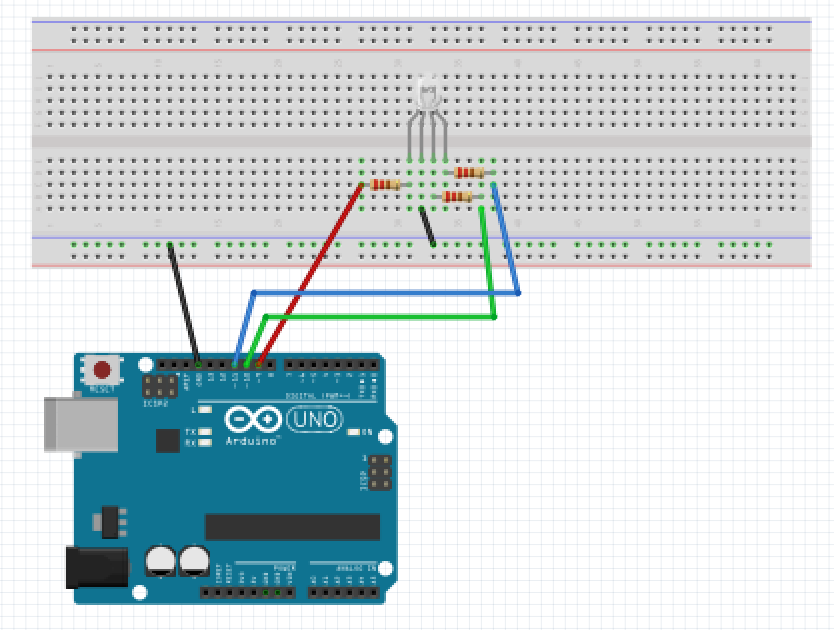
**ARDUINO LED PROJECT**

**RGB LED**



For this project we will need:

* Arduino board.
* RGB led
* Breadboard.
* 7 Male to Male wires.
* 3 470 Ohm resistor
* Temperature sensor

Circuit Design:

1. First make sure that the Arduino is powered off (no USB cable plugged to power).
2. Check the LED, you will see that one of the leg is longer than the other 3.
3. Plug the longer leg of the LED to a hole on the breadboard. Connect that leg to the Ground pin on Arduino., using a black cable if possible (convention for GND).
4. Plug the 1st short leg of the LED to a different hole, on a different and independent line of the breadboard.
5. Add a 470 Ohm resistor between this longer leg and pin 9(has ~ ) of the Arduino, using an additional colored wire (red) for convenience.
6. Plug the 3rd short leg of the LED to a different hole, on a different and independent line of the breadboard.
7. Add a 470 Ohm resistor between this longer leg and pin 10(has ~ ) of the Arduino, using an additional colored wire (green) for convenience.
8. Plug the 4th short leg of the LED to a different hole, on a different and independent line of the breadboard.
9. Add a 220/470 Ohm resistor between this longer leg and pin 11(has ~ ) of the Arduino, using an additional colored wire (blue) for convenience.
10. Plug the common ground of the breadboard to the ground pin of Arduino.

|  |
| --- |
| int redPin=9;  int greenPin=10;  int bluePin=11;  void setup(){  pinMode(redPin,OUTPUT);  pinMode(greenPin,OUTPUT);  pinMode(bluePin, OUTPUT);  Serial.begin(9600);  }  void loop(){  setColor(255,0,0); //red color  delay(100);  setColor(0,255,0); //green color  delay(100);  setColor(0,0,255); //blue color  delay(100);    }  void setColor(int red, int green, int blue){  analogWrite(redPin ,blue);  analogWrite(greenPin ,green);  analogWrite(bluePin ,blue);  }  } |

int redPin=9;

int greenPin=10;

int bluePin=11;

We instruct the Arduino to use these values as the pins attached to the external LED.The pins are attached to pin 1,3,4.

pinMode (redPin,OUTPUT);

pinMode (greenPin, OUTPUT);

pinMode (bluePin, OUTPUT);

This sets the LED pins functionality to output. Orientation of the LED helps to determine color changes.

setColor(0,0,255);

The pins attached to 1,3,4 of the led are unique. They allow us to output analog signals.

The values passed are in the range of 0-255.This will vary the brightness of the specific color.

These particular sets of instructions are likely to be repeated during execution.

Instead of rewriting them we can create a function that will execute this set of instructions.

void setColor(int red, int green, int blue){

This creates a function.

The function receives 3 values red, green ,blue.

The 3 values are considered values fed to the function.

In order to use this function we must pass 3 corresponding values.

void shows that this function does not return any value when executed.

Same with the setup(),loop() functions they do not return any values when executed.

analogWrite(redPin,blue);

analogWrite(greenPin,green);

analogWrite(bluePin,blue);

This assigns the pins of the LED to a particular brightness on corresponding pins.

This combination of LED brightness allows us to modify the color displayed.

This pin has PWM allowing it to write both digital and analog signals.

This sends the brightness value to the attached LED pin.

void setup (){}

This initializes the arduino and assigns functionality to its pins.

This also provides required resources for monitoring.

void loop(){}

After executing the void setup() function, we enter the void loop() and this function is executed continuously and repeatedly, until your Arduino is powered off.