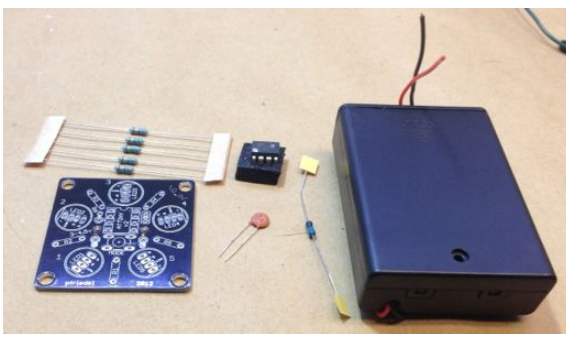
Tiny RGB Build Instructions

PARTS LIST:

Quantity	Name	Location	Description
1	Circuit Board	-	
5	10mm Common Anode RGB LEDs	LED1-LED5	
5	68 Ω resistors	R1-R5	blue - gray - black - silver - brown
1	10k Ω resistor	R6	brown - black - black - red - brown
1	100 nF capacitor	-	Labeled “104”
1	8 pin socket	ATTINY	
1	ATtiny85	ATTINY	ATTINY85-20PU
1	Tact Switch	MODE	6 x 6 x 15 mm
1	3 x AA Battery Box		

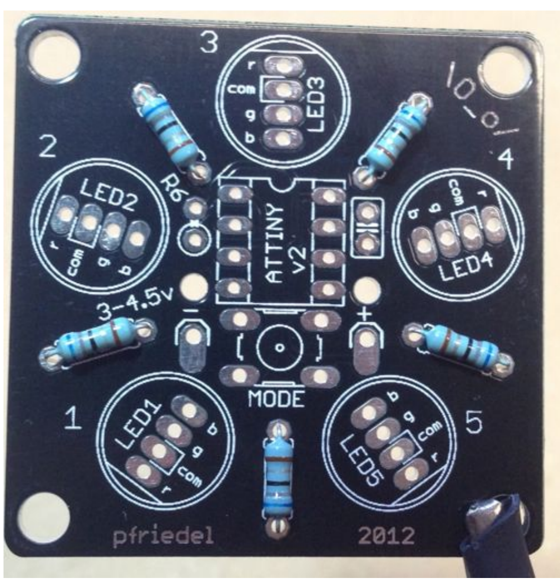
BUILD INSTRUCTIONS:

Please familiarize yourself with all of the instructions before you begin as there are some helpful hints scattered throughout the project that help with later steps.



1. The contents of the kit - the circuit board, five 68 Ω resistors on a strip, a solitary 10k Ω resistor, the 100 nF capacitor, the microcontroller in a piece of foam and the battery box. The LEDs, socket, and tact switch are all inside the battery compartment. If you can’t find a part, open up the battery box and inspect it carefully - I may have changed how I pack everything, but it will all be in the static bag.

There might be slight cosmetic differences to the board between production runs, but the part locations should stay consistent. The general philosophy of what order the parts are installed generally goes from shortest to tallest. Since the resistors are the shorted items in the kit, let’s start with them!

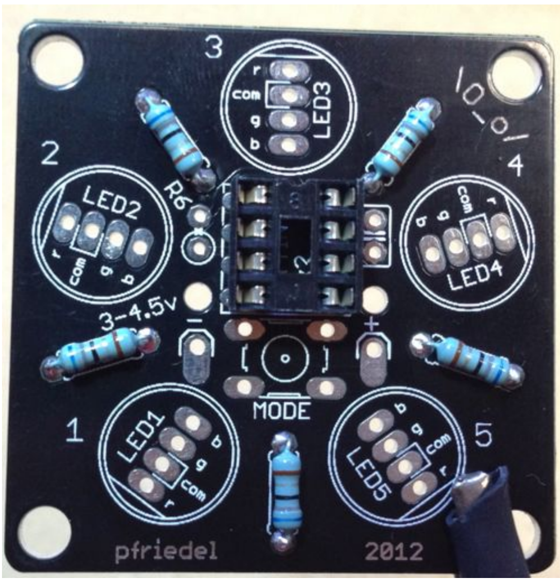


2. First insert all five 68 Ω resistors in positions R1 through R5 as shown. Flip the circuit board over and solder them and clip the excess leads once they’re soldered. Resistors don’t have any polarity, so you can insert them in either direction. You can solder them one by one if you’re more comfortable soldering them that way.

The 68 Ω resistor looks like this:

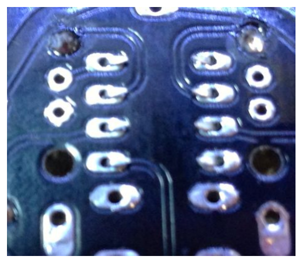
These resistors act as current limiters for the LEDs. If you swap them for ones with smaller values, the first thing that tends to happen is the microcontroller goes into a constant state of reset and nothing happens or the system flickers and restarts continually. If you let it happen for long enough, you could also burn out the microcontroller or possibly damage an LED. This is voltage sensitive: If you intend on using this from a 3v source, you can get away with smaller resistors - 27 Ω is comfortable for 3v for example. 68 Ω is plenty for 4.5v and what is included in the kit.

3. Open the battery box and remove the parts. You'll need the 8 pin socket for the next step, so retrieve it and set the other parts aside.

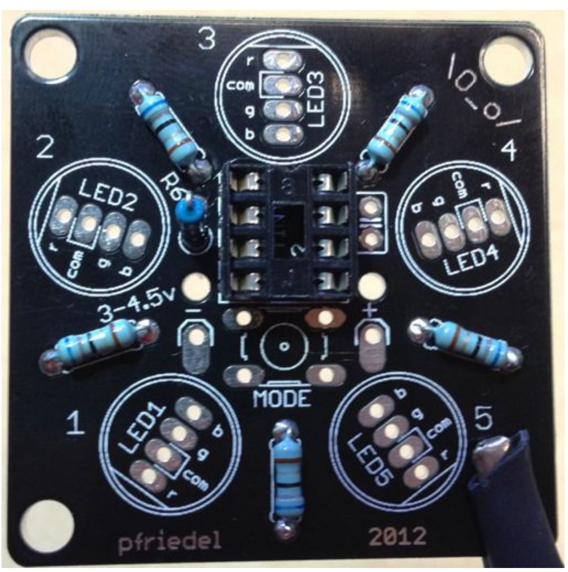


4. Insert the 8 pin socket in the spot labeled ATTINY. Make sure the little cutout in the socket matches the cutout on the silkscreen

Helpful hint: While it isn't strictly necessary, I find that it helps to fold over the socket legs on the other side of the board. Since the power lines run right along here, it helps to reduce the number of points that can stab through the power leads.



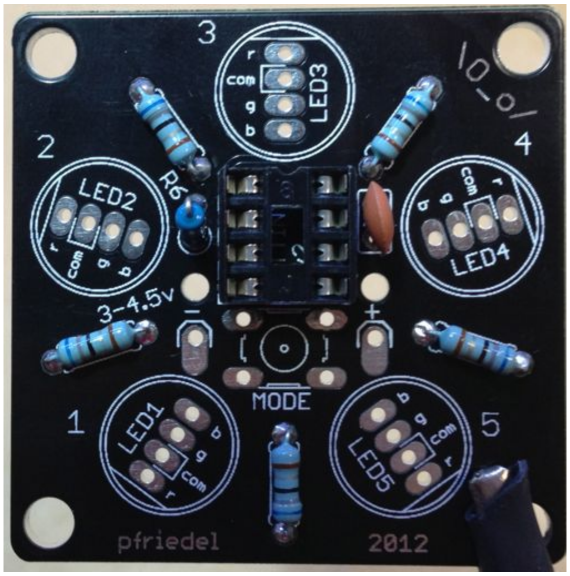
Solder the socket in place. Since the leads are so short, you usually don't need to trim them, and if you fold them over you definitely won't be able to.



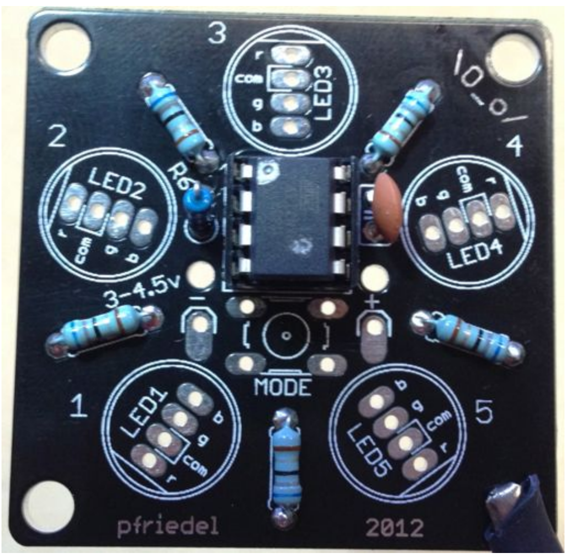
5. Insert the remaining 10k Ω resistor in the spot marked R6. You'll have to fold the resistor in half like so to get it into the holes.

Once you have it in place, solder it and clip the leads again. It doesn't matter if you fit the resistor body up against the circle on the silkscreen, that's merely an aesthetic choice.

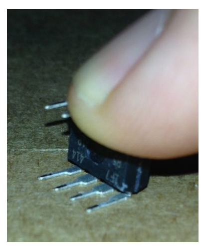
This resistor acts as a pull-up for the reset pin and helps prevent electrical noise from generating an unintended reset.



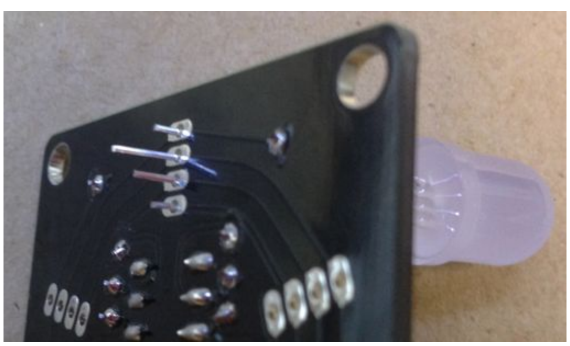
6. Insert the 100 nF capacitor marked “104” in the holes on the opposite side of the socket. Solder it into place and clip the leads. Ceramic capacitors like this aren’t polarized, so it doesn’t matter which lead goes into which hole. This capacitor acts as a noise filter from the high frequency noise that the circuit can generate. Each LED will get turned on and off at a high rate, and every time the LED is turned on and off, the current draw from the battery is affected - this capacitor acts as a buffer for those changes.



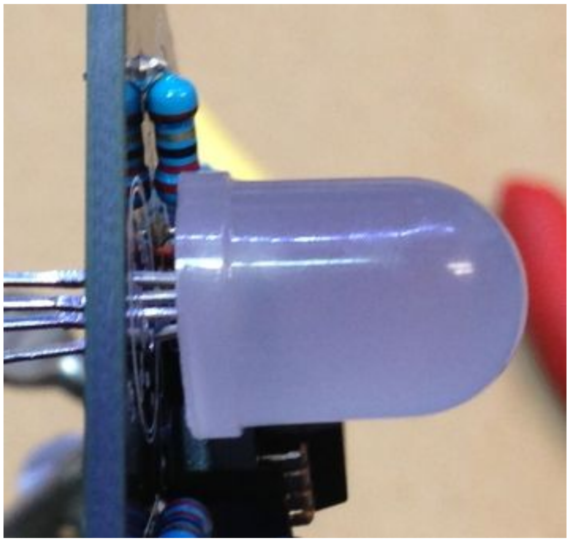
7. Now you should insert the ATtiny85 microcontroller in the socket. I’ve put a little silver paint dot indicating which pin is pin 1 on the microcontroller. This should be on the same side as the cutout in the socket. If you inserted the socket backwards, don’t worry, just make sure that the microcontroller is oriented correctly. When you receive the microcontroller, its legs will be splayed a bit. In order to get the chip in the socket, you will need to bend the legs in so they are parallel. With 8 pin chips such as the ATtiny85, you can usually just gently squeeze them between your fingers, but if you aren’t comfortable doing that, you can also use your bench top to flatten them like so:



Just be gentle and don’t crush the chip - it doesn’t take a lot of force to bend the legs.



8. Now you will need to insert and solder the LEDs. The LEDs are polarized, which means that it’s important you get the right legs into the right holes. You’ll notice that the holes are all marked “r”, “com”, “g” and “b” - those line up with the legs of the LEDs. Insert the longest leg into the “com” hole and the next longest leg into the “g” hole. Feed the next two legs into the “r” and “b” holes, then gently push the LED into place. As you start, the legs will look like the picture to the left.



If you look carefully at the legs you'll see a section where they're slightly flatter than the rest of the leg. Your goal will be to get the LED seated far enough that the shoulder of those flat spots is just through the circuit board. The LED will not be seated flat to the board - it will look like the picture to the left. Solder one of the legs and then look at the LED carefully and try to make it as square to the board as you can. Once you're happy with the LED's position, you can solder in the other 3 legs. Be gentle with the LEDs during this step - if you push too hard, you could rip the leg right out of the LED which would ruin the LED. Once you've soldered all 20 legs, clip the legs flush to the board.

Helpful hints:

• You might want to skip ahead and solder the pushbutton in after you solder in LEDs 2, 3 and 4 and before you solder in LEDs 1 and 5, depending on how small your fingertips are. It can be a tight fit to get it in after all the LEDs are in place.

• If you don't have a helping hand tool, you can balance the unpopulated half of the board on the back of the battery box for the first few LEDs. By time you're soldering the 3rd and later LEDs, the board self stabilizes fairly well on the tops of the LEDs.

9. Once you're done with the LEDs, your board should look like this. Insert the tact switch in the board - it's slightly rectangular, so it should only want to snap in to the board one way. If in doubt, the silver legs come off of the left and right side of the switch. If you get this rotated 90 degrees off, the reset button will be constantly depressed, and the microcontroller won't start to run the program.

10. At this point, the controller board is done. If you want to power it from a USB cable, you should be able to power it directly to the negative and positive terminals on the board. If you want to power it from the included battery box, follow along on the next page.

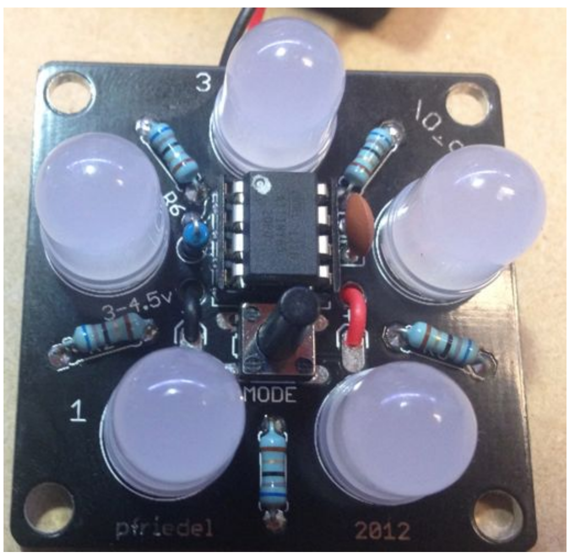
Helpful hint:

I highly recommend temporarily testing the circuit with the battery box before you solder it in to place to ensure that the kit is working properly. Once you solder the board to the box, repair and rework is more difficult. Simply connect the negative (black) lead from the battery box to the negative (-) pad on the board, and the positive (red) lead from the battery box to the positive (+) pad on the board. Be careful to get the power leads connected the right way around - the microcontroller will eventually fail if it's hooked up backwards.

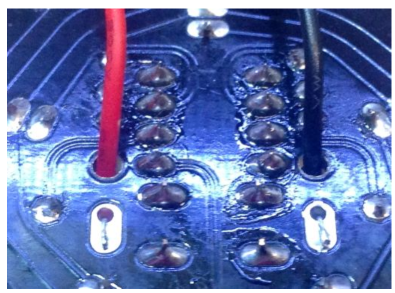
If any of the LEDs fail to light up or if any of the LEDs start up and glow white or any other solid color without changing like their neighbors, skip forward to the troubleshooting section.

11. Thread the black negative lead up from the back of the board to the front of the board and do the same for the red positive lead. Adjust the length of the wires so there isn't too much extra loose wire. At this point I usually try to make a mark on the wire where it passes over the hole and where it passes over the lower bound of the solder pad. Then I unthread the wire and trim it to length with a wire cutter and strip back the insulation and tin the exposed wire with a little bit of solder so the thin wires inside the red and black wires are coated in a small layer of solder.

Make sure the battery box is empty or the power switch is turned off at this point - it's easy to short out power with solder while making the next couple of connections, and that could damage the microcontroller or LEDs.

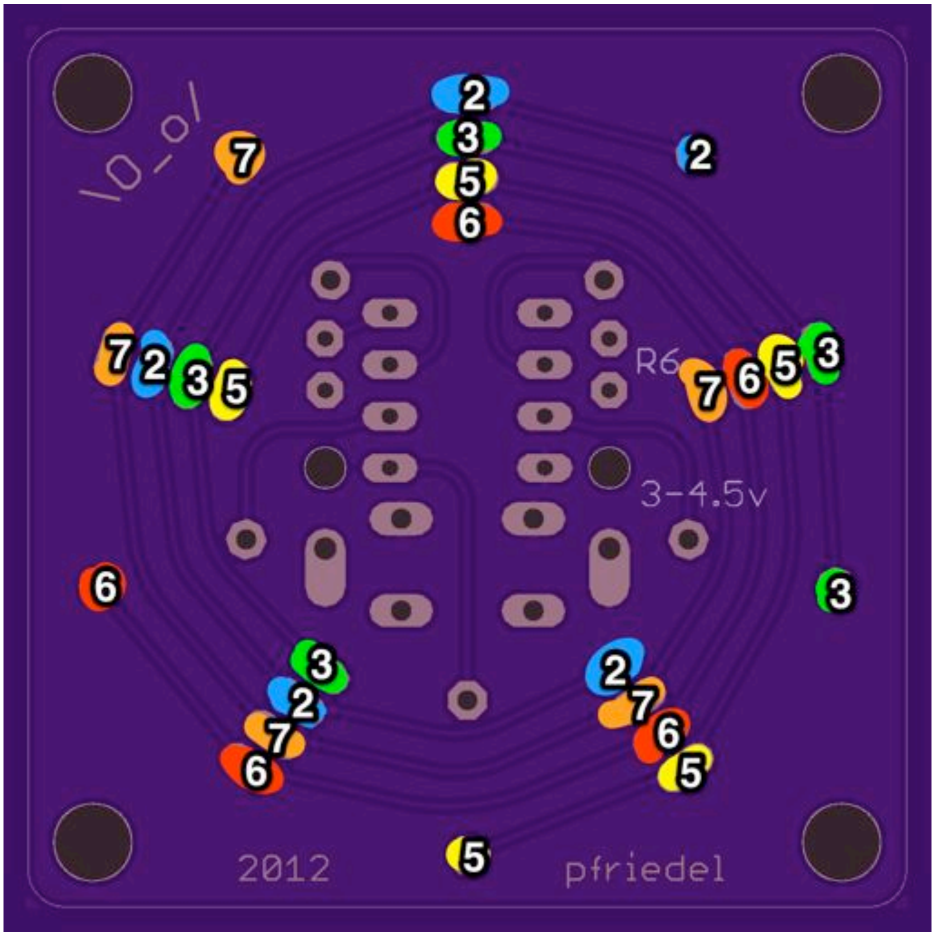


12. Once you have trimmed and tinned the battery wires, thread them up from the back of the board again and loop them over to the solder pads as shown on the left.



The back of the board should look like this. Solder the wires into place and tighten the excess from the front. Attach the board to the battery box with double sided sticky tape or hot glue. The board is now finished. Insert batteries and flip the power switch and it should start from mode 1.

TROUBLESHOOTING:



The first mode should never show a fully white color. If you see white, there's a short somewhere on the board. If you have a multimeter, you can use the diagram to the left to determine which points should have continuity and which ones shouldn't, which will help you narrow down the problem LED to one of three. (e.g. if there's a short between pins 7 and 2, it could be in any of the LEDs in the 10 o'clock position, the 7 o'clock position or the 5 o'clock position.)

If you're feeling especially cautious you can confirm there aren't any short circuits between any two pins on any single LED before moving on to the next one while you're soldering them, which greatly narrows down the possible faulty components to the one you just added.

DISPLAY MODES:

The board will change display modes every time you hit the MODE button. The board will tell you which mode you are entering by lighting up the LED associated with that mode which is next to a number on the board. 1-5 are indicated by red LEDs while 6-10 are indicated by green LEDs.

1. (Red 1) A slow walk through all the hues of the color spectrum.
2. (Red 2) A faster walk through the color hues.
3. (Red 3) Even faster than #2.
4. (Red 4) A slow progression with every color mapped one to one to LEDs. While modes 1-3 work with a wide virtual LED space, this mode and the next one work only with the physical LEDs.
5. (Red 5) A faster version of #4.
6. (Green 1) Random colors at random positions, changing every second or so.
7. (Green 2) Bouncing up and down in hue saturations, slowly walking through hues.
8. (Green 3) The same as #7, with a faster hue progression.
9. (Green 4) Displays the red, green and blue LEDs directly without mixing. Useful for spotting a bad LED.
10. (Green 5) Randomly selects any of the prior 9 modes and displays it for 5 minutes before selecting the next mode.

The system will automatically enter a low power sleep mode after 4 hours of modes 1-9, but mode 10 will run until the batteries run down. If you hit the MODE button while the system is in the low power mode, it will wake up and resume the previously running mode for another 4 hours. If you hit the MODE button again while it is still indicating which mode it is about to enter, it will advance to the next mode.

The low power mode is drawing just under 2 μA, so the power switch isn't strictly necessary - 2 μA is on the order of the battery's self discharge rate.

In normal use the board draws about 20 mA - the expected lifespan on standard Alkaline AAs is about a week nonstop or 168 hours. As the batteries die, the LEDs will slowly start to fade and each color will start to fail one by one. If you only use the 4 hour timed mode once a day, each set of batteries will last over a month given good alkaline cells.

SOURCE CODE:

The software repository for this project is at https://github.com/pfriedel/TinyFiveCircle

Any questions? Mail me - pfriedel@compulsive.net.