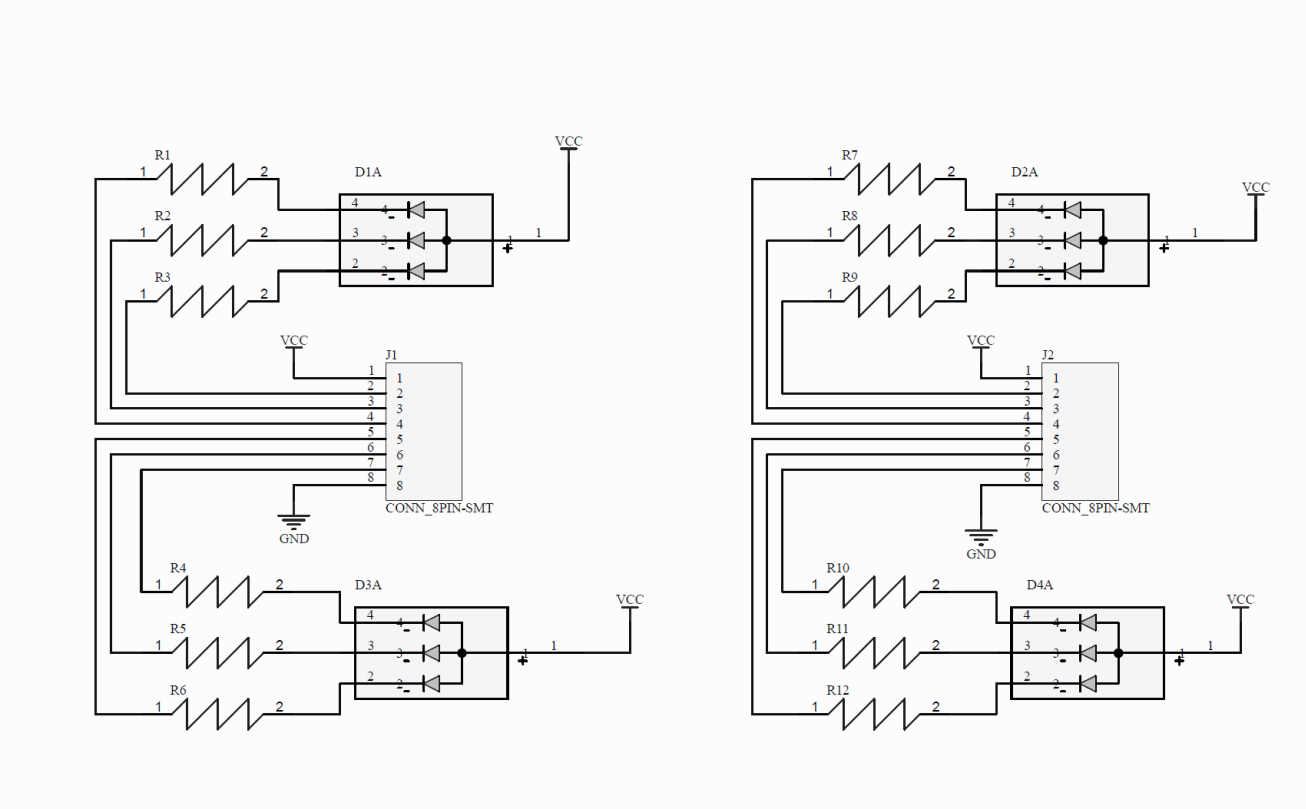
EARTH PEOPLE TECHNOLOGY

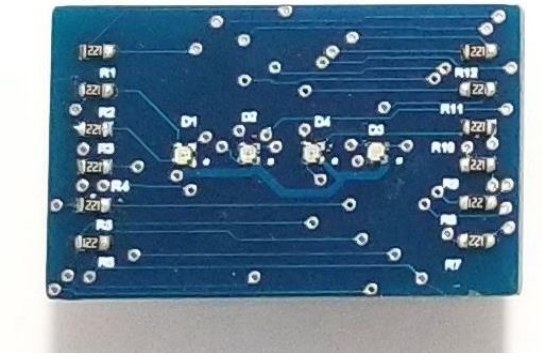
EPT-SML-LX04-LD-X2 LED RGB Breakout Board User Manual

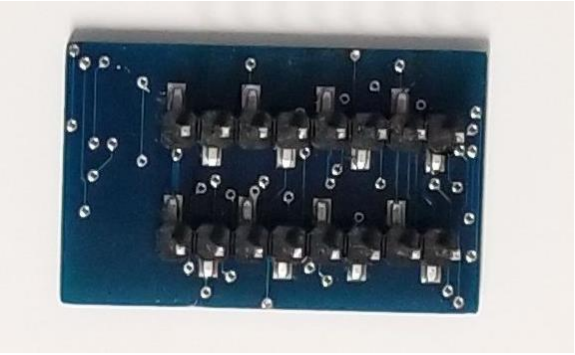
The EPT-SML-LX04-LD-X2 is a breakout board that includes four LED RGBs. This Board provides a simple interface to control four LED RGBs for a total of 12 LEDs from any MCU (including the Arduino family). It is designed to connect directly to the standard bread board with a reduced footprint. Once connected to an MCU, the MCU can provide a sink for each LED.

Description

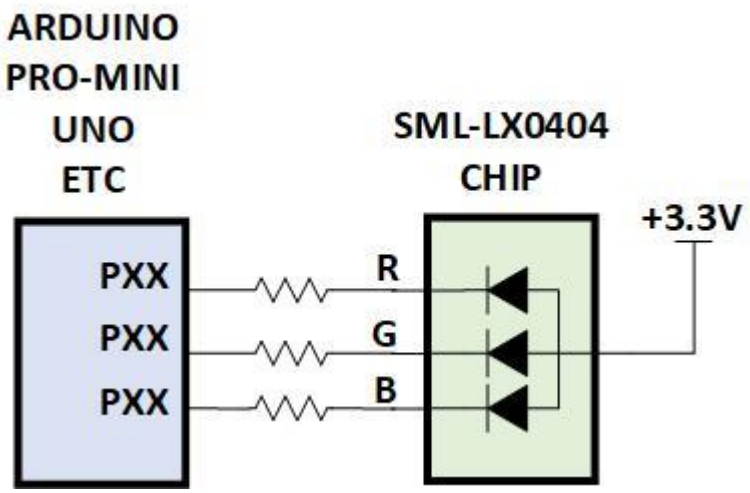
Compact and convenient breakout board includes four SML-LX0404 LED RGB chips. Each LED is attached to a current limiting resistor. Each LED leg is controlled by a pin from one of two SMT connectors on the bottom side of the board.







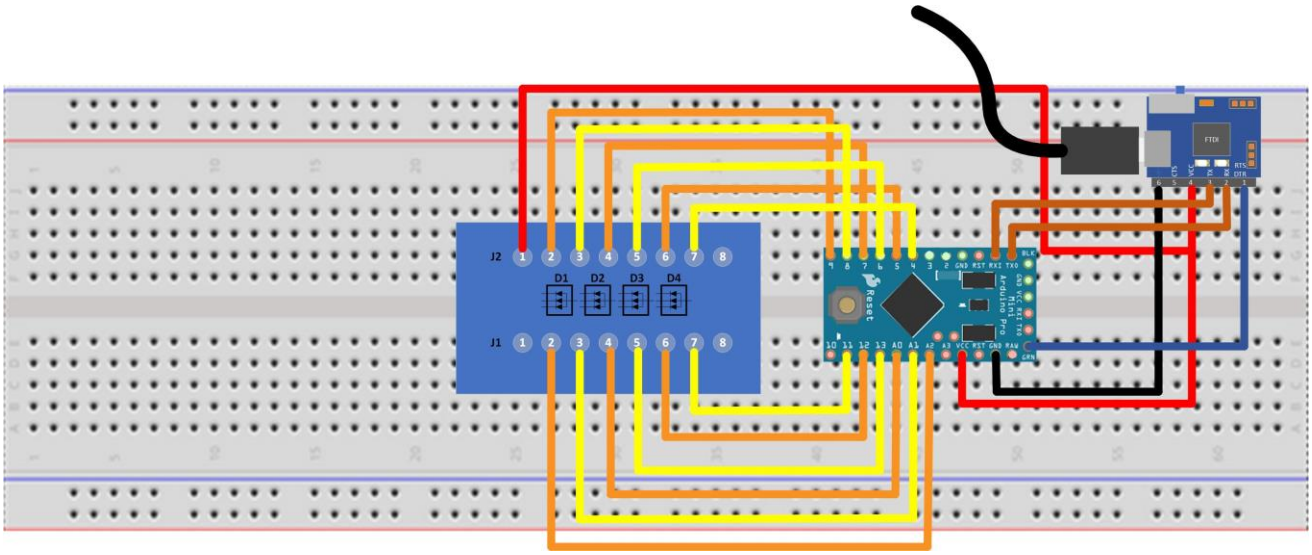
Any GPIO from any MCU (including Arduino family) can be used to control the On/Off of each LED individually. This Breakout board is compatible with both +3.3V and +5V Arduinos. However, the LED RGB Breakout board is for use only with +3.3V.



The SML-LX0404 chip is a current sink and can be connected to any MCU. The anode should be connected to +3.3V. The reason for this is the Series resistors are calculated for current limiting based on +3.3V. Changing the anode voltage to another source will change the brightness emitting from each LED. Because the SML-LX0404 chips are current sink, the user can connect to either +5V or +3.3V Arduino (or other MCUs) and control each LED.

Connecting the Arduino

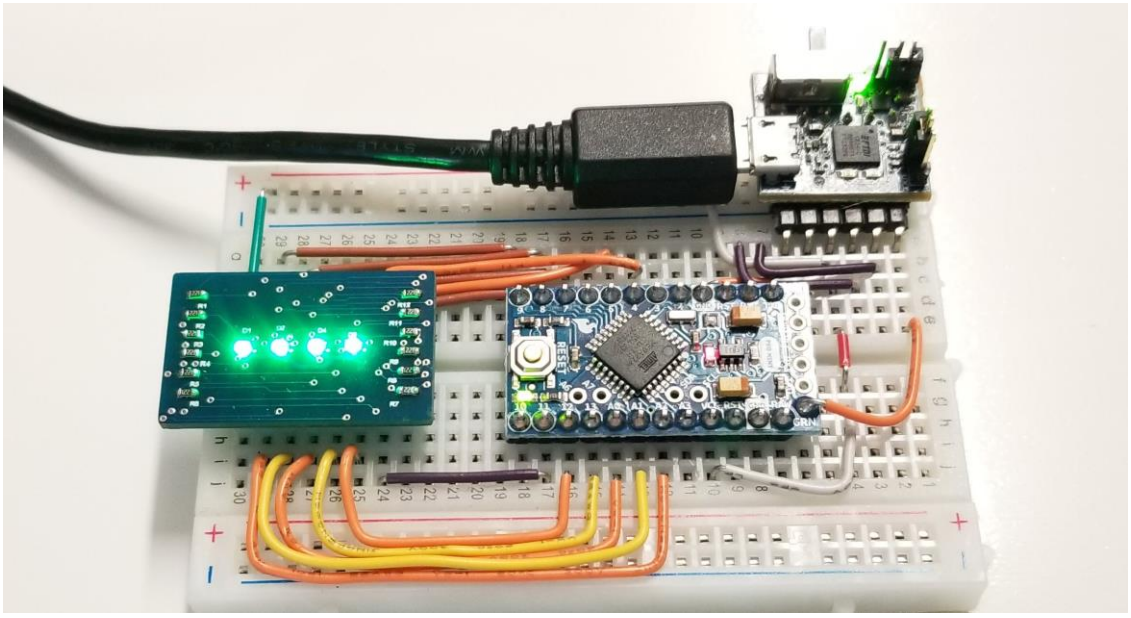
Connecting the Arduino (or any MCU) to the LED RGB is easy. In this example, we use typical bread board, a pro-mini and a VisiPort USB to Serial board.



Arduino Pro-Mini			LED RGB Breakout Board		
Signal	**Port Number**		**Signal Name**	**Pin Number**	**Connector**
4	PD4		D4-Green	7	J2
5	PD5		D4-Blue	6	
6	PD6		D4-Red	5	
7	PD7		D2-Green	4	
8	PB0		D2-Blue	3	
9	PB1		D2-Red	2	
11	PB3		D3-Green	7	J1
12	PB4		D3-Blue	6	

13	PB5		D3-Red	5	
A0	PC0		D1-Green	4	
A1	PC1		D1-Blue	3	
A2	PC3		D1-Red	2	

Arduino Pro-Mini			**USB to Serial FTDI Basic**		
VCC	NA		VCC	4	J1
GND	NA		GND	6	
DTR	NA		DTR	1	
RXI	PD0		TX	3	
TXO	PD1		RX	2	



Arduino Code – Controlling On/Off of LEDs

When you have everything hooked up try running the below sketch. This is a simple loop that will blink each LED individually.

```
/*
  Copyright Earth People Technology Inc. 2022

  LED RGB Breakout Board Blinky

  Platform: Arduino Mini

*/
```

```
  int ledD1GreenPin = A0;
  int ledD1RedPin = A2;
  int ledD1BluePin = A1;
  int ledD2GreenPin = 7;
  int ledD2RedPin = 9;
  int ledD2BluePin = 8;
  int ledD3GreenPin = 11;
  int ledD3RedPin = 13;
  int ledD3BluePin = 12;
  int ledD4GreenPin = 4;
  int ledD4RedPin = 6;
  int ledD4BluePin = 5;
  int LoopCount;

void setup()
{
  pinMode(ledD1GreenPin, OUTPUT);
  pinMode(ledD1RedPin, OUTPUT);
  pinMode(ledD1BluePin, OUTPUT);
  pinMode(ledD2GreenPin, OUTPUT);
  pinMode(ledD2RedPin, OUTPUT);
  pinMode(ledD2BluePin, OUTPUT);
  pinMode(ledD3GreenPin, OUTPUT);
  pinMode(ledD3RedPin, OUTPUT);
  pinMode(ledD3BluePin, OUTPUT);
  pinMode(ledD4GreenPin, OUTPUT);
  pinMode(ledD4RedPin, OUTPUT);
  pinMode(ledD4BluePin, OUTPUT);

  digitalWrite(ledD1GreenPin, HIGH);
  digitalWrite(ledD1RedPin, HIGH);
  digitalWrite(ledD1BluePin, HIGH);
  digitalWrite(ledD2GreenPin, HIGH);
  digitalWrite(ledD2RedPin, HIGH);
  digitalWrite(ledD2BluePin, HIGH);
  digitalWrite(ledD3GreenPin, HIGH);
```

```
  digitalWrite(ledD3RedPin, HIGH);
  digitalWrite(ledD3BluePin, HIGH);
  digitalWrite(ledD4GreenPin, HIGH);
  digitalWrite(ledD4RedPin, HIGH);
  digitalWrite(ledD4BluePin, HIGH);

  Serial.begin(115200);          // start serial communication at 115200bps

  LoopCount = 0;

  Serial.println("LED RGB Communications Active!");

}

void loop ()
{

      delay(500); //Delay 500 ms

      //Set the LED Green Pin Low for D1, D2, D3 and D4
      //to turn on all Green LEDs
      digitalWrite(ledD1GreenPin, LOW);
      digitalWrite(ledD2GreenPin, LOW);
      digitalWrite(ledD3GreenPin, LOW);
      digitalWrite(ledD4GreenPin, LOW);

      delay(500); //Delay 500 ms

      //Set the LED Green Pin High for D1, D2, D3 and D4
      //to turn off all Green LEDs
      digitalWrite(ledD1GreenPin, HIGH);
      digitalWrite(ledD2GreenPin, HIGH);
      digitalWrite(ledD3GreenPin, HIGH);
      digitalWrite(ledD4GreenPin, HIGH);

      delay(500); //Delay 500 ms
```

```
      //Set the LED Red Pin Low for D1, D2, D3 and D4
      //to turn on all Red LEDs
      digitalWrite(ledD1RedPin, LOW);
      digitalWrite(ledD2RedPin, LOW);
      digitalWrite(ledD3RedPin, LOW);
      digitalWrite(ledD4RedPin, LOW);

      delay(500); //Delay 500 ms

      //Set the LED Red Pin High for D1, D2, D3 and D4
      //to turn off all Red LEDs
      digitalWrite(ledD1RedPin, HIGH);
      digitalWrite(ledD2RedPin, HIGH);
      digitalWrite(ledD3RedPin, HIGH);
      digitalWrite(ledD4RedPin, HIGH);

      delay(500); //Delay 500 ms

      //Set the LED Blue Pin Low for D1, D2, D3 and D4
      //to turn on all Blue LEDs
      digitalWrite(ledD1BluePin, LOW);
      digitalWrite(ledD2BluePin, LOW);
      digitalWrite(ledD3BluePin, LOW);
      digitalWrite(ledD4BluePin, LOW);

      delay(500); //Delay 500 ms

      //Set the LED Blue Pin Low for D1, D2, D3 and D4
      //to turn off all Blue LEDs
      digitalWrite(ledD1BluePin, HIGH);
      digitalWrite(ledD2BluePin, HIGH);
      digitalWrite(ledD3BluePin, HIGH);
      digitalWrite(ledD4BluePin, HIGH);

      Serial.println(LoopCount++);    // print the LoopCount

}
```

Note that you must set your serial monitor to a speed of 115200 baud to try out the sketch.

Code Explanation:

This sketch simply applies a low to the selected LED to turn it on. Then applies a high to all other LEDs to turn them off. First, 12 signals are declared and each signal is assigned a name and a unique pin on the Pro-Mini. Initially, all 12 LED control signals are declared as outputs in the Setup function and set to high. This turns off all 12 LEDs.

During the Loop function, a delay of 500 milliseconds is applied. No output/LED is affected during the delay. Then, all four Green LEDs are set to low which turns them all on. All other LEDs are still set to high from the initial setup. A delay of 500 milliseconds is applied. During this delay, the Green LEDs remain on, all other LEDs remain off and no actions are taken on any outputs. Next, all four Green LEDs are set to high which turns them off. All other LEDs are still set to high/off. Another delay of 500 milliseconds is applied in which no output is changed. This cycle repeats for the Red and Blue LEDs. Once the code gets to the bottom of the loop, a counter variable called LoopCount is incremented by one and then transmitted to the Serial Port. The loop function starts over at the top of the loop code and repeats the cycle. This repeating cycle goes on forever until the power is removed from the Pro-Mini.