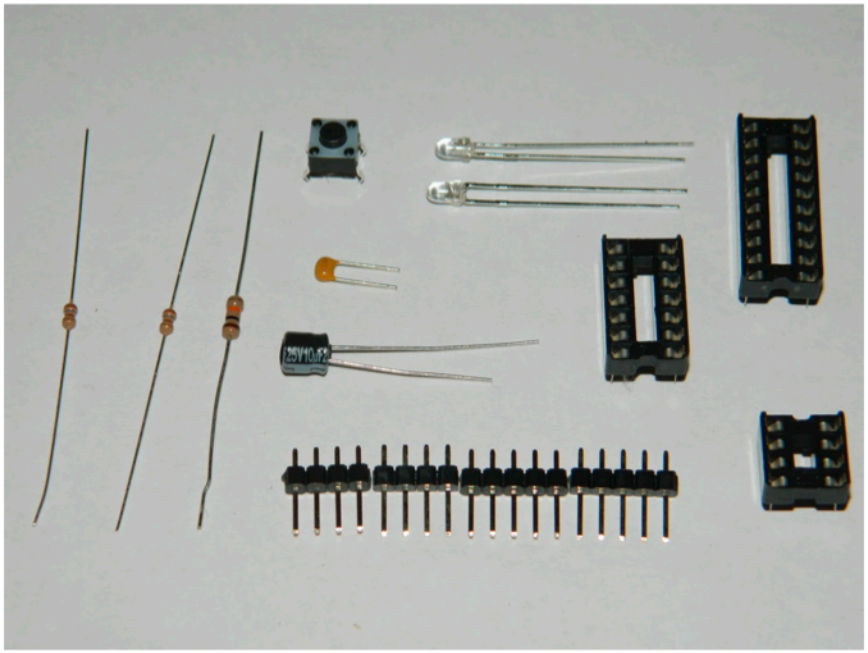
Chipper Assembly Guide



Kit Parts:

- (1) Chipper PCB
- (1) 10KΩ Resistor
- (2) 180Ω Resistor
- (1) DIP 8 Socket
- (1) DIP 14 Socket
- (1) DIP 20 Socket
- (1) .1uF Capacitor
- (1) 10uF Capacitor
- (2) Green 3mm LED
- (1) Push Button
- (1) Strip of 20 header pins



Tools:

- Soldering Iron
- Solder
- Flux pen (optional)

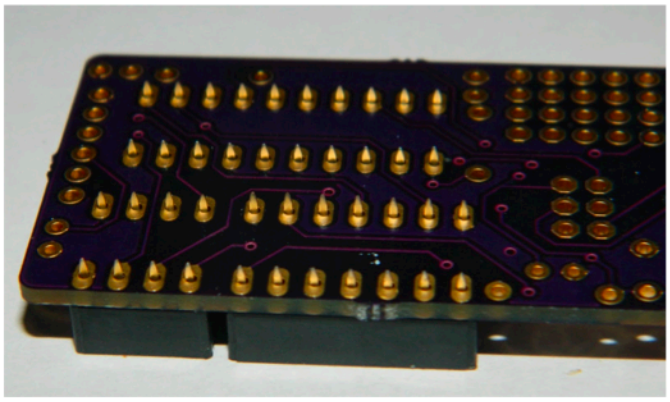
ATtiny Add-on Chips:

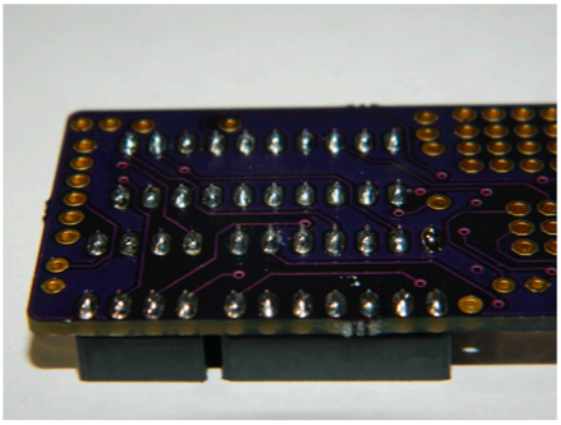
- 1x ATtiny84
- 1x ATtiny85
- 1x ATtiny2313

Soldering

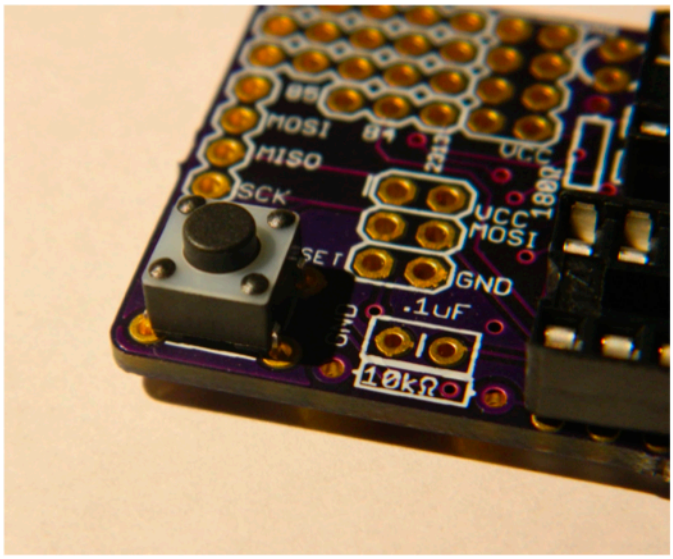
This board was designed to be relatively easy - even for a beginning hobbyist to solder together - because all of the parts are through-hole and the parts are spaced apart.

The first thing to solder on are the three DIP (dual in-line package) sockets. These are what will hold the chips in place while you program them. The benefit of using DIP sockets in a project is that you do not have to unsolder your chip from the board in order to reprogram it. If you have one, use the flux pen on all of the pads on the underside of the board. Then, on the underside of the board, solder all of the legs to the appropriate pads.

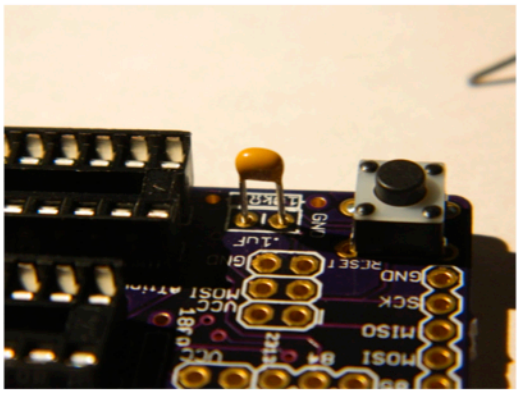




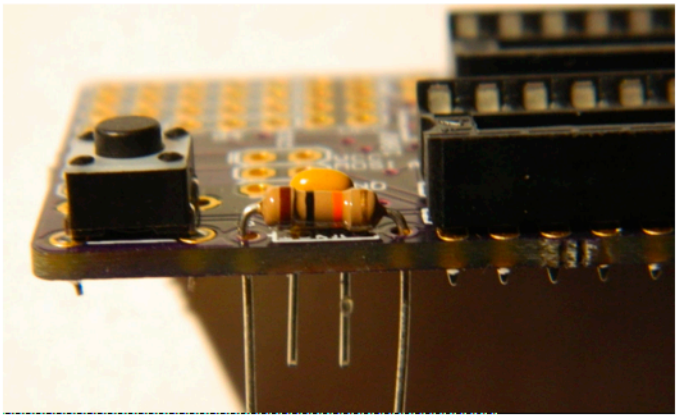
Next, press the four legs of the reset button through the board and solder them on as well.



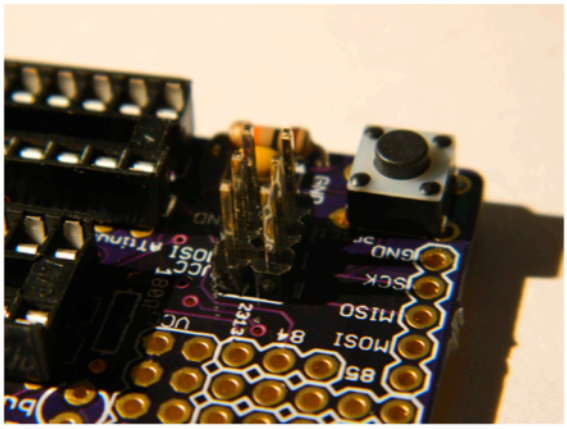
Next up is the .1uF capacitor, which goes to the right of the rest button. Unlike the 10uF capacitor which will we attach later, this is **not** polar so you can solder it in either direction.



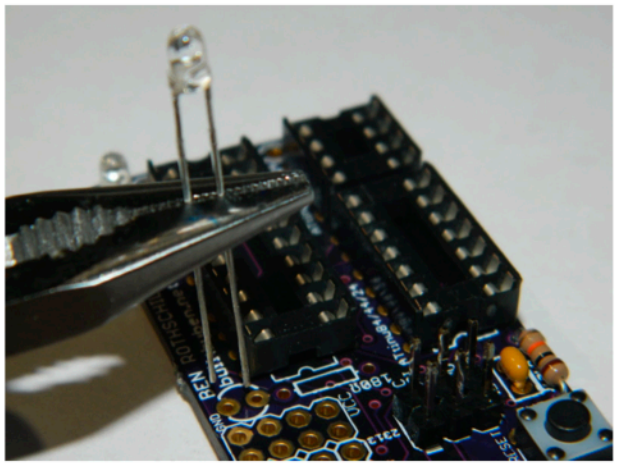
Since we are working on this portion of the board, it would also be a good time to add the 10kΩ resistor. It can be identified by the Brown Black Orange Gold color code. Additionally it is the biggest resistor of the 3 in the kit. You can solder the resistor in either direction.



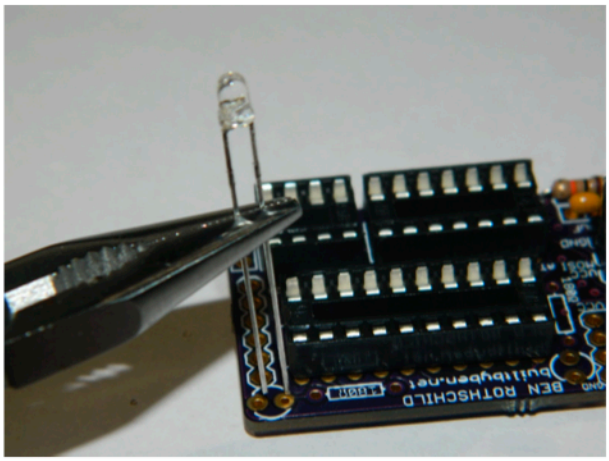
The 6-pin AVR header provides another port for programming your ATtiny chips. Instead of using an Arduino as the programmer, you can use something like the USBtinyISP from Adafruit, which I will later demonstrate in the programming section. To create the header, break of two 3 pin sections from the headers pins and solder them to the board.



Next up are the Green LEDs. LEDs **are polar**, which means that the long leg (anode) is always attached to power and the short leg (cathode) is attached to ground. In our case this means that the long leg is always attached to the side closest to the outline of the resistor.

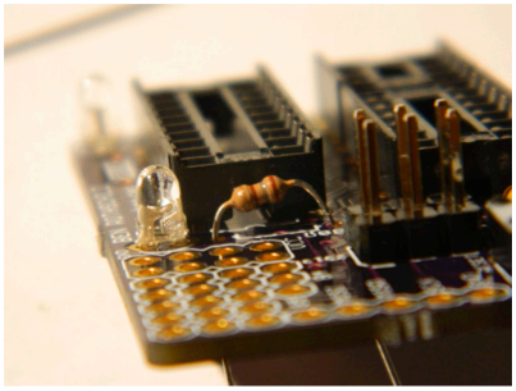


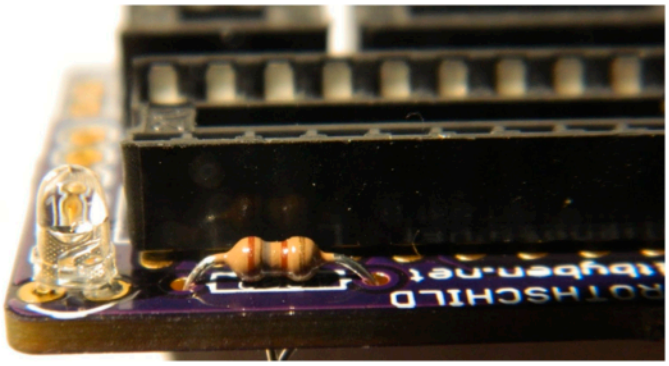
(long leg on the right)



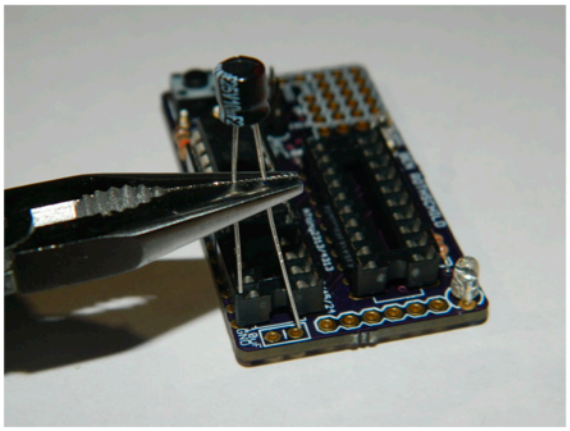
(long leg on the right)

We also need to attach the 180Ω (Brown Grey Brown Gold) resistors in order to limit the amount of current supplied to the LEDs so that they do not burn out. The LEDs that we are using have a forward voltage of 2.1V and forward current of 20mA. Since the Arduino and USBtinyISP will supply a voltage of 5V the resistance is equal to $\frac{5-2.1}{.02} = 145\Omega$. I choose to use a higher resistance to be on the safe side and because no one wants a super bright LED shining in their face when trying to program the chips.





Then comes the 10uF capacitor which **is polar**. To identify the correct alignment look for the grey strip running along one side of the capacitor. This side of the capacitor should be placed into the hole closest to the GND text in the bottom right hand corner of the board.



Time for the last part! The header pins that will attach into the Arduino. Break off a 6 pin and 5 pin portion from the strip of header pins and solder on the 6 pin portion on the right and 5 pin section on the left side of the board. Congratulations! Everything is fully assembled and now it is time to get programming!