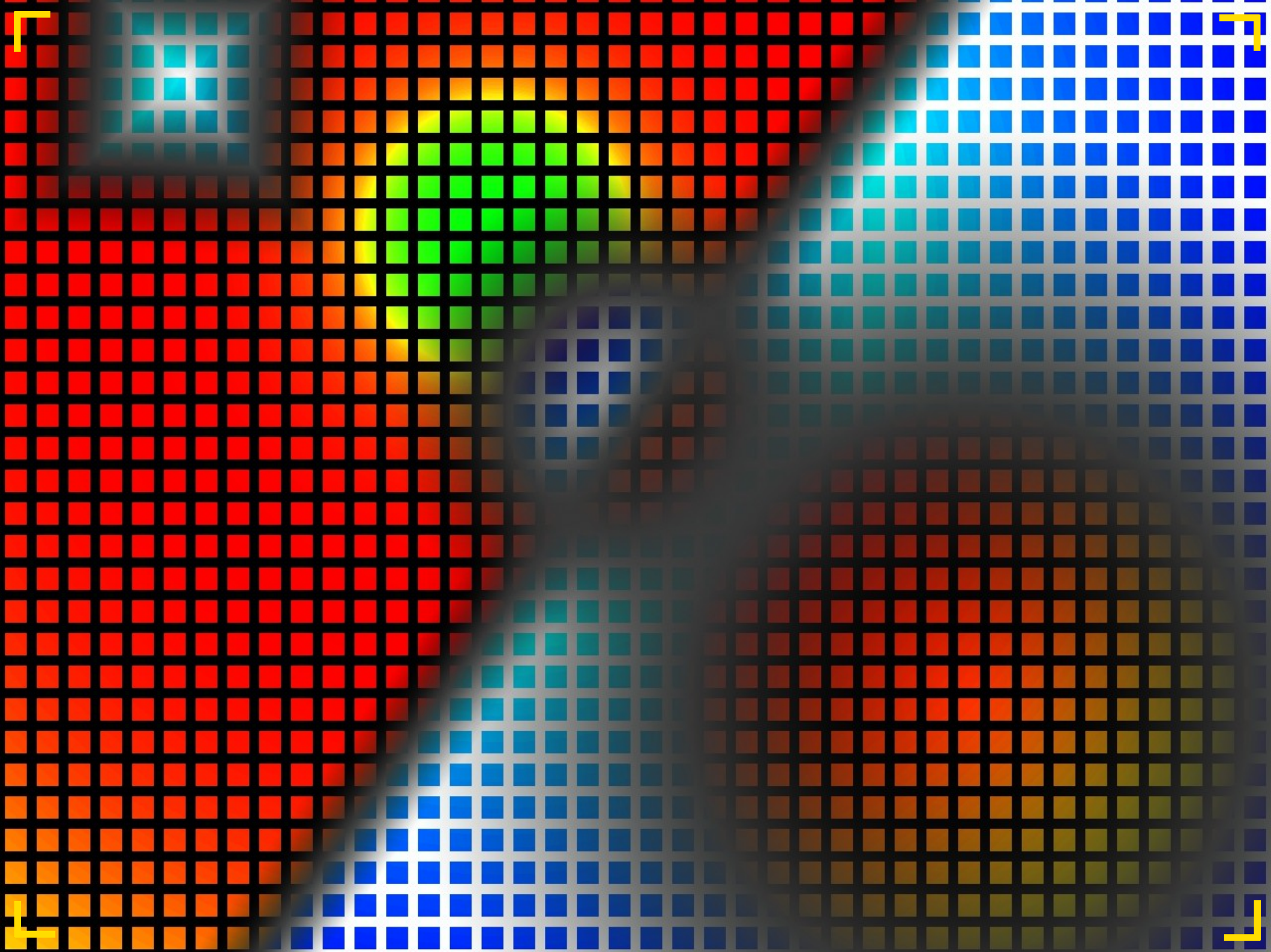
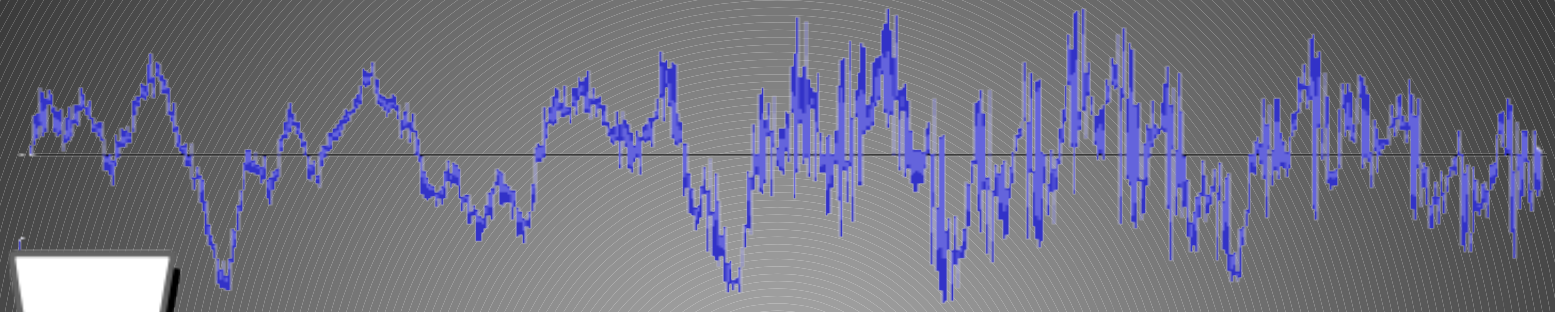




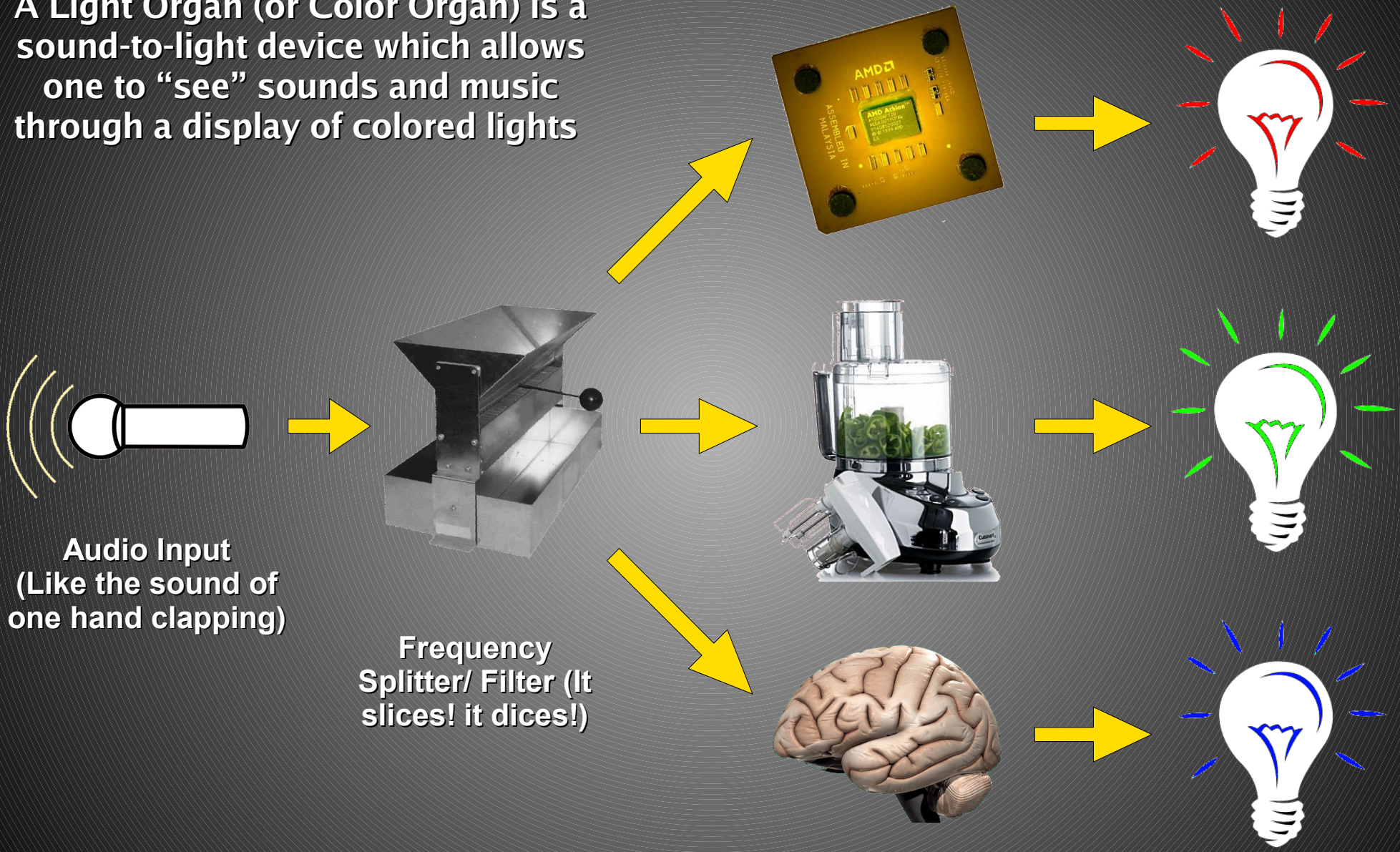
LIGHTOR

Six - Channel Digitally - Controlled (Color) Light Organ

What's a Light Organ?

Typical Light Organ Block Diagram

A Light Organ (or Color Organ) is a sound-to-light device which allows one to “see” sounds and music through a display of colored lights



Audio Input
(Like the sound of one hand clapping)

Frequency Splitter/ Filter (It slices! it dices!)

Processors / Drivers

Flashing Effects Lights
(Now in technicolor)

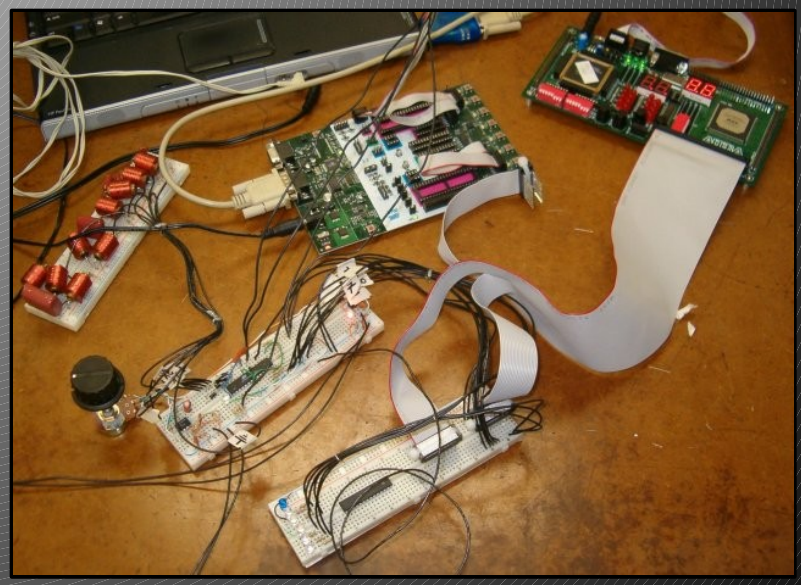
Design Comparison

Traditional Light Organs:

- Fully analog designs
- Usually use coarse RC filters
- Typically 3-channel designs for lows (100Hz), mids (1kHz), and highs (3kHz)
- Mediocre performance due to their often “low” tech implementations

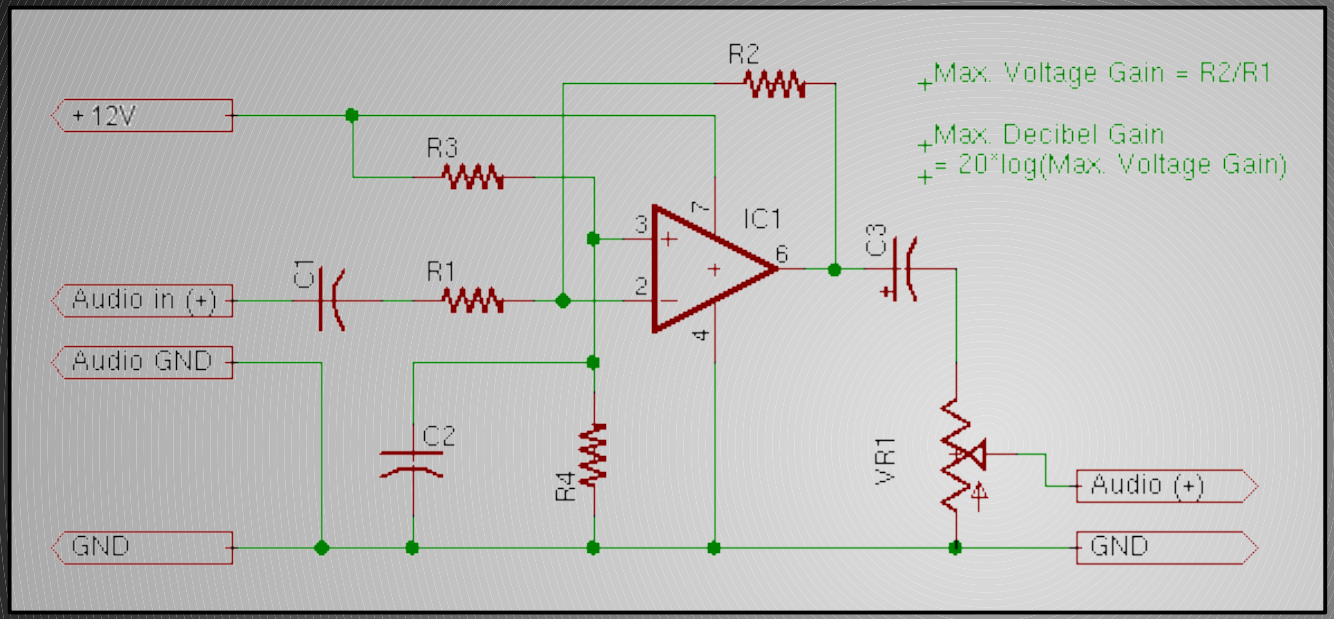
The Lightor Design:

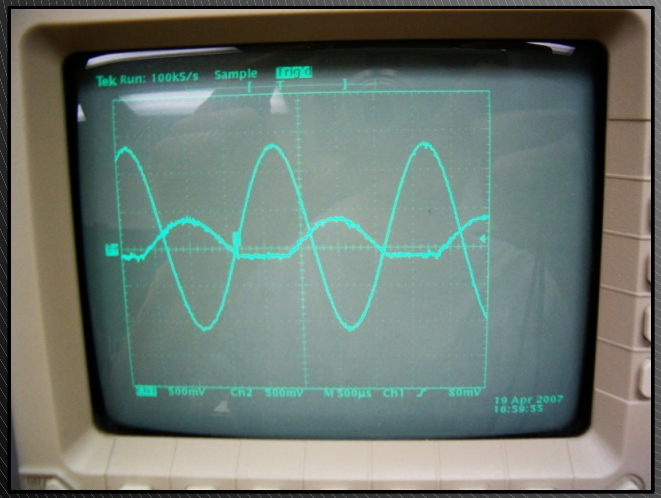
- Analog/ digital hybrid design
- Resonant RLC filters for tighter control
- 6-channel (60Hz, 250Hz, 500Hz, 1kHz, 3kHz, 6kHz) design aims to provide more clarity across the audible spectrum and more flashing lights mean extra fun
- Dimming/ flashing of the lights is achieved using 8-bit PWM for precise control of the light show.

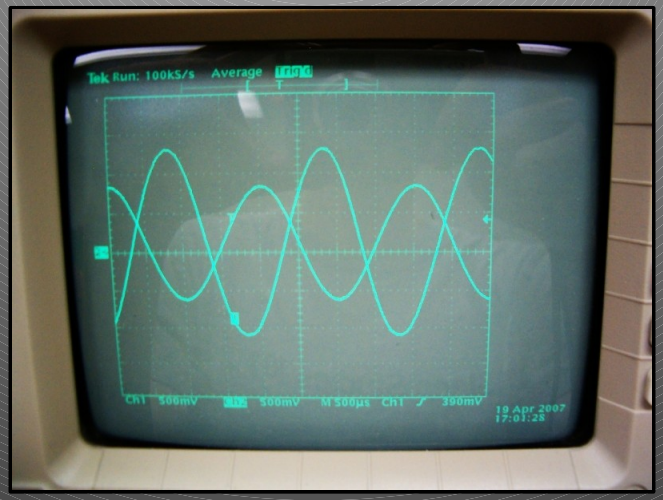


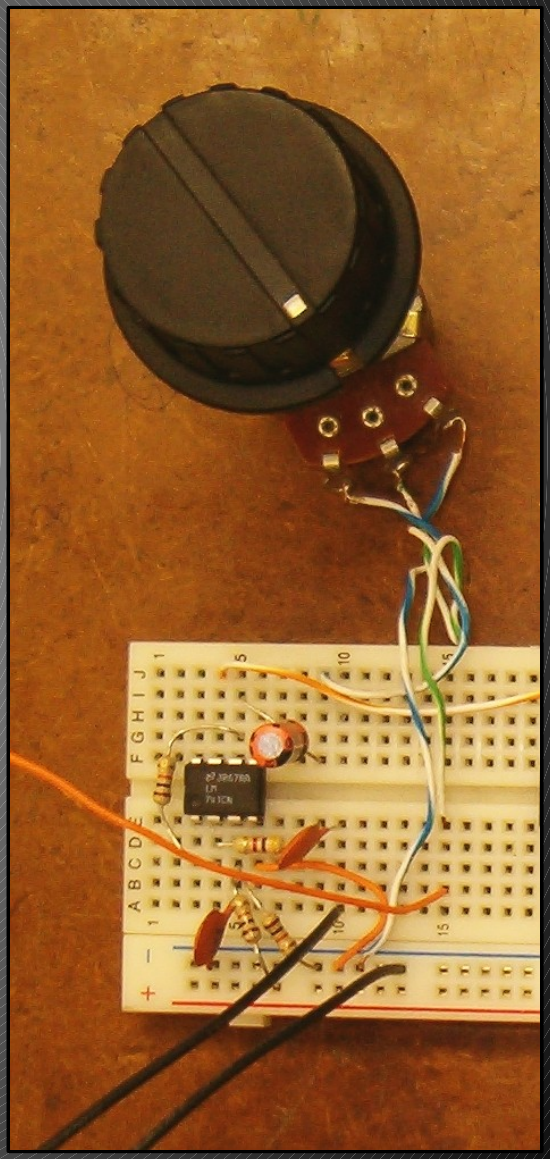
Preamp [Input] Stage

The Preamp Stage uses a simple op-amp circuit to amplify the inputted line-level audio signal to a more usable level



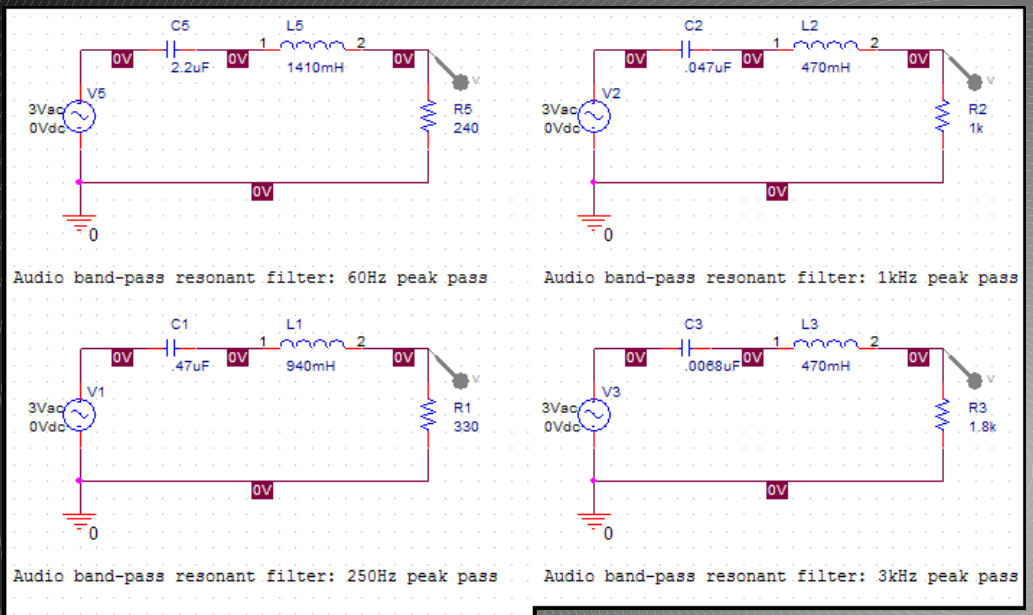


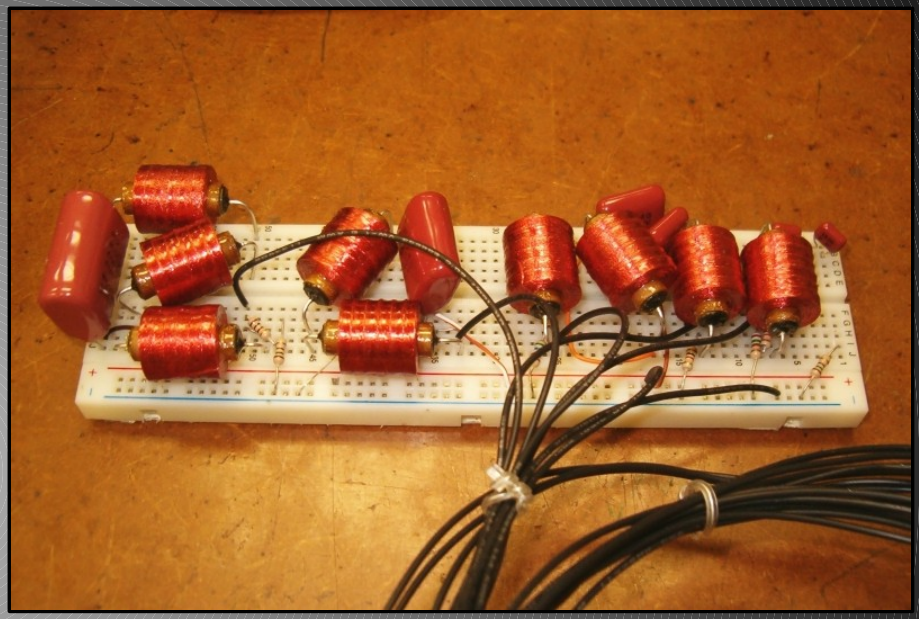


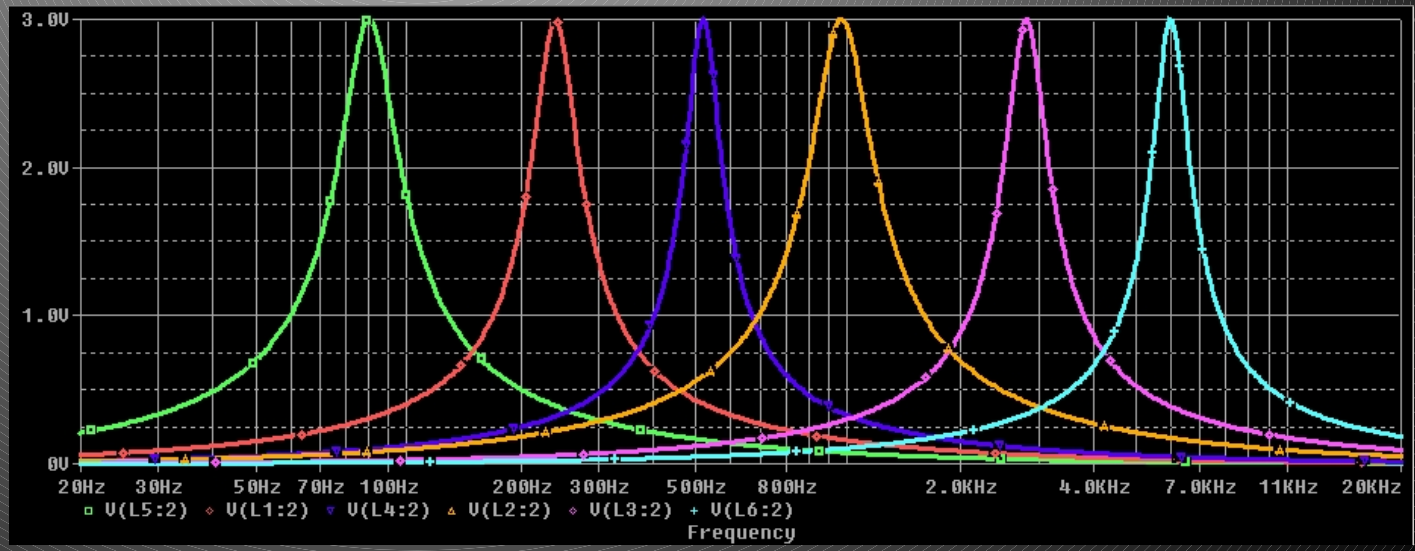


Filter Stage

The Filter Stage uses capacitors, resistors, and inductors to form six resonant band-pass filters tuned to six frequencies across the audible spectrum

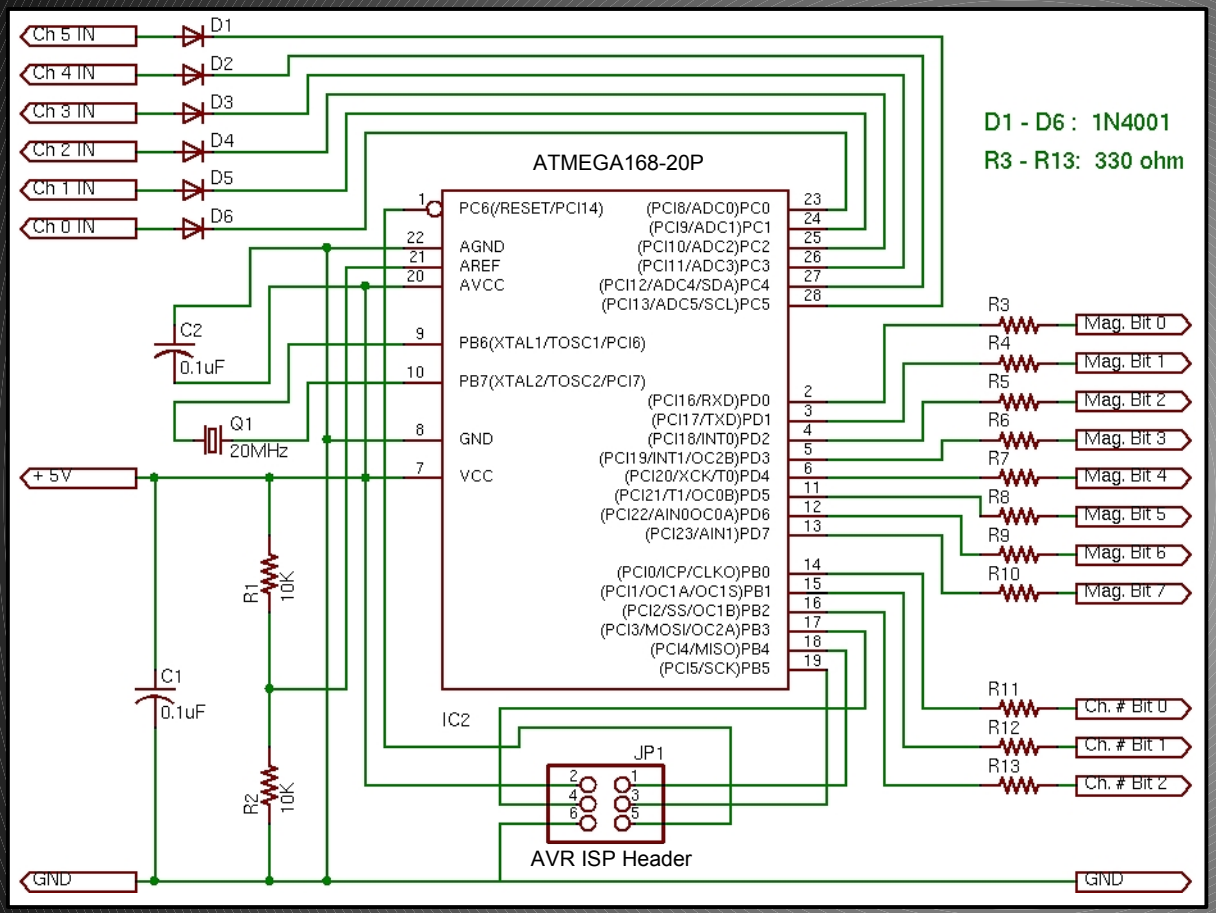


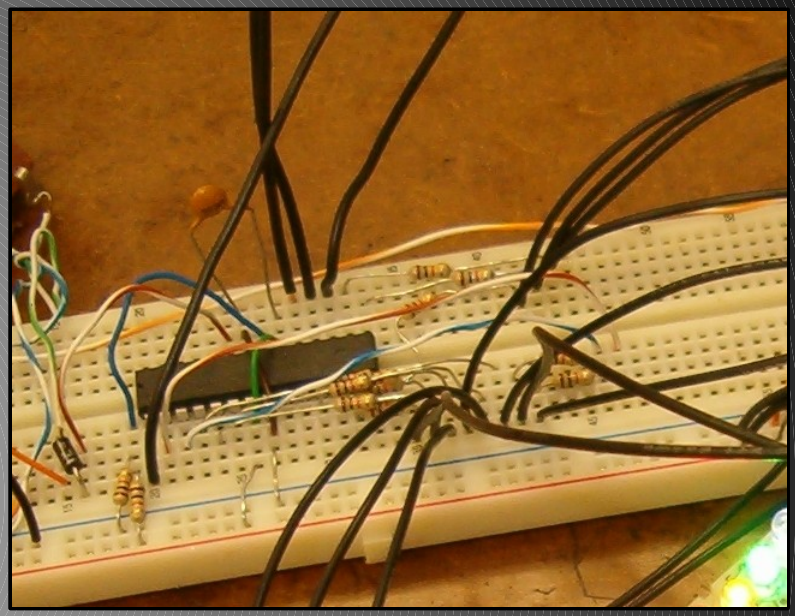


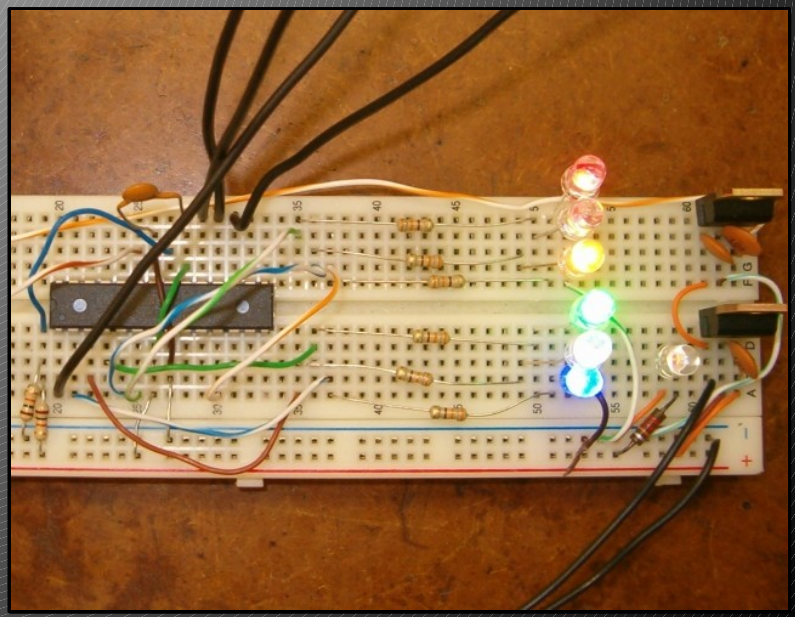


Conversion Stage

The Conversion Stage uses a microcontroller with integrated ADC to convert the audio signals to 8-bit binary numbers which it then multiplexes and outputs



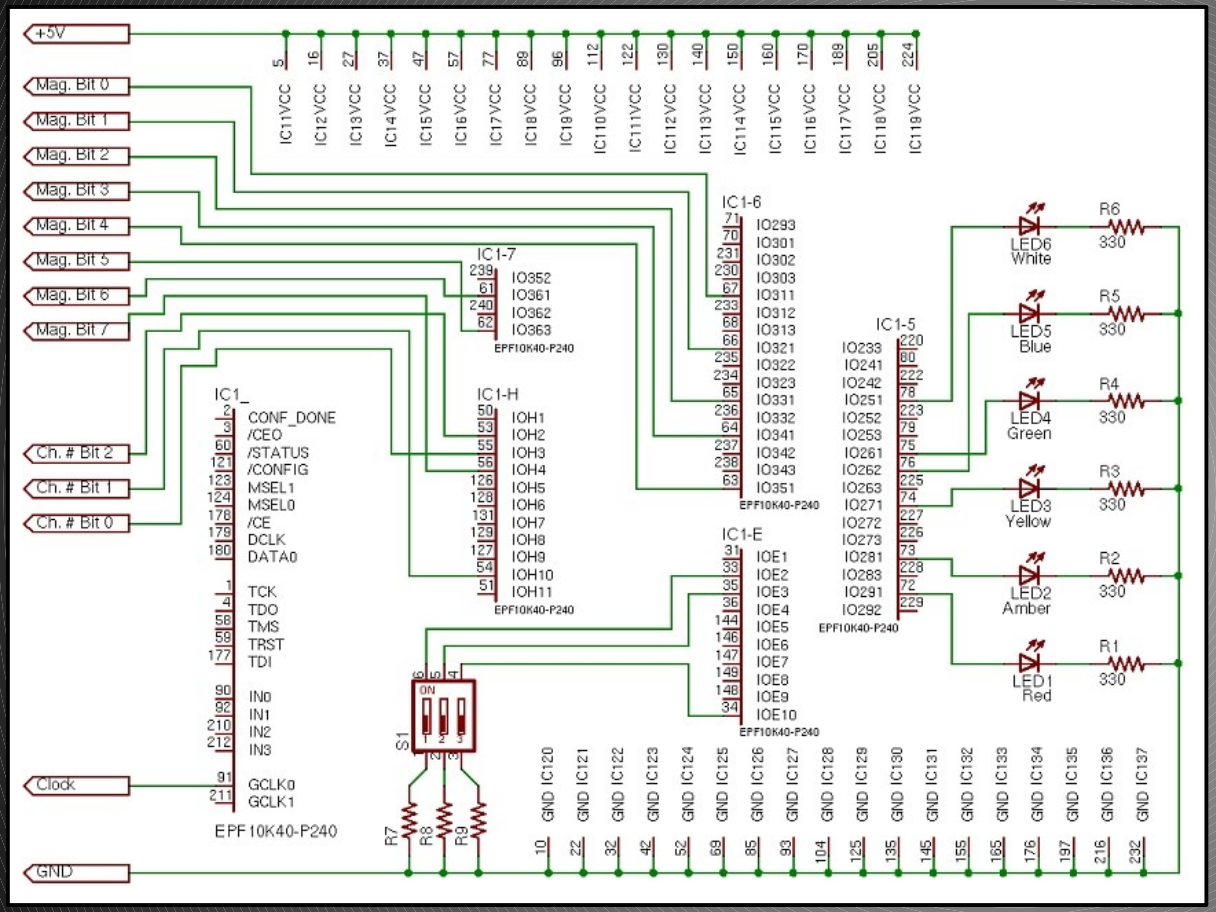


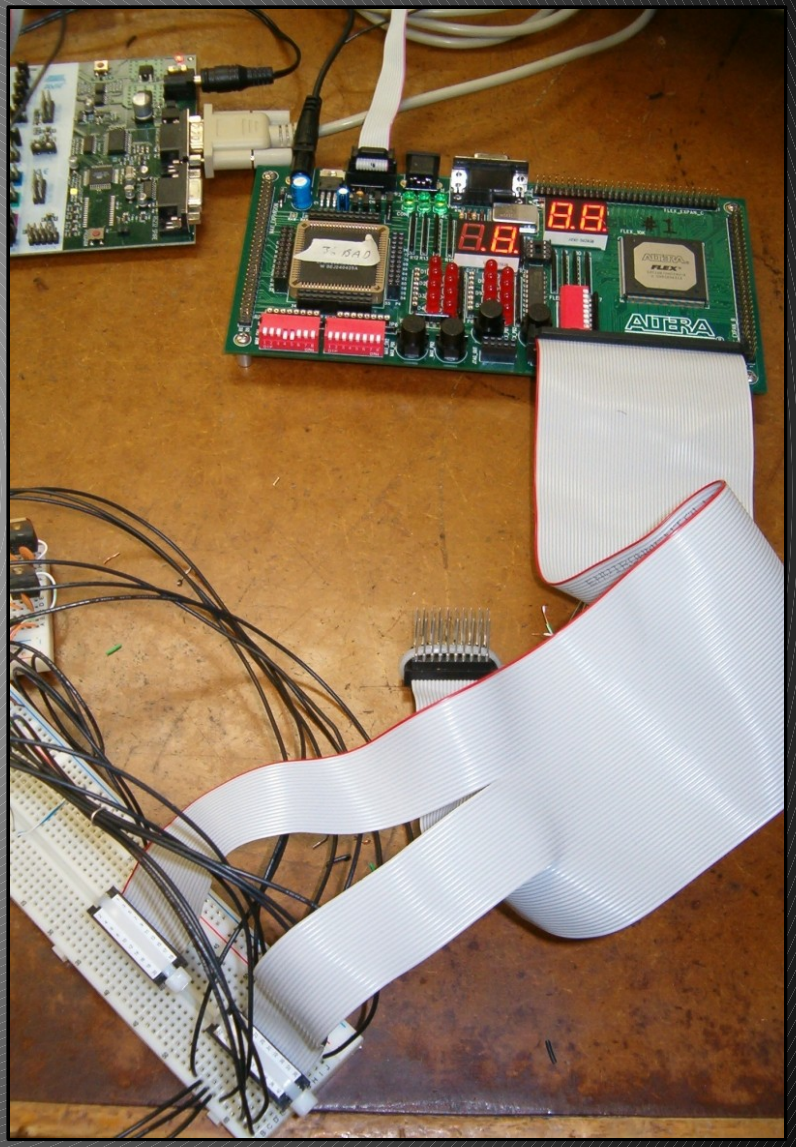


```
if (ADMUX == 0b00100000) {
     PORTB = 0b00000000;    // Send the channel # out on PORTD
     PORTD = adc_res1;      // Send the magnitude # out on PORTB
     ADMUX = 0b00100001;    // Change to the next ADC channel
     ADCSRA |=_BV(ADSC);    // Begin another ADC conversion (set "1")
}
else if (ADMUX == 0b00100001){
     PORTB = 0b00000001;    // Send the channel # out on PORTD
     PORTD = adc_res1;      // Send the magnitude # out on PORTB
     ADMUX = 0b00100010;    // Change to the next ADC channel
     ADCSRA |=_BV(ADSC);    // Begin another ADC conversion (set "1")
}
```

Main [PWM] Stage

The Main Stage collects & processes data from the Conversion Stage. It controls the voltage of each output channel based on that data using Pulse Width Modulation





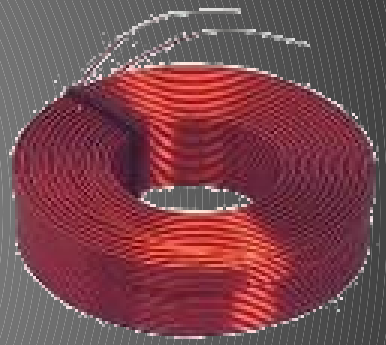

```
pwm := pwm + '1';                    -- Increment the main PWM counter
if (pwm > "11111111") then           -- Check for counter overflow
    pwm := "00000000";               -- Start over again
end if;

                                     -- Channel 0 PWM routine
if (pwm <= per0) then output0 <= '0'; -- Turn off the output
else output0 <= '1';                 -- Now light it up!
end if;

                                     -- Channel 1 PWM routine
if (pwm <= per1) then output1 <= '0'; -- Turn off this output
else output1 <= '1';                 -- Now light it up!
end if;
```

Finishing It Off

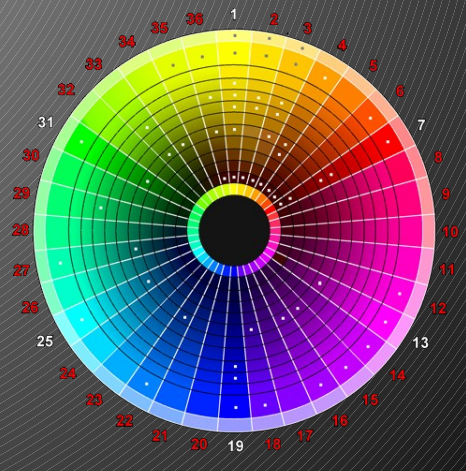
To evolve Lightor from a protoype to a more polished, finished "product" consider:

- **Add a power stage** (to allow the driving of line-voltage, higher power loads

- Use either **real audio-grade inductors** or switch to **RC filters** for tighter frequency control



- **Add averaging capabilities** to the main stages' VHDL code – to reduce jitteriness of the lights



- **More "fun" features**, such as the ability to rotate which outputs respond to which frequencies