

**Parts List:**

1 – Arduino Uno  
1 – Adafruit NeoPixel  
1 – LED for holo (warm white works well)  
6 – LEDs for radar eye  
1 – 9V to barrel plug  
1 – 1M Ohm resistor (rating varies depending on LED specs)  
Jumper wires

**Arduino Libraries:**

Adafruit Neopixel Library - [https://github.com/adafruit/Adafruit\\_NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel)

**Arduino Sketch:**

```
// NeoPixel Ring simple sketch (c) 2013 Shae Erisson
// released under the GPLv3 license to match the rest of the AdaFruit NeoPixel library

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif

#define PIN    7 //Neopixel Pin
#define PIN_COUNT 6 //Number of LEDs for radar eye
#define UPDATE_DURATION 30 //speed of cylon effect

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS  1

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send signals.
// Note that for older NeoPixel strips you might need to change the third parameter--see the strandtest
// example for more information on possible values.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ400);

//cylon effect variables
int delayval = 500; // delay for half a second
int pins[PIN_COUNT] = { 3, 5, 6, 9, 10, 11 };
```

```

int states[PIN_COUNT];
int current_pin = 0;
int dir = 1;
int update_count = 0;

long loopCount = 0;
long twitchPSITime = 4525;

unsigned long t=0;
int randomPSIFX;
int randomHOLOFX;

// the setup function runs once when you press reset or power the board

void setup()
{
// initialize digital pin 13 as an output for LED
pinMode(13, OUTPUT);

// initialize NeoPixel library
pixels.begin();

//CYLON
for ( int i = 0; i < PIN_COUNT; i++ ) {
pinMode(pins[i], OUTPUT);
states[i] = 0;
}
}

// the loop function runs over and over again forever

void loop() {

loopCount++;
t=millis();
if (t > twitchPSITime)
{
twitchPSI(); // call playRandom routine
twitchHOLO(); // call random routing
loopCount = 0; // reset loopCount
twitchPSITime = (random(1,10)*500)+millis(); // set the next twitchTime
}

decay();
states[current_pin] = 255 * update_count / UPDATE_DURATION;
updatePins();
}

```

```

update_count++;
if ( update_count > UPDATE_DURATION ) {
    update_count = 0;
    current_pin += dir;
    if ( current_pin == 0 ) {
        dir = 1;
    }
    else if ( current_pin == (PIN_COUNT-1) ) {
        dir = -1;
    }
}

}

void twitchPSI() {
    randomPSIFX = random (1,7); // Pick a number 1 - 6
    switch (randomPSIFX) {
    case 1:
        pixels.setPixelColor(0,pixels.Color(255,255,255)); //white
        break;
    case 2:
        pixels.setPixelColor(0,pixels.Color(0,0,0)); //off
        break;
    case 3:
        pixels.setPixelColor(0,pixels.Color(0,255,255)); //cyan
        break;
    case 4:
        pixels.setPixelColor(0,pixels.Color(255,0,255)); //magenta
        break;
    case 5:
        pixels.setPixelColor(0,pixels.Color(255,255,0)); //yellow
        break;
    case 6:
        pixels.setPixelColor(0,pixels.Color(0,255,0)); //green
        break;
    case 7:
        pixels.setPixelColor(0,pixels.Color(0,0,255)); //blue
        break;
    case 8:
        pixels.setPixelColor(0,pixels.Color(255,0,0)); //red
        break;

    default:
        // if nothing else matches, do the default
        // so we are going to do nothing... for that matter not even waste time
        break;
}
}

```

```

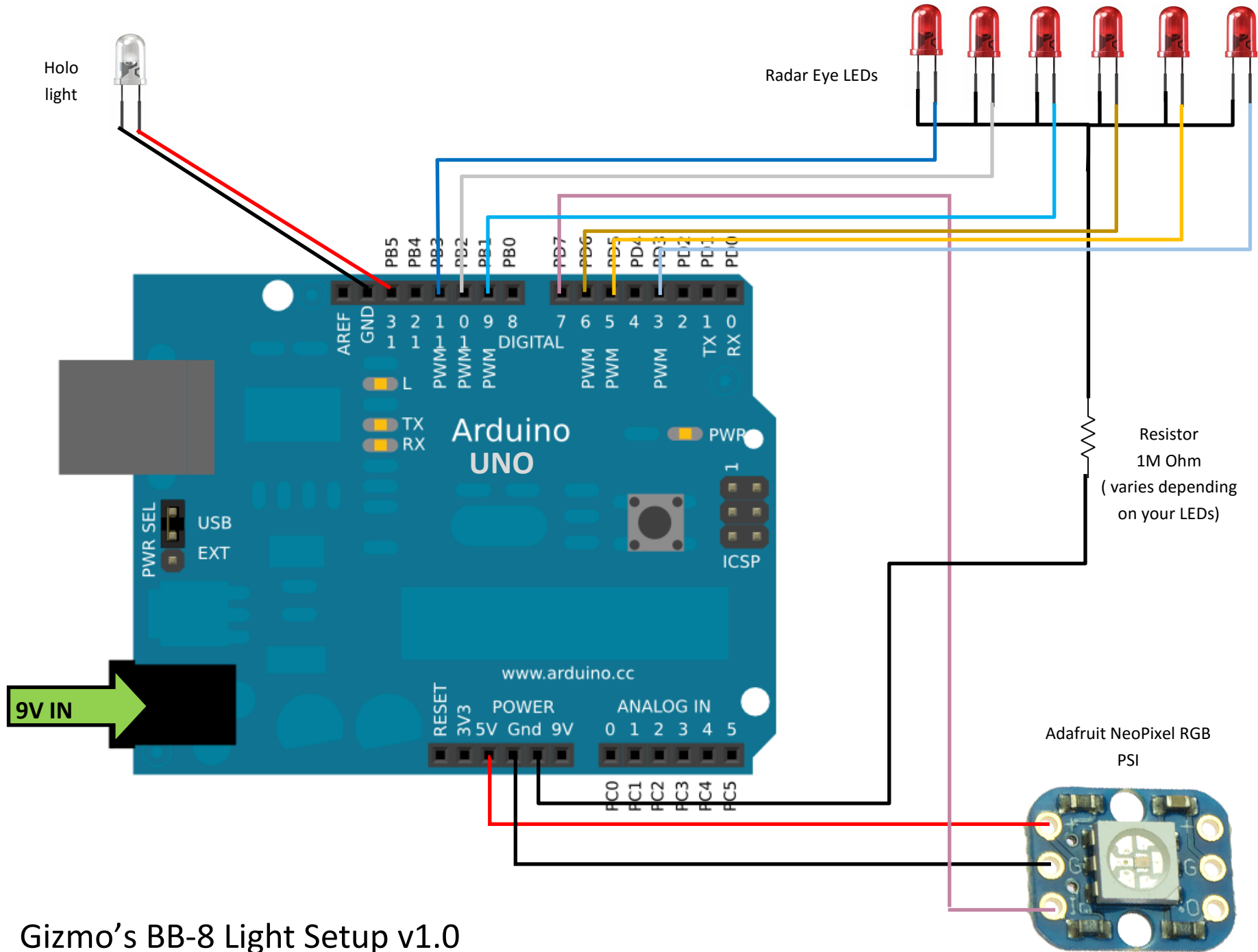
    }
    pixels.show(); // This sends the updated pixel color to the hardware.
}

void twitchHOLO() {
    randomHOLOFX = random (1,3); // Pick a number 1 - 2
    switch (randomHOLOFX) {
    case 1:
        digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
        break;
    case 2:
        digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
        break;
    default:
        // if nothing else matches, do the default
        // so we are going to do nothing... for that matter not even waste time
        break;
    }
}

void updatePins() {
    for ( int i = 0; i < PIN_COUNT; i++ ) {
        analogWrite(pins[i], states[i]);
    }
    delay(6);
}

void decay() {
    for ( int i = 0; i < PIN_COUNT; i++ ) {
        states[i] = (19*states[i]/20);
    }
}

```



Gizmo's BB-8 Light Setup v1.0