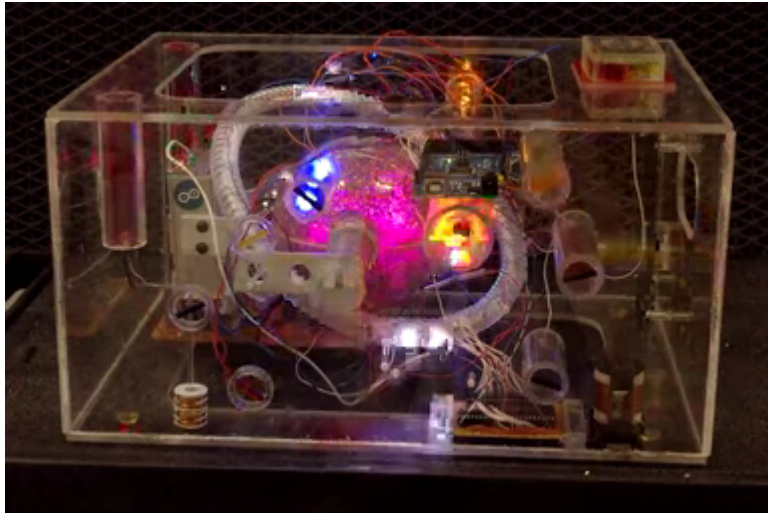


Orac - The Wrap-up

Here is a short video of my ORAC in action: [SEE ORAC RUN.mkv](#) 2.9MiB, h264, vorbis.

Here is a nice silent loop snipped from the middle of the above:



Well, I made it! Both figuratively and literally. Below is the complete code of the Arduino program. You will notice some variations to the sample snippets which were stripped to the minimum to enhance their value as examples. The below code needs a bit more of a clean-up which I hope to get to in the next few weeks, but it works fine.

```
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
int count = 0;      // throw-away variable to count loops in
int on = 0;          // track if we are on (on-off state will lag the switch somewhat)

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
  pinMode(6, OUTPUT); // Tick Channel - pulse this pin to advance lights
  pinMode(7, OUTPUT); // Power Chanel - "HIGH" to have lights on. "Analog" PCM output is an option.
  pinMode(8, INPUT);  // Key inserted input

  for(count=22; count <53; count+=2)
    pinMode(count, OUTPUT); // set up all the even-numbered board-end outputs
}

void lightShow()
{ 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 == -25 ) // as tickerValue is reset to -7, change the circumfrence LEDs
  { digitalWrite(4,LOW);
    digitalWrite(5,HIGH);
  }

  tickerValue++; // Tick value will count from -25 to 25

  if(tickerValue == 26 ) // when we overflow
  { tickerValue = -25; // go back to start

    analogWrite(7, 255); // full brightness

    digitalWrite(6,HIGH);
    delay(50); // pause so the pulse exists long enough to register
    digitalWrite(6,LOW);
```

```

    digitalWrite(random(11,27)*2, random(2) );    // turn on or off a random tarriel cell LED state

    tone(9,100+random(20)); // make the slightly-varying low-level buzzing noise characteristic to Orac
}

void powerUp()
{ on = 1;
  tone(9, 2000);          // start-tone - higher pitch pulse
  delay(100);
  for(count=1500; count >100; count-=100)
  { tone(9, count);        // drop pitch to running hum
    delay(10);
  }
}

void killShow()
{ digitalWrite(4, 0);      // turn off the circumfrence lights
  digitalWrite(5, 0);      // turn off the circumfrence lights
  analogWrite(7, 0);        // turn off the ring
  for(count=22; count <53; count+=2)
    digitalWrite(count, 0); // turn off all the tarriel cells
  for(count=230; count >30; count-=5)
  { if(redChanValue > count-30)
    { redChanValue = count-30;
      analogWrite(2, redChanValue);    // fade out the ball red channel
    }
    if(bluChanValue > count-30)
    { bluChanValue = count-30;
      analogWrite(3, bluChanValue);    // fade out the ball red channel
    }
    if(count % 30 > 15)
      tone(9, count);    // kill-tone-low - pitch drops
    else
      tone(9, count*2);  // kill-tone-high - modulate the dropping pitch up periodically
    delay((250-count)/5); //taper out the timing too
  }
  noTone(9);             // turn off sound completely
  analogWrite(2, 0);      // turn off the ball red channel
  analogWrite(3, 0);      // turn off the ball red channel
  on = 0;
}

void loop()
{ if( digitalRead(8) == HIGH && on == 0)    // key in unit off = startup
  powerUp();
  else if( digitalRead(8) == HIGH && on == 1) // key in unit on = keep running
    lightShow();
  else if( digitalRead(8) == LOW && on == 1) // key removed, unit on = shut down

```

```
        killShow();  
    else  
        delay(100); // if we are all off, wait 1/10th of a second and check the switch again  
}
```

Orac will likely never be entirely finished, as I will no-doubt find extra bits to glue into it, and will probably keep tweaking-away at the code from time to time as I learn new tricks in other projects. I will update these pages as appropriate in this case.

Hope you enjoyed sharing the creation and hope you can take away some useful ideas for your own projects.

▲ **Back to Glenn's Project page.**



Glenn's home page.