# Overview

RGB LED modules can emit various colors of light. Three LEDs of red, green, and blue are packaged into a transparent or semitransparent plastic shell with four pins led out. The three primary colors of red, green, and blue can be mixed and compose all kinds of colors by brightness, so you can make an RGB LED emit colorful light by controlling the circuit.

# Step 1: Connect It

Use the following wiring diagram to connect the sensor to your Raspberry Pi



**Important Notes:**

* The wire colors do not matter. Just use whatever is available at your station.
* **Make sure that you are plugging the wires into the correct location on both the Pi, the Sensor, and the analog to digital converter (especially the power and ground)**
* The wires move around a lot, so make sure they are firmly in place when you are testing

# Step 2: Code It

We have provided you with a Python code template that can run this sensor/device. The code is located at:

**/home/pi/Documents/rpi-iot-demos/rgb.py**

Use the Geany editor to open this file (NOTE: this will be slow, so be patient). Once it is open, take some time to read the code to see how it works.

Once you are ready, do the following:

1. Find the function called loop().
   1. This function starts when the script runs, and runs forever
2. Find the list called colors (it’s at the top).
   1. This list contains a list a series of hex values (0-F) specifying a color
   2. Format: 0xRRGGBB
      1. 0xFFFFFF is white
      2. 0xFF0000 is red
      3. 0x00FF00 is green
      4. 0x0000FF is blue
3. Modify this function so that it displays all of the colors in the colors list

def loop():

while True:

for col in colors:

setColor(col)

time.sleep(1)

# Step 3: Run / Test It

1. To run the script, open a Linux console and navigate to the folder with your code
   1. Helpful Linux Commands:  
      **ls** lists the contents of the directory  
      **cd <folder\_name>** opens a folder (don’t type the < > characters)  
      **cd ..** exits the folder you are currently in  
      **cd ~** takes you back to your home folder
2. Run the python program by typing the following:

**python3 rgb.py**

1. Watch the light bulb!
   1. If it doesn’t work, try getting closer to the microphone
   2. You can update the colors array to produce any color you want!
2. **To exit the program, press Ctrl-C in the terminal**

# Step 4: Make It Controllable

To make your program listen for MQTT messages, modify your main program to look like the following:

# --------------------------------------------

# Main Program Starts Here

# --------------------------------------------

if \_\_name\_\_ == "\_\_main\_\_":

try:

comm.connect(channels=[("jbcs/rgb", 0)])

comm.listen(on\_message)

setup(R, G, B)

print("RGB Running. Enjoy the light show!")

loop()

except KeyboardInterrupt:

destroy()

Now, create a new function above your main program. This function will process any messages transmitted to your program.

# --------------------------------------------

# This is the code that will be run every time

# a message is received

# --------------------------------------------

def on\_message(client, userdata, msg):

global active

print("Received:", msg)

try:

hex\_int = int(msg, 16)

colors.clear()

colors.append(hex\_int)

except:

print("Problem encountered")