RGB LED 8×8 模块

BMD26M088
Arduino Library V1.0.1 说明

版本：V1.20　日期：2023-10-24

www.bestmodulescorp.com

目录

简介

BMD26M088 是倍创推出的 RGB LED 8×8 模块，使用 I²C 通信方式。本文档对 BMD26M088 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了灯板显示功能。

Arduino Library

Arduino Lib 名称：BMD26M088		Lib 版本：V1.0.1
构造函数 & 初始化 & 复位		
1	BMD26M088(TwoWire *theWire=&Wire)	
	描述	构造函数，使用 Wire 通信接口
	参数	*theWire：选择 Wire 接口 (默认 Wire 接口)
	返回值	—
	备注	—
2	void begin(uint32_t i2c_addr=0x67, uint32_t i2c_Clock=400000)	
	描述	模块初始化
	参数	i2c_addr：I²C 从机地址，默认地址 0x67 i2c_Clock：I²C 通信速率，默认 400kHz
	返回值	void
	备注	模块内部默认设置亮度为最亮，恒流率为 6mA
功能函数		
3	void reset(uint8_t i2c_addr=0x67)	
	描述	模块复位
	参数	i2c_addr：I²C 通信地址，默认地址 0x67
	返回值	void
	备注	—
4	bool isConnected(uint8_t i2c_addr)	
	描述	判断 I²C 通信是否连接成功
	参数	i2c_addr：I²C 通信地址
	返回值	连接状态： true：连接成功 false：连接失败
	备注	—
5	void writeCmd(uint8_t i2c_addr, uint8_t cmd, uint8_t data)	
	描述	写命令
	参数	i2c_addr：I²C 通信地址 cmd：命令 data：数据
	返回值	void
	备注	cmd 见 HT16D33B 规格书

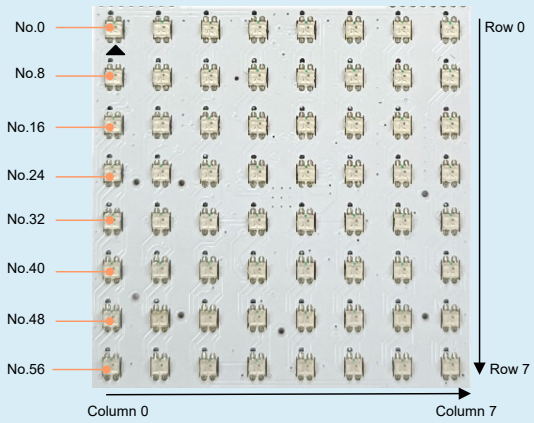
6	uint8_t readCmd(uint8_t i2c_addr,uint8_t cmd)	
	描述	读取寄存器的命令
	参数	i2c_addr：I²C 通信地址 cmd：命令
	返回值	命令下读取的返回值
	备注	cmd 见 HT16D33B 规格书
7	void setOverTempertaureProtect(uint8_t i2c_addr,uint8_t state,bool auto_control=true)	
	描述	设置芯片内部过温保护功能
	参数	i2c_addr：I²C 通信地址 state：过温保护开关 1：打开 0：关闭 auto_control：触发过温保护后自动关闭 LED 显示 (默认为 true) true：自动关闭 false：不自动关闭
	返回值	void
	备注	进入过温保护的阈值为 150℃，退出过温保护状态的阈值为 125℃
8	bool getOverTemperatureFlag(uint8_t i2c_addr)	
	描述	获取过温触发状态标志位
	参数	i2c_addr：I²C 通信地址
	返回值	过温触发状态 true：触发 false：未触发
	备注	过温高于 150℃ 标志位置高，低于 125℃ 清零； 当设置为过温不自动关闭，则需要读取标志确认是否关闭 LED 显
9	void setLedRAMEnable(uint8_t i2c_addr)	
	描述	使能 LED 的 RAM 存储器
	参数	i2c_addr：I²C 通信地址
	返回值	void
	备注	开启 RAM 后，只要往对应 RAM 地址里写入数据即可点亮对应的灯珠 请参考使用手册的 RAM 地址映射章节
10	void setLedRAMDisable(uint8_t i2c_addr)	
	描述	除能 LED 的 RAM 存储器
	参数	i2c_addr：I²C 通信地址
	返回值	void
	备注	请参考使用手册的 RAM 地址映射章节
11	void setBrightness(uint8_t i2c_addr, uint8_t brightness)	
	描述	设置全局亮度
	参数	i2c_addr：I²C 通信地址 brightness：亮度，范围 0~255，数值越大亮度越高
	返回值	void
	备注	—

12	void setCurrent(uint8_t i2c_addr, uint8_t level)	
	描述	设置恒流大小
	参数	i2c_addr：I²C 通信地址 level：恒流率 0(BMD26M088_CCR_3MA): 3mA 1(BMD26M088_CCR_6MA): 6mA 2(BMD26M088_CCR_9MA): 9mA 3(BMD26M088_CCR_12MA): 12mA 4(BMD26M088_CCR_15MA): 15mA 5(BMD26M088_CCR_18MA): 18mA 6(BMD26M088_CCR_21MA): 21mA 7(BMD26M088_CCR_24MA): 24mA 8(BMD26M088_CCR_27MA): 27mA 9(BMD26M088_CCR_30MA): 30mA 10(BMD26M088_CCR_33MA): 33mA 11(BMD26M088_CCR_36MA): 36mA 12(BMD26M088_CCR_39MA): 36mA 13(BMD26M088_CCR_42MA): 42mA 14(BMD26M088_CCR_45MA): 45mA 15(BMD26M088_CCR_48MA): 48mA
	返回值	void
	备注	—
13	void clearAll(uint8_t i2c_addr)	
	描述	清除所有显示
	参数	i2c_addr：I²C 通信地址
	返回值	void
	备注	—
14	void clearRGB(uint8_t i2c_addr, uint8_t RGB_Number)	
	描述	清除指定 RGB 的显示
	参数	i2c_addr：I²C 通信地址 RGB_Number：RGB 序号 (3) 0~63
	返回值	void
	备注	—
15	void clearRow(uint8_t i2c_addr, uint8_t RowIndex)	
	描述	清除一行 RGB 的显示
	参数	i2c_addr：I²C 通信地址 RowIndex：行序号 (2) 0~7
	返回值	void
	备注	—
16	void clearColumn(uint8_t i2c_addr, uint8_t ColumnIndex)	
	描述	清除一列 RGB 的显示
	参数	i2c_addr：I²C 通信地址 ColumnIndex：列序号 (1) 0~7
	返回值	void
	备注	—

17	void writeRGB(uint8_t i2c_addr, uint8_t RGB_Number, uint8_t R, uint8_t G, uint8_t B)	
	描述	点亮指定的 RGB 灯
	参数	i2c_addr：I²C 通信地址 RGB_Number：RGB 序号 (3) 0~63 R/G/B：颜色数值
	返回值	void
	备注	—
18	void writeAllRGB(uint8_t i2c_addr, int8_t R, uint8_t G, uint8_t B)	
	描述	点亮所有的 RGB 灯
	参数	i2c_addr：I²C 通信地址 R/G/B：颜色数值
	返回值	void
	备注	—
19	void writeColumn(uint8_t i2c_addr, uint8_t ColumnIndex, uint8_t R, uint8_t G, uint8_t B)	
	描述	点亮一列 RGB 灯
	参数	i2c_addr：I²C 通信地址 ColumnIndex：列序号 (1) 0~7 R/G/B：颜色数值
	返回值	void
	备注	—
21	void writeRow(uint8_t i2c_addr, uint8_t RowIndex, uint8_t R, uint8_t G, uint8_t B)	
	描述	点亮一行 RGB 灯
	参数	i2c_addr：I²C 通信地址 RowIndex：行序号 (2) 0~7 R/G/B：颜色数值
	返回值	void
	备注	—
21	void DrawAsciiChar(uint8_t i2c_addr, char Ch, uint8_t R, uint8_t G, uint8_t B)	
	描述	显示一个 ASCII 字符
	参数	i2c_addr：I²C 通信地址 Ch：ASCII 数值或字符 R/G/B：颜色数值
	返回值	void
	备注	字符的 ASCII 码值，范围 32~126。如 97 对应字符‘a’。

22	void setGradient(uint8_t i2c_addr,uint8_t mode,uint8_t T1,uint8_t T2=T2_IS_T1,uint8_t T3=T3_IS_T1,uint8_t T4=T4_IS_T1)	
	描述	设置渐变功能
	参数	i2c_addr：I²C 通信地址 mode：渐变的工作模式 (4) 0(GFS_GRADIENT_GARMA)：伽马强度渐变的渐变模式 1(GFS_GRADIENT_LINEAR)：线性强度渐变的渐变模式 2(GFS_BLINK)：闪烁模式 T1：渐变 T1 时间 (4) (注：本模块每帧的时间为 1.813ms) 0(GFT_OFF)：关闭 1(GFT_256_FRAME)：256 帧 2(GFT_512_FRAME)：512 帧 3(GFT_1024_FRAME)：1024 帧 4(GFT_1536_FRAME)：1536 帧 5(GFT_2048_FRAME)：2048 帧 6(GFT_2560_FRAME)：2560 帧 7(GFT_256_FRAME)：3072 帧 T3：渐变 T3 时间 (4) 0(T3_IS_T1)：T1 (默认) 1(T3_IS_T1x2)：T1×2 T2：渐变 T2 时间 (4) 0(T2_IS_T1x1_4)：T1×0.25 1(T2_IS_T1x1_2)：T1×0.5 2(T2_IS_T1)：T1 (默认) 3(T2_IS_T1x2)：T1×2 T4：渐变 T4 时间 (4) 0(T4_IS_T1x1_4)：T1×0.25 1(T4_IS_T1x1_2)：T1×0.5 2(T4_IS_T1)：T1 (默认) 3(T4_IS_T1x2)：T1×2
	返回值	void
	备注	详细内容请见 HT16D33A 规格书

注：RGB 序号如下：



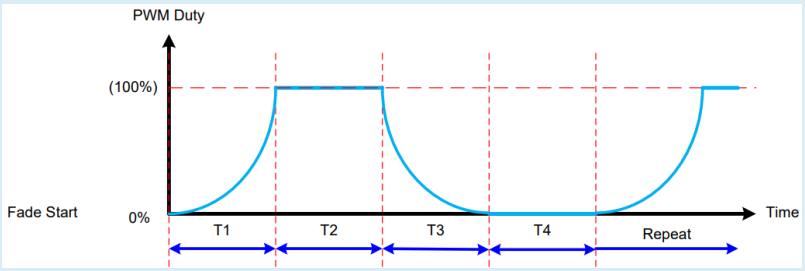
注 1：从左到右灯珠列号增加。列号范围 0~7

注 2：从上到下灯珠行号增加。行号范围 0~7

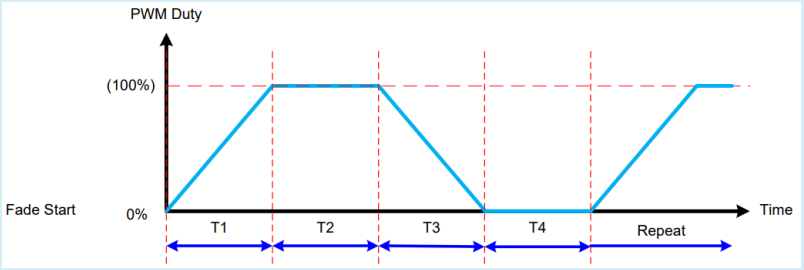
注 3：RGB 序号 = 行号 × 8 + 列号

注 4：渐变模式与时间说明

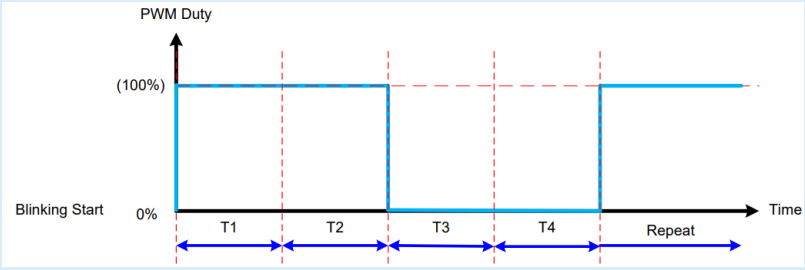
伽马强度渐变的渐变模式



线性强度渐变的渐变模式



闪烁模式

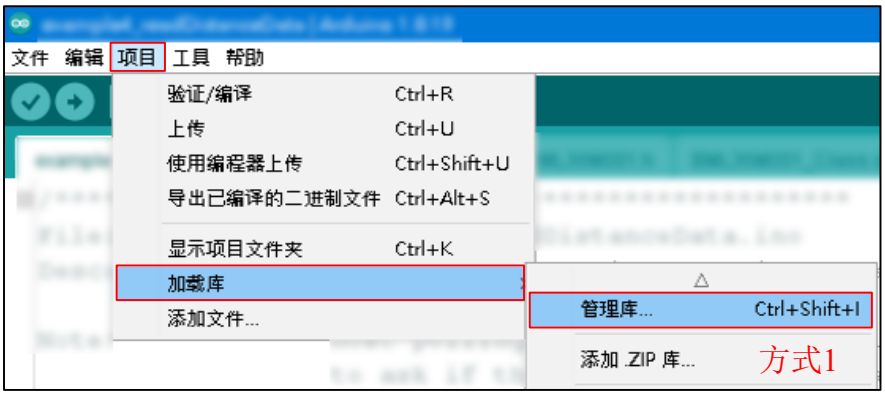


Arduino Lib 下载及安装

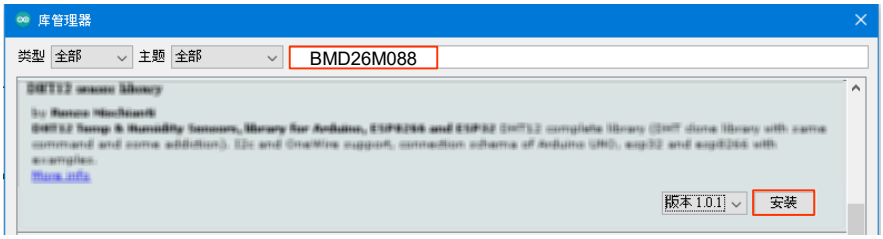
BMD26M088 Library：可参考下面两种方法安装 BMD26M088 的 Arduino Library

方式 1：搜索安装

搜索安装：Arduino IDE → 项目 → 加载库 → 管理库 ... → 搜索 BMD26M088 → 安装



搜索安装流程 **1**

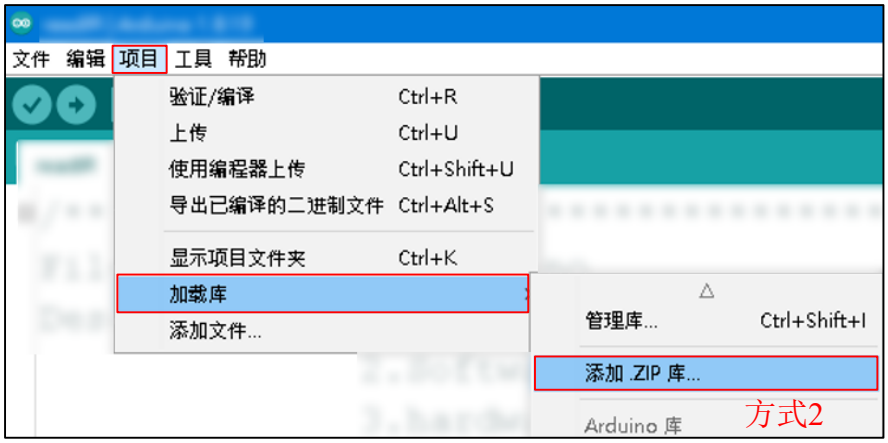


搜索安装流程 **2**

方式二：添加 .ZIP 库，需提前下载 .ZIP 库

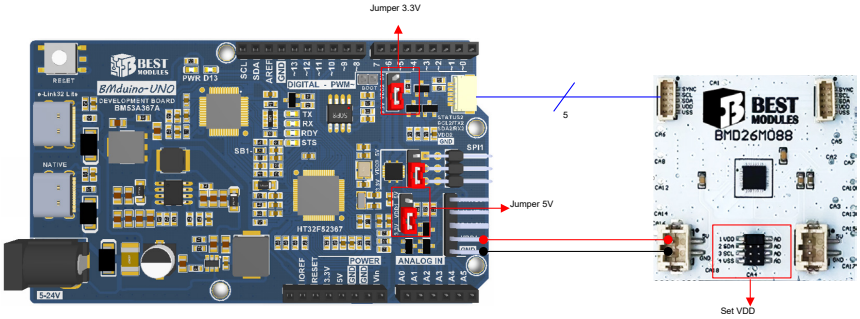
下载方法：打开倍创官方网站 (https://www.bestmodulescorp.com/bmd26m088.html) 文件目录下的 Arduino 范例程序 (BMD26M088 Library)。

添加 .ZIP 库：Arduino IDE → 项目 → 加载库 → 添加 .ZIP 库 ...



Arduino 范例

范例 1: blink



实物连接示意图

范例实现功能：所有 RGB 闪烁

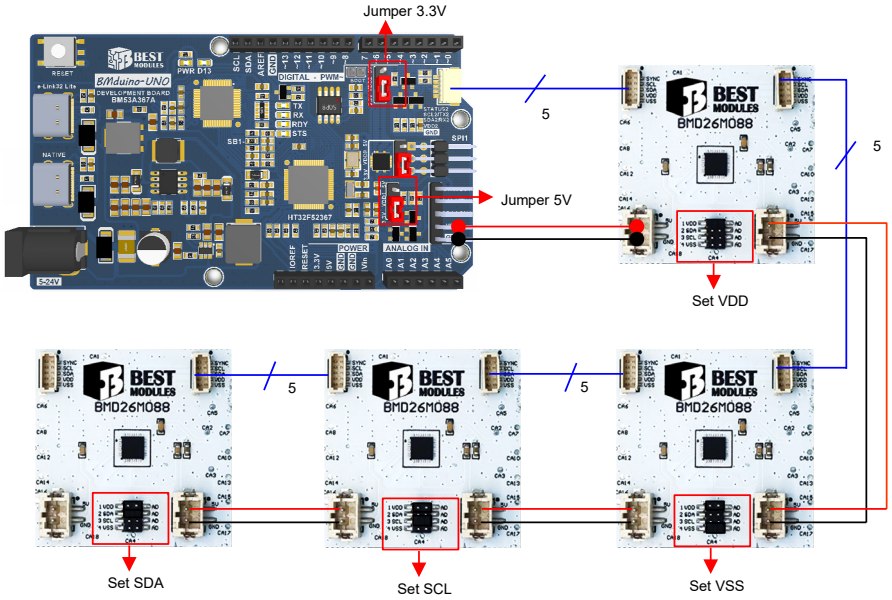
1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMD26M088) → 选择范例 (blink)
2. 范例说明：
   a. 创建对象 & 模块初始化及配置

```
#include <BMD26M088.h>
// 设定 I²C 地址，范例采用 I²C 地址跳帽处于 VDD on 位置
BMD26M088 myBMD26M088(&Wire2);
#define ADDRESS BMD26M088_I2C_ADDRESS_VDD
void setup()
{
  myBMD26M088.begin(ADDRESS);                  // 模块初始化
  myBMD26M088.setBrightness(ADDRESS,0xff); // 设置全局亮度为最大
  myBMD26M088.setCurrent(ADDRESS,BMD26M088_CCR_6MA); // 设置恒流率为 6mA
}
```

   b. 使所有 RGB 闪烁

```
void loop()
{
  // 点亮所有 RGB 为红色
  myBMD26M088.writeAllRGB(ADDRESS,0xff,0,0);
  delay(200);                          // 延时 200ms
  myBMD26M088. clearAll (ADDRESS); // 熄灭所有 RGB 灯珠
  delay(200);                          // 延时 200ms
}
```

范例 2: cascade4



实物连接示意图

范例实现功能：单路 I²C 控制级联的 4 个模块，显示不同颜色

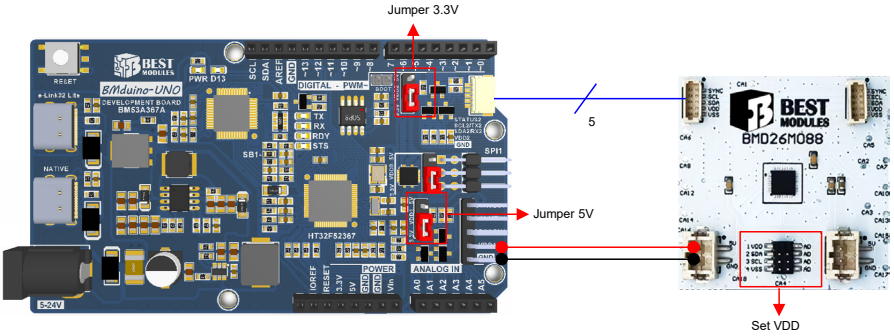
注意：由于 BMduino 供电能力有限，本次设计在程序中设置恒流率为 6mA。

1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMD26M088) → 选择范例 (cascade4)
2. 范例说明：
   a. 创建对象 & 模块初始化 & 控制级联的 4 个模块显示不同颜色

```
#include “BMD26M088.h”
#include “BMB22T124.h”
BMD26M088 myBMD26M088(&Wire2);
#define i2c_addr_boardcast BMD26M088_I2C_ADDRESS_BOARDCAST// 广播地址
// 利用跳帽设置 4 个模块分别为不同的 I²C 地址
#define i2c_addr_board1 BMD26M088_I2C_ADDRESS_VDD// 地址 1
#define i2c_addr_board2 BMD26M088_I2C_ADDRESS_GND// 地址 2
#define i2c_addr_board3 BMD26M088_I2C_ADDRESS_SCL// 地址 3
#define i2c_addr_board4 BMD26M088_I2C_ADDRESS_SDA// 地址 4
void setup()
{
  myBMD26M088.begin(i2c_addr_boardcast);          // 模块初始化
  myBMD26M088.setBrightness((i2c_addr_boardcast,0xff); // 设置全局亮度
                                                   // 为最大
  myBMD26M088.setCurrent((i2c_addr_boardcast,BMD26M088_CCR_6MA);
                                                   // 设置恒流率为 6mA
// 根据第一个模块的 I²C 地址，设定第一个模块为红色
  myBMD26M088.writeAllRGB(i2c_addr_board1, 0xff,0,0);
// 根据第二个模块的 I²C 地址，设定第二个模块为绿色
  myBMD26M088.writeAllRGB(i2c_addr_board2, 0, 0xff,0);
```

```
// 根据第三个模块的 I²C 地址，设定第三个模块为蓝色
  myBMD26M088.writeAllRGB(i2c_addr_board3, 0,0, 0xff);
// 根据第四个模块的 I²C 地址，设定第四个模块为黄色
  myBMD26M088.writeAllRGB(i2c_addr_board4, 0xff, 0xff,0);
}
```

范例 3: gradient



实物连接示意图

范例实现功能：亮度渐变控制

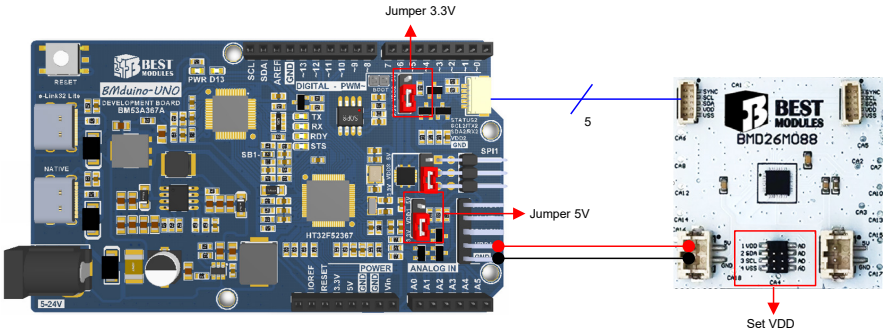
注意：由于 BMduino 供电能力有限，本次设计在程序中设置恒流率为 6mA。

1. 范例打开方式：Arduino IDE→ 文件 → 示例 →Lib 选择 (BMD26M088)→ 选择范例 (gradient)
2. 示例说明：

a. 创建对象 & 模块初始化 & 亮度渐变控制

```
#include <BMD26M088.h>
BMD26M088 myBMD26M088(&Wire2);
#define ADDRESS BMD26M088_I2C_ADDRESS_VDD
void setup()
{
  myBMD26M088.begin(ADDRESS);  //I2C 初始化
  myBMD26M088.setBrightness(ADDRESS,0xff);// 设置亮度
  myBMD26M088.setCurrent(ADDRESS,BMD26M088_CCR_6MA);// 设置恒流率
// 亮度渐变 RGB 的位置和颜色设置
  //myBMD26M088.writeAllRGB(ADDRESS, 0xff, 0xff, 0xff);// 设置所有 RGB
myBMD26M088.writeRGB(ADDRESS, 1,0xff, 0, 0);// 设置单个 RGB
 // 设置渐变参数
  myBMD26M088.setGradientControl(ADDRESS, FFEN_ENABLE, GMEN_GARMA,
GFEN_GLOBE, FOT_T1, FET_T1x1_4, FET_T1x1_2);
  // 全局渐变配置
 myBMD26M088.setGlobeGradient(ADDRESS,GFS_GRADIENT,GFT_1024_FRAME);
  }
void loop()
{
 }
```

范例 4: colorful



实物连接示意图

范例实现功能：七彩流光显示

注意：由于 BMduino 供电能力有限，本次设计在程序中设置恒流率为 6mA。

1. 范例打开方式：Arduino IDE→ 文件 → 示例 →Lib 选择 (BMD26M088)→ 选择范例 (colorful)
2. 示例说明：

a. 创建对象 & 模块初始化 & 流光控制

```
#include <BMD26M088.h>
BMD26M088 myBMD26M088(&Wire2);
#define ADDRESS BMD26M088_I2C_ADDRESS_VDD // 模块位址

typedef struct RGB_DATA
{
  uint8_t r;
  uint8_t g;
  uint8_t b;
}rgb_def;
rgb_def get_color(int param);// 颜色获取函数
void setup()
{
    myBMD26M088.begin(ADDRESS);  //I2C 初始化
    myBMD26M088.setBrightness(ADDRESS,0xff);// 设置全局亮度
    myBMD26M088.setCurrent(ADDRESS,BMD26M088_CCR_6MA);// 设置恒流 6mA
 }
int flow = 0;
void loop()
{
    u8 i;
    rgb_def rgb;
    for(i = 0; i < 8; i++)// 更新 8 行 RGB
    {
      rgb = get_color(flow + i * (420 / 16));// 获取颜色
      myBMD26M088.writeRow(ADDRESS, i, rgb.r, rgb.g, rgb.b);// 更新一
                                                            // 行 RGB
    }
     if((flow++) > 420) {flow = 0;}// 每个循环刷新 420 次颜色
}
```

```
// 颜色获取函数，根据输入的参数获取对应的颜色
rgb_def get_color(int param) {
  rgb_def rgb;
  if(param<0){param+=420;}
  else  if(param>419){param-=420;}
  if(param<60)  //step 1:red->yellow
  {
    rgb.r=240;
    rgb.g=4 * param;
    rgb.b=0;
  }
  else if(param<120)//step 2:yellow->green
 {
    rgb.r=240-(4*(param-60));
    rgb.g=240;
    rgb.b=0;
  }
  else if(param<180)//step 3:green->cyan
  {
    rgb.r=0;
    rgb.g=240;
    rgb.b=4*(param-120);
  }
  else if(param<240)//step 4:cyan close to blue
  {
    rgb.r=0;
    rgb.g=240-(3*(param-180));
    rgb.b=240;
  }
  else if(param<300)//step 5:blue->purple
  {
    rgb.r=(param-240);
    rgb.g=60-(param-240);
    rgb.b=240-(2*(param-240));
  }
  else if(param<360)//step6:purple->pink
  {
    rgb.r=60+(param-300);
    rgb.g=0;
    rgb.b=120;
  }
  else if(param<420)//step7:pink->red
  {
    rgb.r=120+(2*(param-360));
    rgb.g=0;
    rgb.b=120-(2*(param-360));
  }
  return rgb;
}
```