**BEAT –SYNC**

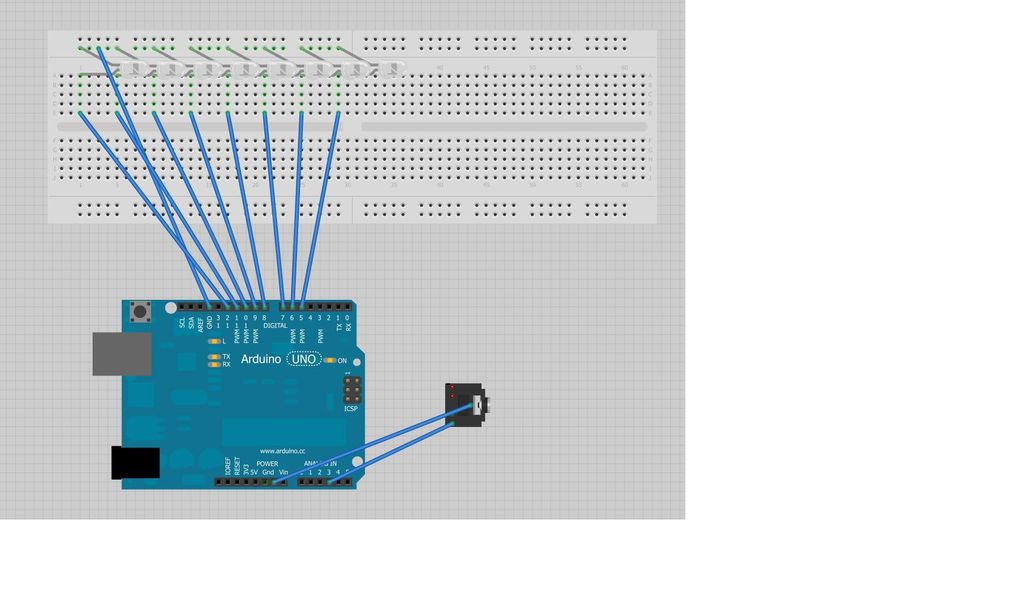
Parts:

RBG LED’s, resistors accordingly, Arduino uno, breadboard, electret microphone, wires/ribbon cables.

Stand for LEDs: cardboard, small-knife, adhesive

1) Make the enclosure: Wood, plastic, cardboard anything can be used, the enclosure in basically a stand which will contain the LED’s, glue the pieces of the material in an interlocking fashion

Appropriate holes can be made into the green part as per the design you want arrange your LED’s in, once all the LED’s are fixed cut out the blue portion and fold the cardboard at 90 degree along the black lines i.e border of the green portion and fit pieces into the slots which will interlock both the folded sides.

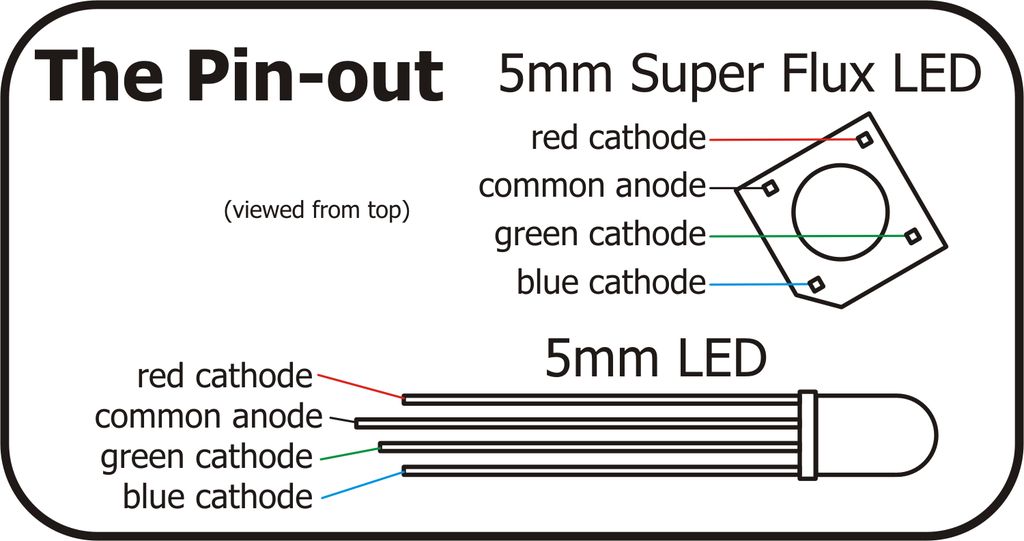
Circuit: 

The circuit is simple enough, although a bit spread when everything is connected

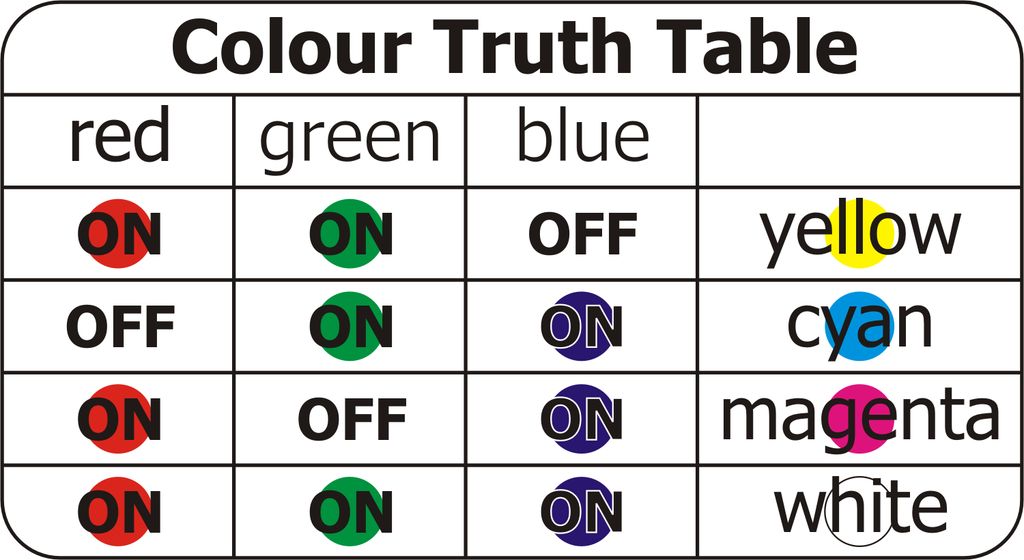
The LED’s have to be connected using limiting resistors as they generally work between 1.5v to 3v and arduino operates at 5v.

270 ohms (red-purple-brown) would be decent fit as it limits the current below 20 mA which is well within the ratings of most of the LED’s. We can use a simple LED or an RBG led.

RGB led is nothing but three LED’s housed in a single package (red, blue and green), it has 4 leads 3 cathode pins for each colour and one common anode pin.



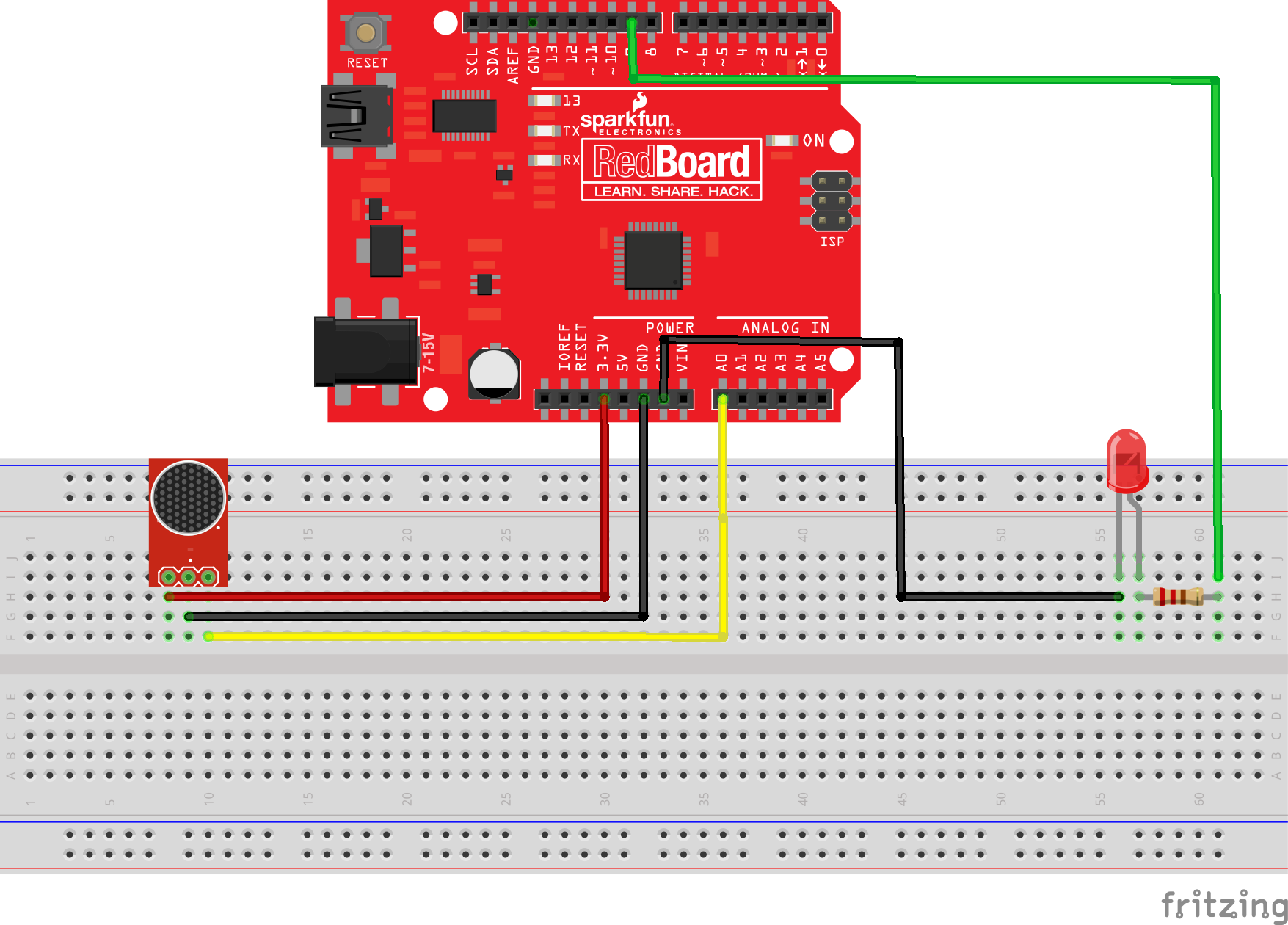
RGB led can be controlled using digital pins which gives the option of choosing from 7 colours (red, blue, green, cyan, yellow, magenta, white)



Or we can also control it using arduino’s PWM pins which lets us create almost any colour moreover lets us fade between colours.

Once all the LED’s are connected we can connect the electret microphone.

Electret microphone is an omnidirectional microphone which reads amplitude and not volume. This microphone is outputting an analog waveform representing the incoming sound.



It is very easy to connect the electret microphone, has only 3 pins

Vcc = 5 volts, Ground = ground, Aud = an analog pin.

Once the electret microphone is in place, the required sound is reaching the arduino in analog format.

(One can also use female audio connector instead of the microphone)

What we are actually doing to isolate frequencies, is using the FAST FOURIER TRANSFORM library

(<http://forum.arduino.cc/index.php/topic,38153.0.html>)

Essentially the fast Fourier transform library which we are using breaks down the audio signal into 14 frequency bands. We will choose the lowest (bass) and highest (treble) because

20hz – 80hz Low Bass

80hz - 320hz Hi Bass, also referred to as Midbass.

320hz - 1280hz Midrange

1280hz - 5120hz High Midrange/Low Treble

5120hz - 20840hz High Treble is a reasonable assumption.

Code is attached in another file.

Once we know the highest and lowest frequency component of the incoming signal we can control the LED’s by using simple IF else statements.