

Li'l Larson LED Scanner

Dale Wheat - 3 January 2011

Here's a fun and easy first project to try with your new Breadboard Arduino kit. Glen Larson is credited with producing several iconic American television shows, including *Battlestar Galactica* and *Knight Rider*, among others. Both shows featured mesmerizing "light scanners" that are easy to simulate and hard to reproduce exactly. Here's how to get started on building a your own "Larson Scanner".

Materials:

Breadboard Arduino kit -OR- a regular Arduino and a solderless breadboard

5 red LEDs

5 resistors, 470 Ω , 1/4W

Jumper wires

LED Polarity [could be a sidebar]

Look at your LEDs. If they still have their original leads, you'll notice that one is slightly longer than the other. The long lead is the *anode* or positive lead. The short lead is the *cathode* or negative lead. This is an important distinction as LEDs are *polarized* and will only work one way. If the leads have already been trimmed to the same length, you have a 50% chance of guessing right the first time and a 100% chance of guessing right the second time, so don't despair! It won't hurt the LED to install it backwards; it just won't light up.

Install the First LED

Take one of your LEDs and insert the long lead into tie-point C-20, with the shorter lead going into tie-point C-21. Install a jumper wire from tie-point D-4 (labeled "D2" on the chip) to tie-point E-20. Now install a resistor from tie-point B-21 to the left [blue \[-\]](#) ground rail. See Photo 1.

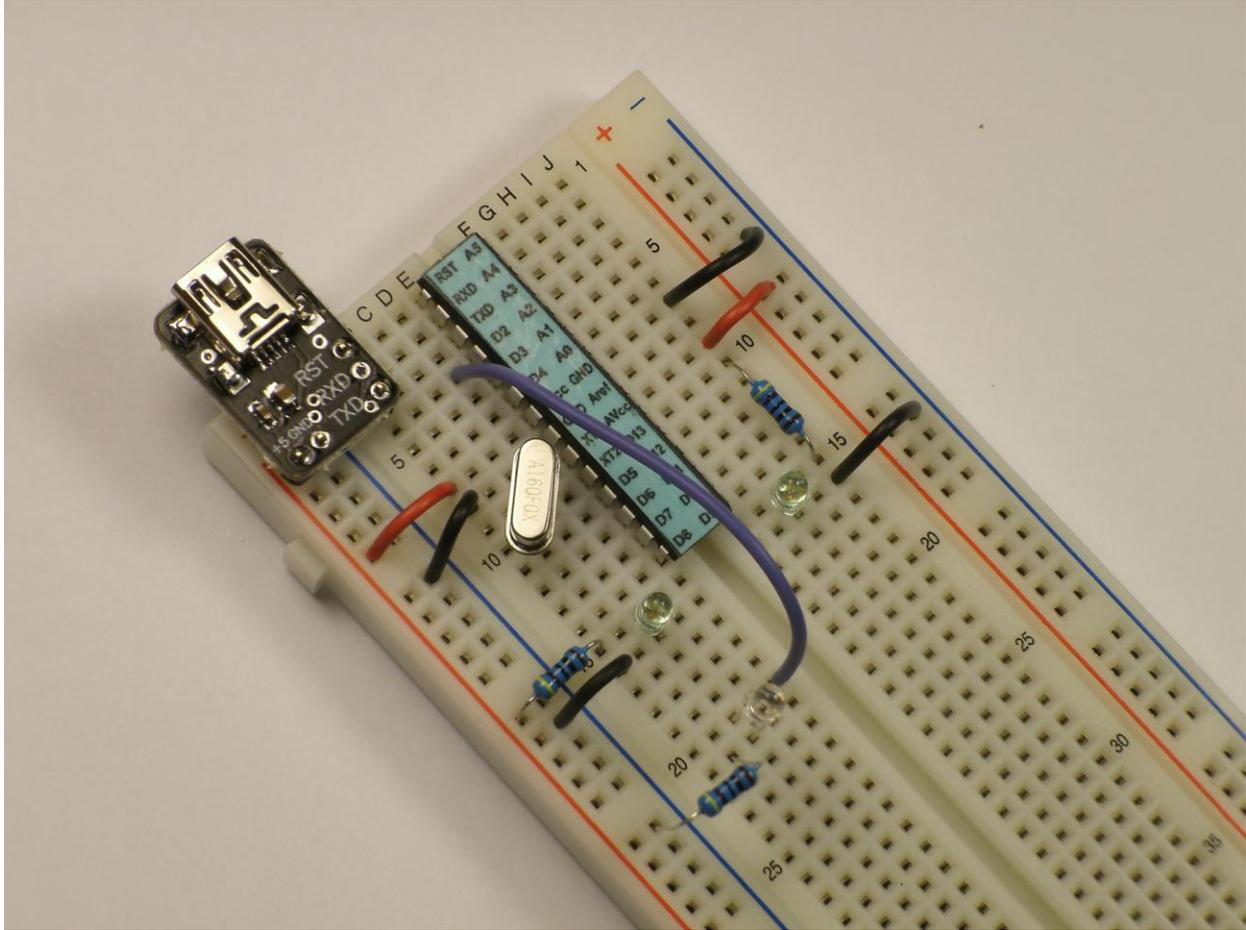


Photo 1. The first LED is installed.

The Remaining LEDs

In a similar manner, install the remaining four LEDs right next to each other in a row going down the breadboard. If you are using 3mm LEDs, use the table to find the exact locations, at least for the first time. 5mm LEDs or larger will need to be spaced out further to be able to line them up. Feel free to rearrange the LEDs to suit your imagination once you've got them blinking properly. See Photo 2.

Jumper wire		[+] LED [-]		Resistor	
Tie-point ("Pin")	Tie-Point	Tie-Point	Tie-Point	Tie-Point	
D-4 ("D2")	E-20	C-20	C-21	B-21	Blue [-] ground
D-5 ("D3")	E-22	C-22	C-23	B-23	Blue [-] ground
D-6 ("D4")	E-24	C-24	C-25	B-25	Blue [-] ground
(Skip over the power connections and the quartz crystal here)					
D-11 ("D5")	E-26	C-26	C-27	B-27	Blue [-] ground
D-12 ("D6")	E-28	C-28	C-29	B-29	Blue [-] ground

Table 1. All of the connections to light up the LED scanner circuit.

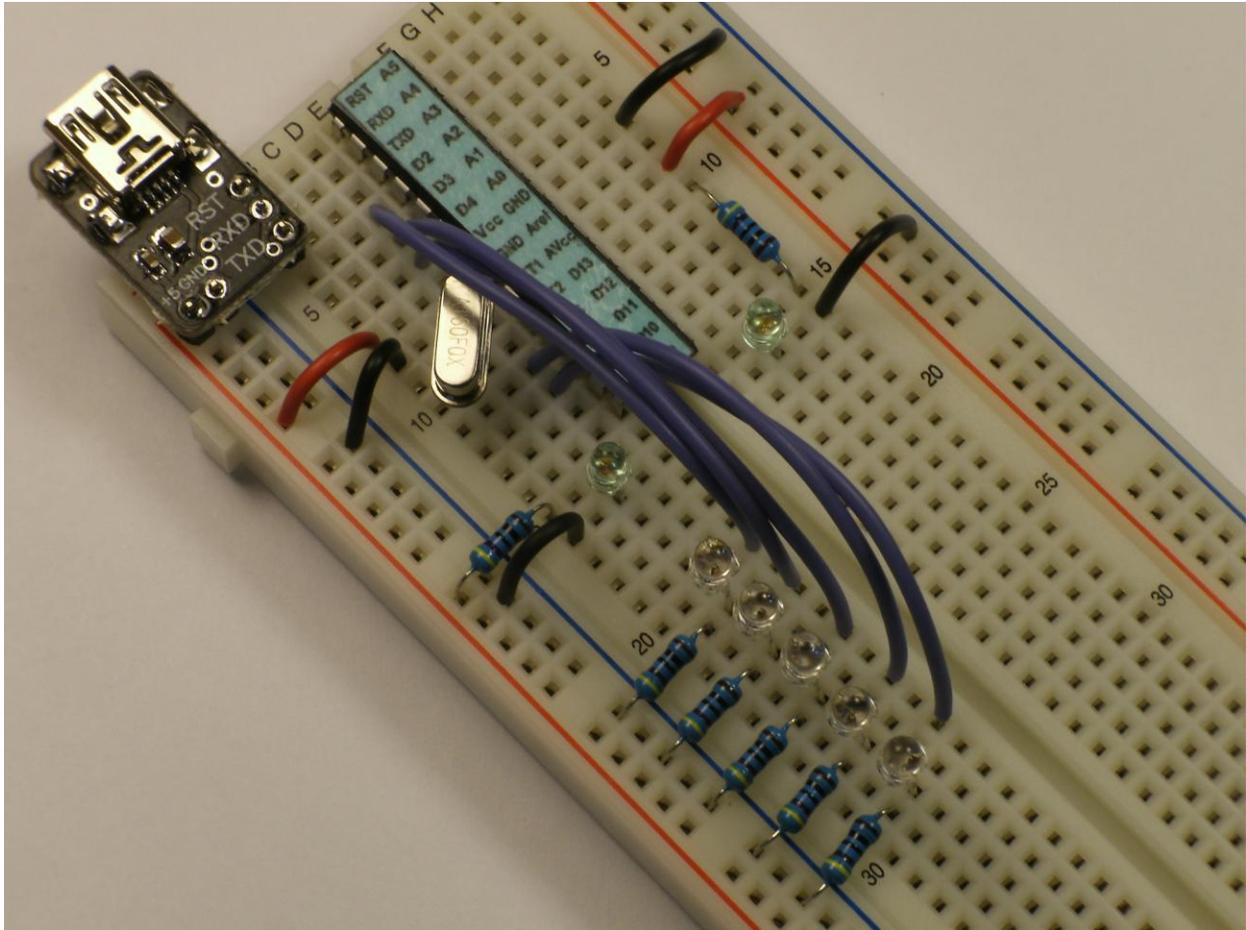


Photo 2. All of the LEDs are now installed and wired to the chip.

Software

You can use an existing Arduino example "sketch" to light up your newly-crafted Larson Scanner. From the Arduino IDE menu, select "File/Examples/5.Control/ForLoopIteration". This is a great example and introduction to a popular computer programming "control structure" called a "for loop".



ForLoopIteration

```
void setup() {  
  // use a for loop to initialize each pin as an output:  
  for (int thisPin = 2; thisPin < 8; thisPin++) {  
    pinMode(thisPin, OUTPUT);  
  }  
}  
  
void loop() {  
  // loop from the lowest pin to the highest:  
  for (int thisPin = 2; thisPin < 8; thisPin++) {  
    // turn the pin on:  
    digitalWrite(thisPin, HIGH);  
    delay(timer);  
    // turn the pin off:  
    digitalWrite(thisPin, LOW);  
  }  
  
  // loop from the highest pin to the lowest:  
  for (int thisPin = 7; thisPin >= 2; thisPin--) {  
    // turn the pin on:
```

Done uploading.

Binary sketch size: 1076 bytes (of a 30720 byte maximum)

```
void setup() {
  // use a for loop to initialize each pin as an output:
  for (int thisPin = 2; thisPin < 7; thisPin++) {
    pinMode(thisPin, OUTPUT);
  }
}

void loop() {
  // loop from the lowest pin to the highest:
  for (int thisPin = 2; thisPin < 7; thisPin++) {
    // turn the pin on:
    digitalWrite(thisPin, HIGH);
    delay(timer);
    // turn the pin off:
    digitalWrite(thisPin, LOW);
  }

  // loop from the highest pin to the lowest:
  for (int thisPin = 5; thisPin >= 3; thisPin--) {
    // turn the pin on:
```

Figure 1. The Arduino example sketch, "ForLoopIteration", can be used as a basis for your Larson Scanner. Compare it to the modified version.

Compile ("Verify" in Arduino-speak) the sketch and upload it. You should see the LEDs begin to scan back and forth. You should also see something is a bit "off". The original example sketch was written for six LEDs instead of five. You can fix this by making three small changes to the code.

On program line 24, change the "8" to a "7".

On program line 31, change the "8" to a "7".

On program line 40, change the "7" to a "5", and change the "2" to a "3".

Save the sketch under a new file name (don't over-write the example sketch) by using the Arduino IDE menu item, "File/Save As...". Choose a new name, such as "LarsonScanner" or similar. Remember, the Arduino IDE works on multiple operating systems and doesn't allow spaces or punctuation in filenames.

Compile the newly-modified sketch and then upload it to your Breadboard Arduino. If all went well, you should see your LEDs bouncing back and forth.

Going Further

For more fun, add an LED bezel to align your LEDs and give it that finished look. There's a template file available to make your own. I used a laser cutter and semi-transparent red 1/8" acrylic plastic. See Photo 3.

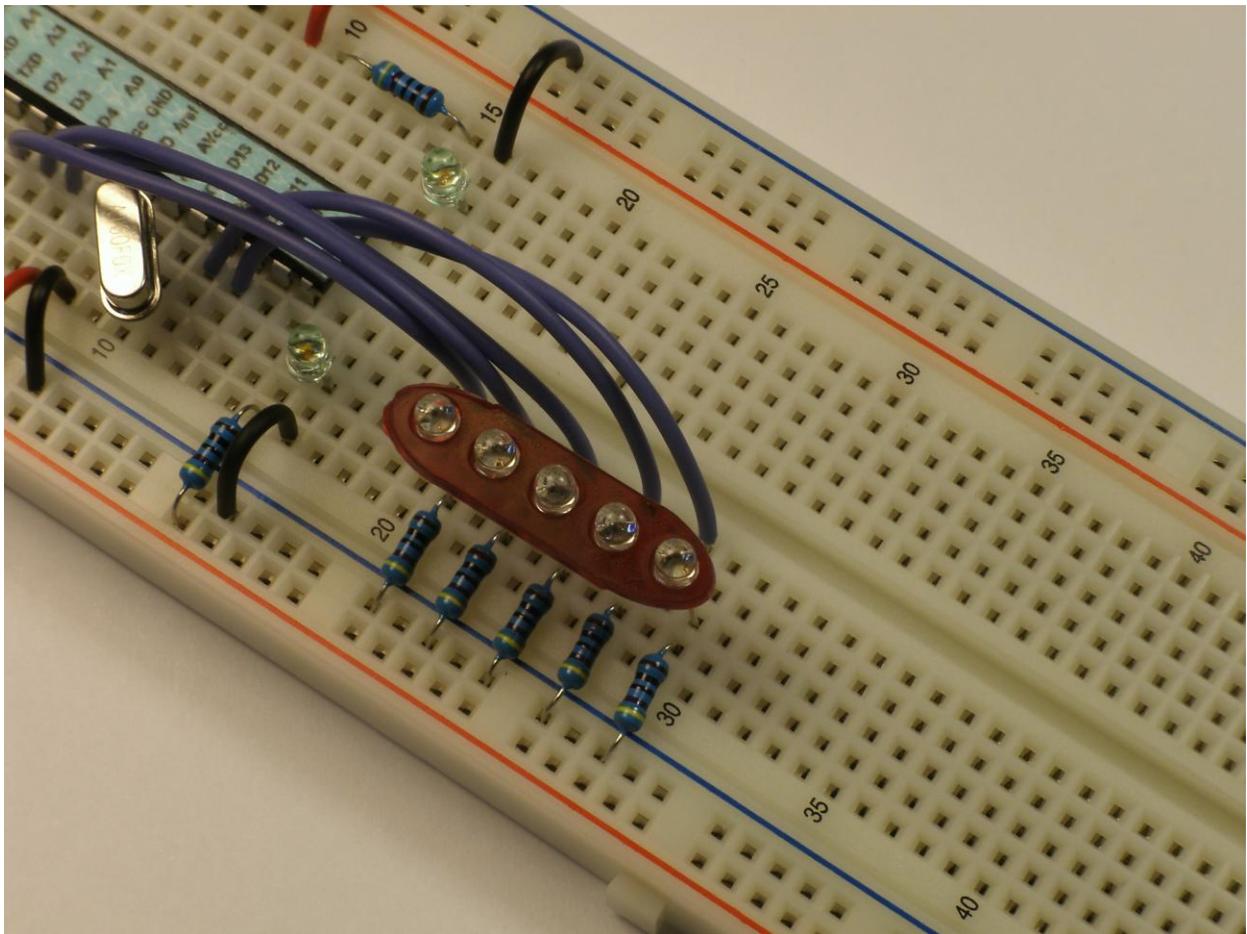


Photo 3. A small plastic bezel aligns the LEDs and gives your project a finished look.

All the materials to build this fun add-on to your Breadboard Arduino kit are available from my web site, <http://DaleWheat.com>, including the laser-cut LED bezel.

As I mentioned before, the Larson Scanner is easy to simulate with a few LEDs and a microcontroller, but it is tricky to reproduce exactly. Speed adjustment, LED fading and trailing turn a simple project into a work of art. I'd really love to see *your* version in action!

[end]