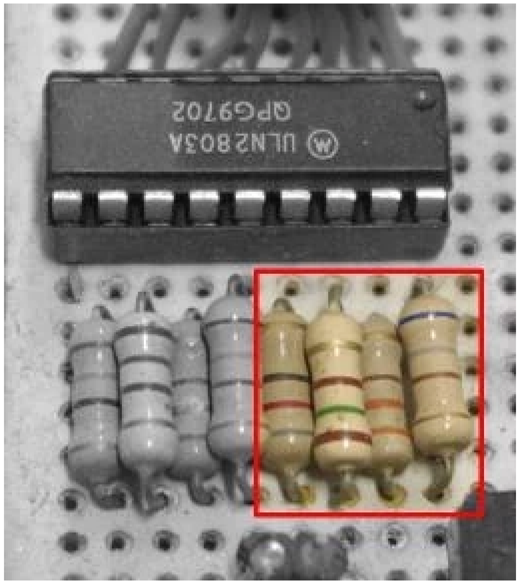
# Orac - The Light Ball

*Goodness gracious! Great balls of ... sentient processing capability.*



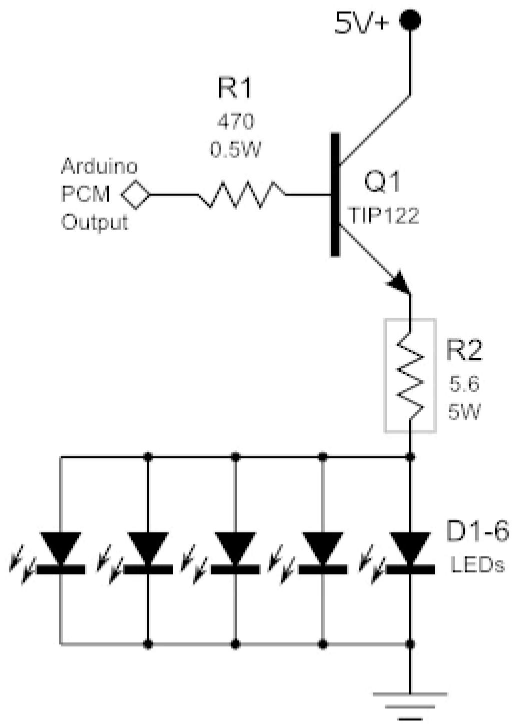

A central feature of Orac was a ball about 15cm in diameter. In the TV version this pulsed with white light on a regular cycle. It also had eight red lamps studding its equator.

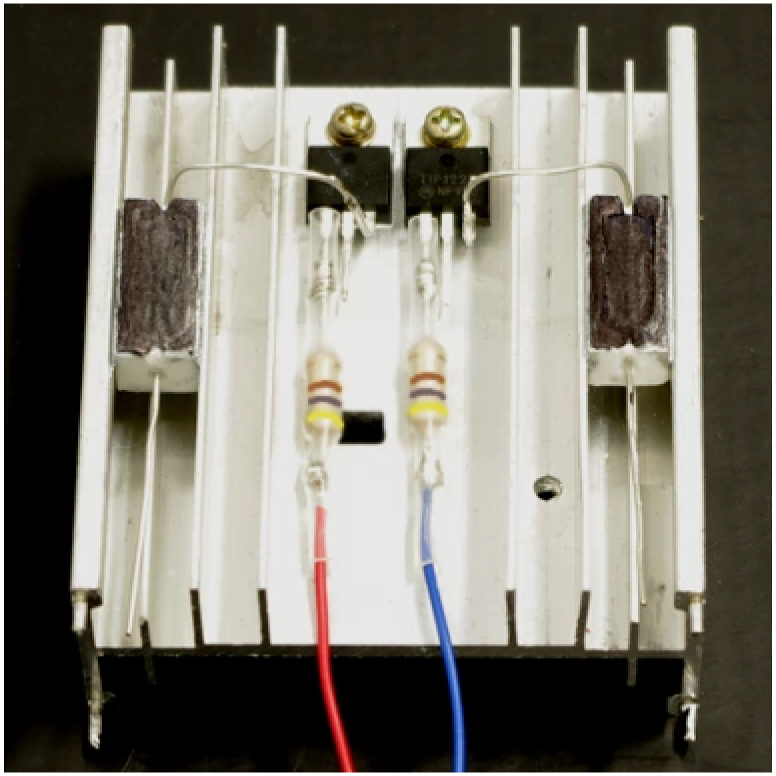
My version is fitted with high-brightness red and blue LEDs under independent control, so it can pulse red-violet-blue. My original circuit had some four-bit counters clocking at different rates into two simple resistor-based DACs* (*right*). With the Arduino's ability to drive LEDs with pulse-code modulation (see `AnalogWrite(pin,val)`), I will be using this more energy-efficient method.

* **D**igital-to-**A**nalog **C**onverters

### Drive Circuit

The outputs of the Arduino can only drive up to two 20-30mA display LEDs, the LEDs I am using for the re-make are lighting-grade LEDs and draw around 350mA each. And I have 6 of them for a total of 2100mA (about 2 amps) per colour channel. Obviously I would either kill the Arduino or fail to light the LEDs with direct drive, so I used a simple transistor switch based around a TIP122 part. These transistors can drive a DC load up to 4 amps so are perfect.



In the diagram *above, left:*

- R1 - limits the amount of current drawn by the transistor switch pin (base pin). The TIP122 is technically a [Darlington Pair](), which is a transistor driving another transistor inside the case, so it doesn't need much current to switch.
- R2 - a big fat low-ohms resistor to limit the current to the LEDs so they don't run too hot and burn themselves out.
- Q1 - the The TIP122 switching transistor
- D1-6 - Lighting-grade LEDs

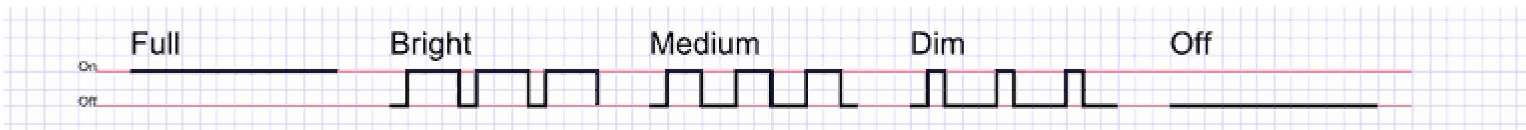
This circuit is duplicated, one for each of the Red/Blue LED banks.

This circuit is upside-down compared to the recommend (you normally would have the load - the LEDs in this case - between the positive supply and the collector of the transistor). Doing it this way is good enough for my purposes and saves me having to insulate the transistors' heat-sink tabs (which in this transistor are connected to the collector). I can just supply the drive voltage to the heat-sink on which both transistors are bolted. The side-effect of having the circuit this way is a bit of trickle current from the transistor base going through the LED banks but this won't be noticeable here.

*Above, right:* carrying two amps each, the load resistors get a bit warm, so they have been wedged between the fins of a heat-sink with some silver-based heat transfer paste on them. The transistors are also on the heat-sink, though they don't get so hot. Having it all mounted to one block of aluminium makes the circuit easier to assemble.

What is this **Pulse-code-modulation** you speak of?

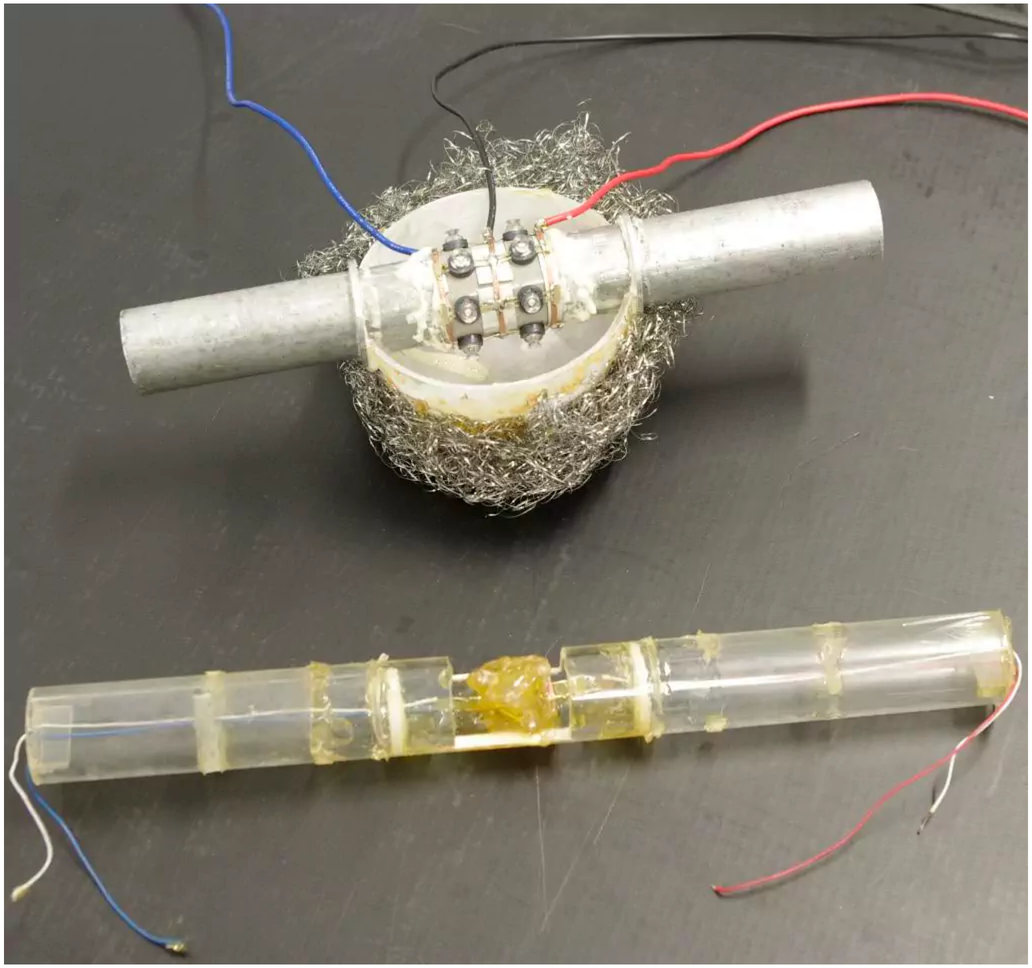
Pulse Code Modulation (PCM) is a long compound-word that simply means turning the output on and off at high speed with a particular ratio of off-time to on-time. When driving a light source, you make the on/off speed faster than the human eye can perceive (around 50Hz - the Arduino cycles at over 400Hz, so is way over the line) and the eye will average the light output. The longer the on-time relative to the off-time in a single pulse, the brighter the eye will perceive the light source, and vice-versa. The advantage here is that you are not burning power into a dump-resistor to dim it - you are instead not supplying power at all for part of the cycle, so the circuit will be more efficient and run cooler when the lights are dimmed. This trick will work for LEDs and Incandescent (domestic, xenon, halogen, etc.) light sources, but not for fluorescent lights, which require much more complex circuitry to facilitate dimming. Neon *lamps* should work, but I don't think neon *tubes* would.




## (re)Construction

### Primary Light Source

The epoxy-resin on my original was quite degraded (also a horrid-looking snot-brown after 15 years) and quite easy to break, so I was able to re-use the plastic balls from the original. Here is an image of (*bottom*) the old internal lights: 4x 5mm high-brightness LEDs per colour channel buried in a blob of epoxy - it was the '90's and that was as good as I could get; and *top*, the new 6x lighting-class LEDs per colour channel attached around an aluminium pipe with heat-transfer tape (these LEDs are so bright, they need heat sinking).



Around the LEDs I have a small (5cm) plastic ball, frosted by applying some acetone with a paintbrush, to better diffuse the light. Around this is a layer of somewhat-teased-apart stainless-steel wool to obscure the details of the interior while allowing the light out (and further diffusing it). All this is in a 12cm plastic ball. The smaller ball was the tops of two of some sort of mini gum-ball dispenser from the supermarket for a couple of dollars each. The outer ball came from a baby's roll-along toy, also from the supermarket for about $10.

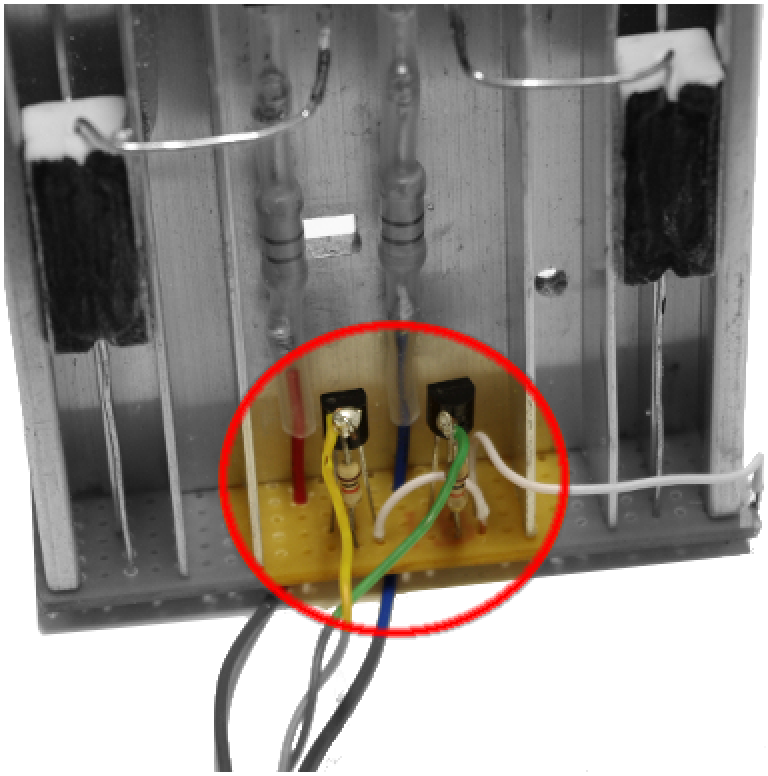
### Circumfrence LEDs

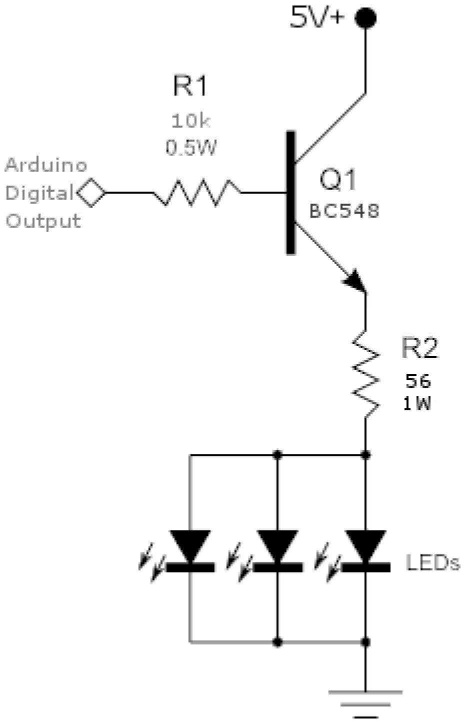
The TV Orac had eight red lamps around the outside of the light ball, flashing in two tandem sets. Because of the colouration of the ball, I decided to dispense with the red lamps in my original. I am bringing them back for the remake as 5mm high-brightness display LEDs. I will make them white, to contrast with the red-violet-blue of my ball (green was the other option, but I think that would look awful). I had a supply of spare high-brightness white LEDs from a string of decorative lanterns I bought for the lantern part, not the LEDs (see: RGB lantern demo project).

I will be using six circumference lamps (2 sets of 3) rather than 8 to avoid the assembly nightmare of having two of the lamps on the seam between the ball-halves. I am also mounting them in black push-in bezels, so they are more visible, since the LEDs themselves are clear plastic mounted in the clear plastic of the ball. Filing the top of each LED flat and leaving the surface rough provides a diffusion layer in place of the lens provided by the original LED shape. Once pressed into the holes in the ball, a blob of acrylic resin over their backs holds them in place (and keeps the steel wool out of short-circuit range). I must say!, with the bezels they turned out A LOT better looking than I had been expecting!

I was originally going to have only 4 LEDs before looking at an image of the TV Orac and realising this was just too few. I had already drilled holes at 45 degree offsets in the bottom ball-half, so now am using these extra holes to route the cabling out of the ball, in the finest tradition of "*it was meant to be like that all along*."

Three LEDs (even of this lower-power type) are a bit much for the Arduino, so I will be implementing another transistor switch as above, though I can use much smaller transistors this time. The BC548 transistor can switch up to 100mA, which is 5 display-class LEDs (4 if you want to play it safe):





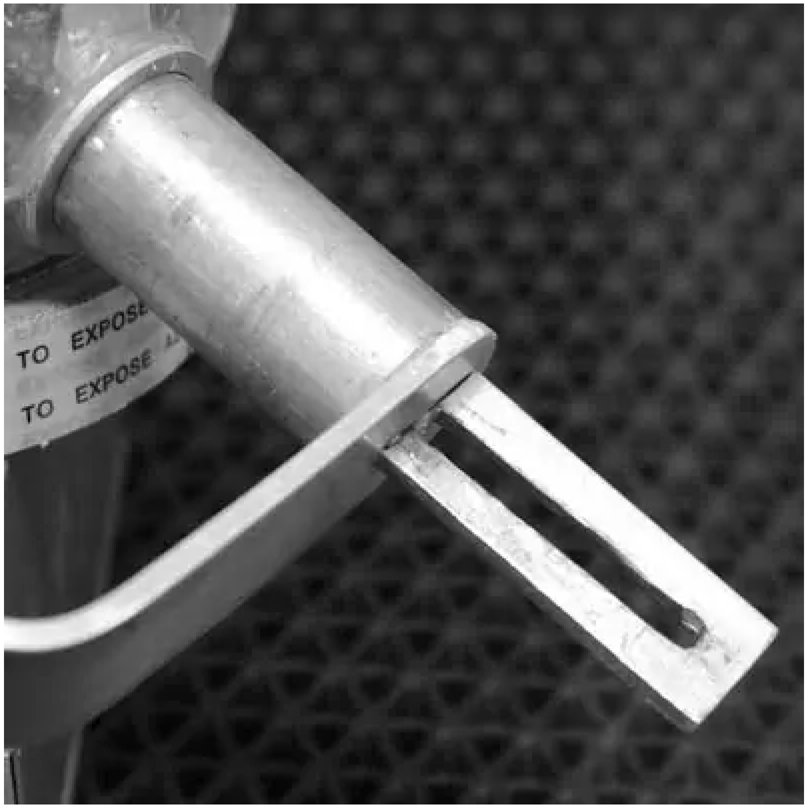

Again, the circuit *right* is duplicated, once for each triplet of LEDs.

The heat-sink used for the light-ball transistors and resistors still had the circuit board mounting pins on the bottom, so attaching a small rectangle of stripboard to the bottom for the smaller circuit was a simple task.

**Assembly**

Having squeezed it all together, I put some acrylic joiner around the seam of the outer ball (while the ball isn't quite acrylic, some testing on an out-of-sight edge indicated it was close enough for the joiner to work with). Rather than hold it together with my hands for several hours, or set up some awkward clamps, I sat the ball on a roll of tape, with another roll on top and a 4 litre tin of paint on that. I had a 10 litre tin on reserve, but the 4 was just enough to counter the force of the compressed steel wool trying to get back out.


Some strips of aluminium were cut and bent to hold the heat-sink and pinned into the aluminium tube with another piece of aluminium on each side. In order to securely attach the metal to the perspex, some locking pegs of perspex are put through slots in the pins. These perspex pegs are held in the slots mechanically and are joined to the perspex tube extensions with acrylic joiner. This mechanical joining avoids the need for gluing metal to plastic - a difficult and unreliable bond to make.

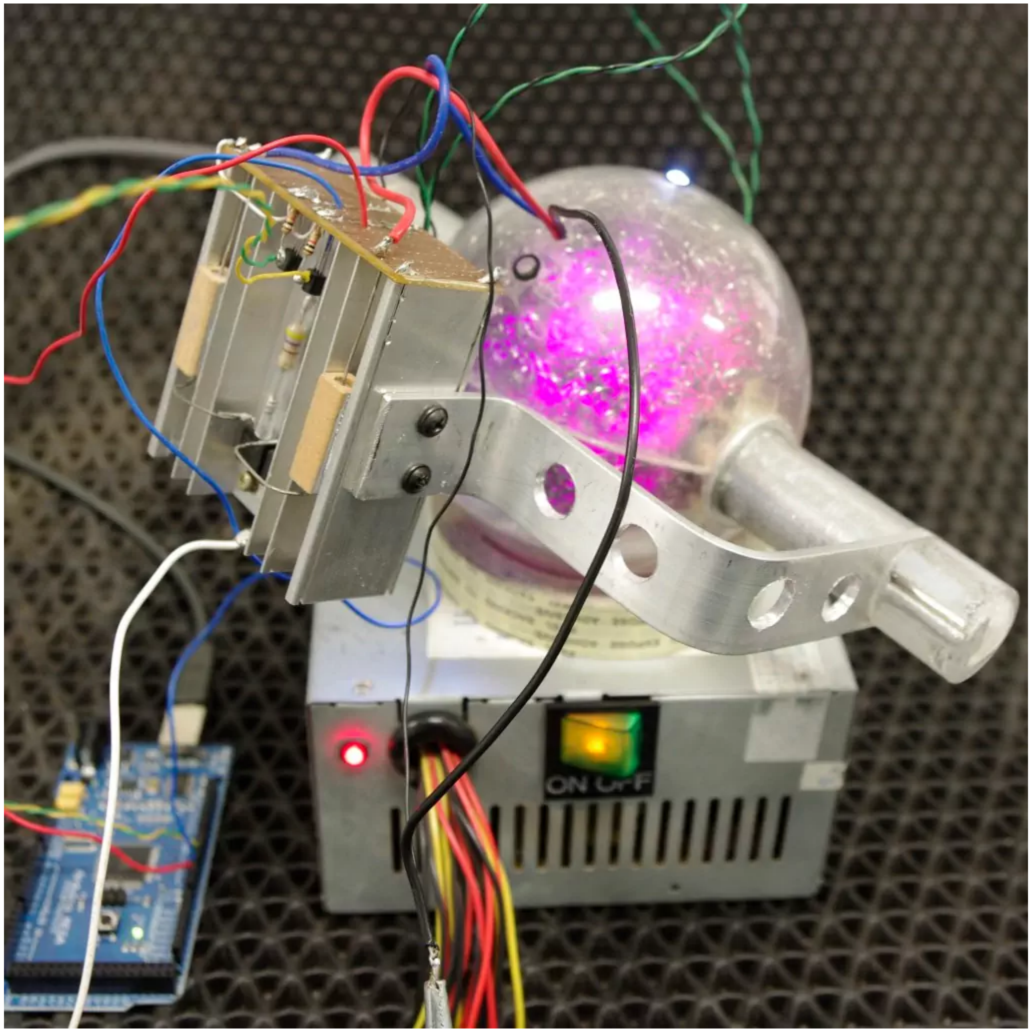



I ended up cutting the ends so short that the pin-in-slot idea was no longer effective (except for stopping rotational twisting - which is necessary in itself!) so the join is now held simply by the surrounding box-walls. But I left this example stand as it is a generally useful idea for other situations.

### Testing

Wrote a quick Arduino script and hooked the ball up to the controller and a big chunky 5V bench power supply* - the ball can draw up to 4.5 Amps if everything is at full brightness! We have pretty much left wall-wart territory. **Result:** The ball changed colours and the circumference lights blinked back-and-forth. Very happy! I don't have to turn out the lights to demo my new version!! - the bright white reflection you can see in the top-front of the ball below is the ceiling fluros in my office.

I have also drilled large holes in the heat-sink support arms: because the future is full of bits of metal with holes in them! Also, it is more places for lighting to be visible through and provides some anchor-points if I need to thread any wires or cable-tie anything in that area. (Remember, Orac is *supposed* to look a bit home-built!)




* My bench supply is made from an old PC power supply, so can deliver 5V@23Amps and 12V@9.5Amps. When I say old, I mean back in the days when you had to turn your computer off manually via a big chunky switch in response to a "It is now safe to turn off your computer" message - modern PSUs need some extra wiring trickery to convert to a bench supply to overcome the main-board switch-on/off feature. You will notice I have replaced the dangly wire with the switch on the end with a chassis-switch in the side of the case - because dangly wires with 240V in them scare me! I also added a red LED lamp to the (no-longer needed externally) *Power Good* wire.

```
//quick test of Orac's light ball - make lights flash and pulse

int redChanValue = 0; // we randomly walk up and down the red channel brightness
int bluChanValue = 0; // we randomly walk up and down the blue channel brightness
int tickerValue = 0;  // count the loops and switch the circumfrence LEDs every 8 loops
```

```
void setup()
{ pinMode(2, OUTPUT); // Red Ball Channel - "Analog" PCM output
  pinMode(3, OUTPUT); // Blue Ball Channel - "Analog" PCM output
  pinMode(4, OUTPUT); // Circumfrence LED Channel 1 - Digital output
  pinMode(5, OUTPUT); // Circumfrence LED Channel 2 - Digital output
}

void loop()
{ redChanValue += random(-16,16);             // increase or decrease the output brightness by up to 16/256ths each cycle
  if(redChanValue > 255) redChanValue = 255;  // don't overflow the maximum permissible value
  if(redChanValue <0 ) redChanValue = 0;      // don't underflow the minimum permissible value
  analogWrite(2, redChanValue);

  bluChanValue += random(-16,16);             // remember, adding a negative value is subtracting
  if(bluChanValue > 255) bluChanValue = 255;
  if(bluChanValue <0 ) bluChanValue = 0;
  analogWrite(3, bluChanValue);

  if(tickerValue == 0 )    // as tickerValue crosses 0, change the circumfrence LEDs
  { digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
  }

  if(tickerValue == -7 )   // as tickerValue is reset to -7, change the circumfrence LEDs
  { digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
  }

  tickerValue++;           // Tick value will count from -7 to 7

  if(tickerValue == 8 )    // when we overflow
  tickerValue = -7;        // go back to start

  delay(100);              // pause 1/10th second before starting the loop over
}
```

---

[Next: The Ring](#)

**Back to Glenn's Project page.**

 **Glenn's home page.**