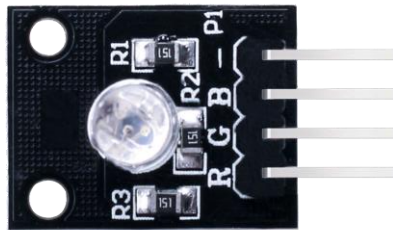

RGB LED Module

DESCRIPTION:

RGB LED module consists of a full-color LED made by R, G, B three pin PWM voltage input can be adjusted. Primary colors (red / blue / green) strength in order to achieve full color mixing effect. Control of the module with the Arduino can be achieved Cool lighting effects. And three resistor has been soldered on this board, so needn't worry to burn out the LED light under the long time work.



Specification:

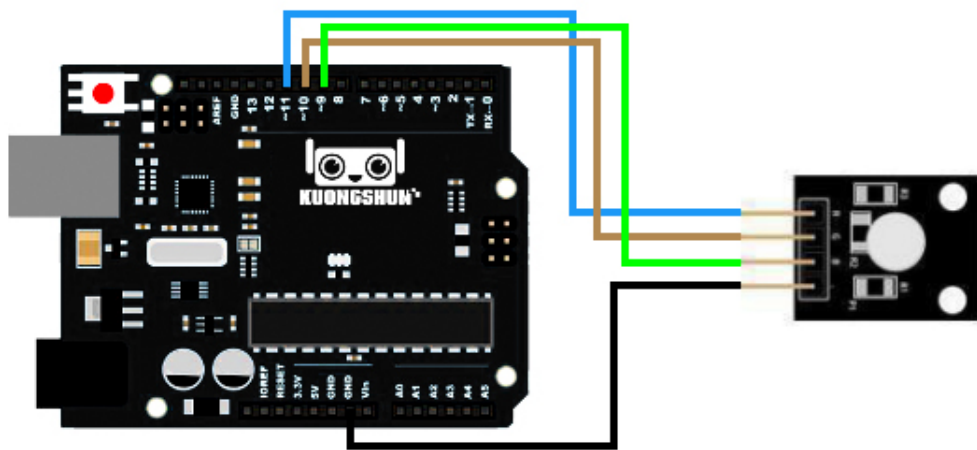
- Red Vf: 1.8 to 2.1V
- Green Vf: 3.0 to 3.2V
- Blue Vf: 3.0 to 3.2V
- Red color: 620-625 nm
- Green color: 520-525 nm
- Blue color: 465-470 nm
- Red brightness @ ~20mA: 600-800 mcd
- Blue brightness @ ~20mA: 800-1000 mcd
- Green brightness @ ~20mA: 1500-2000mcd

PIN CONFIGURATION:

- 1、 “R”: Red light
- 2、 “G” : Green light
- 3、 “B” : Blue light
- 4、 “-”:Ground

Example:

In this example, we blink an LED and using an RGB LED we can generate any color we



Code:

```
#define BLUE 11
#define GREEN 10
#define RED 9

void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
```

```
digitalWrite(GREEN, LOW);
digitalWrite(BLUE, LOW);
}

// define variables
int redValue;
int greenValue;
int blueValue;

// main loop
void loop()
{
#define delayTime 10 // fading time between colors

redValue = 255; // choose a value between 1 and 255 to change the color.
greenValue = 0;
blueValue = 0;

// this is unnecessary as we've either turned on RED in SETUP
// or in the previous loop ... regardless, this turns RED off
// analogWrite(RED, 0);
// delay(1000);

for(int i = 0; i < 255; i += 1) // fades out red bring green full when i=255
{
redValue -= 1;
greenValue += 1;
// The following was reversed, counting in the wrong directions
// analogWrite(RED, 255 - redValue);
```

```
// analogWrite(GREEN, 255 - greenValue);
analogWrite(RED, redValue);
analogWrite(GREEN, greenValue);
delay(delayTime);
}

redValue = 0;
greenValue = 255;
blueValue = 0;

for(int i = 0; i < 255; i += 1) // fades out green bring blue full when i=255
{
greenValue -= 1;
blueValue += 1;
// The following was reversed, counting in the wrong directions
// analogWrite(GREEN, 255 - greenValue);
// analogWrite(BLUE, 255 - blueValue);
analogWrite(GREEN, greenValue);
analogWrite(BLUE, blueValue);
delay(delayTime);
}

redValue = 0;
greenValue = 0;
blueValue = 255;

for(int i = 0; i < 255; i += 1) // fades out blue bring red full when i=255
{
// The following code has been rearranged to match the other two similar sections
```

```
blueValue -= 1;
redValue += 1;
// The following was reversed, counting in the wrong directions
// analogWrite(BLUE, 255 - blueValue);
// analogWrite(RED, 255 - redValue);
analogWrite(BLUE, blueValue);
analogWrite(RED, redValue);
delay(delayTime);
}
}
```