



小功率雾化模块

BM52D5121-1

Arduino Library 说明

版本: V1.00 日期: 2023-05-30

www.bestmodulescorp.com

目录

简介	3
Arduino Lib 函数	3
Arduino Lib 下载及安装	7
Arduino 范例	8
范例 1: setAtoAndFan	8
范例 2: setRGB	10
范例 3: calibrateWaterLevel	11

简介

BM52D5121-1 是倍创推出的小功率雾化模块，使用 UART 通信方式。本文档对 BM52D5121-1 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了设置风扇和雾化功率以及控制外接 RGB 灯等功能。

Arduino Lib 函数

Arduino Lib 名称: BM52D5121-1		Lib 版本: V1.0.1
构造函数 & 初始化		
1	BM52D5121_1(HardwareSerial *theSerial=&Serial)	
	描述	构造函数，使用 HW Serial 接口
	参数	*theSerial: 选择 HW Serial 接口 (默认 Serial 接口)
	返回值	—
	备注	—
2	BM52D5121_1(uint8_t rxPin, uint8_t txPin)	
	描述	构造函数，使用 SW Serial 接口
	参数	rxPin: TX 连接的引脚 txPin: RX 连接的引脚
	返回值	—
	备注	—
3	void begin(int32_t baud=UART_BAUD)	
	描述	模块初始化
	参数	baud: 模块波特率 9600(UART_BAUD): 固定 9600bps
	返回值	void
	备注	—
功能函数		
4	uint8_t readWaterValue()	
	描述	读取检水感度值
	参数	—
	返回值	检水感度值
	备注	范围 0x00~0x80，检水感度值越大，代表水位越高
5	uint8_t readWaterThreshold()	
	描述	读取缺水阈值
	参数	—
	返回值	缺水阈值
	备注	检水感度值低于这个值则模块处于缺水状态，检水校准后产生缺水阈值

6	uint8_t getAtoPower()	
	描述	获取雾化功率
	参数	—
	返回值	power: 雾化功率参数, 范围 0x00~0x20 雾化功率 = 5W + 雾化功率参数 × 5/32W 0xff: 关闭雾化
	备注	—
7	uint8_t getFanPower()	
	描述	获取风扇功率
	参数	—
	返回值	风扇功率: bit1=0, bit0=0: 风扇全功率 bit1=0, bit0=1: 风扇半功率 bit1=1: 风扇关闭
	备注	—
8	uint8_t getRGB(uint8_t buff [])	
	描述	获取 RGB 数据
	参数	buff []: 储存 RGB 数据, 4 bytes buff [0]: 控制呼吸灯周期时间和 RGB 模式选择 bit5~bit4: 呼吸周期时间 00: 4.7s 01: 6.4s 10: 8.3s 11: 9.3s bit1~bit0: 00: 关灯 01: 流光灯模式 10: 呼吸灯模式 11: 正常灯模式 (某个灯光常亮) buff [1]~buff [3]: 呼吸灯以及正常灯模式: buff [1]~buff [3]: 引脚 R、G、B 的 PWM 输出调节, 范围都为 0x00~0x80。 流光灯模式: buff [1]~buff [2]: 流光灯变化速度参数, 建议范围 0x0064~0x012c, 流光灯变化速度 = 流光灯变化速度参数 × 60, 单位 ms buff [3]: 任意值, 无作用。
	返回值	执行情况: 0: 成功 1: 失败
	备注	—
9	uint16_t getTimer()	
	描述	获取定时剩余时间
	参数	—
	返回值	定时剩余时间, 单位 s
	备注	—

10	uint16_t getFWver()	
	描述	获取版本号
	参数	—
	返回值	版本号
	备注	—
11	uint8_t getModuleStatus()	
	描述	获取模块状态
	参数	—
	返回值	模块状态： bit0: 检水状态位 0: 检测无水 1: 检测有水 bit1: 检水校准状态位 0: 校准成功 (或上电后未校准过) 1: 校准失败 bit2: 过流状态位 0: 雾化正常 (或上电后未雾化过) 1: 雾化过流 bit3: 追频状态位 0: 追频成功 (或上电后未追频过) 1: 追频失败 bit4: 休眠状态位 0: 模块运行中 1: 模块休眠
	备注	—
配置函数		
12	uint8_t calibrateWaterLevel(uint8_t cmd)	
	描述	设置模块检水校准命令
	参数	cmd: 检水校准命令 0x01: 空杯校准 0x02: 满杯校准
	返回值	执行情况： 0: 成功 1: 失败
	备注	—
13	uint8_t setAtoPower(uint8_t power)	
	描述	设置雾化功率
	参数	power: 雾化功率参数, 范围 0x00~0x20 雾化功率 = 5W + 雾化功率参数 × 5/32W 0xff: 关闭雾化
	返回值	执行情况： 0: 成功 1: 失败
	备注	—

14	uint8_t setFanPower(uint8_t power)	
	描述	设置风扇功率
	参数	power: 设置风扇功率 bit1=0, bit0=0: 全功率 bit1=0, bit0=1: 半功率 bit1=1: 关闭风扇
	返回值	执行情况: 0: 成功 1: 失败
	备注	—
15	uint8_t setScanFrequency()	
	描述	设置模块重新追频
	参数	—
	返回值	执行情况: 0: 成功 1: 失败
	备注	上电追频失败、切换雾化片不重新上电可使用函数重新追频, 追频要等待 3.2s
16	uint8_t setRGB(uint8_t buff[])	
	描述	设置 RGB 数据 (4 bytes)
	参数	buff []: 储存 RGB 数据, 4 bytes buff [0]: 控制呼吸灯周期时间和 RGB 模式选择 bit5~bit4: 呼吸周期时间 00: 4.7s 01: 6.4s 10: 8.3s 11: 9.3s bit1~bit0: 00: 关灯 01: 流光灯模式 10: 呼吸灯模式 11: 正常灯模式 (某个灯光常亮) buff [1]~ buff [3]: 呼吸灯以及正常灯模式: buff [1]~buff [3]: 引脚 R、G、B 的 PWM 输出调节, 范围都为 0x00~0x80。 流光灯模式: buff [1]~buff [2]: 流光灯变化速度参数, 建议范围 0x0064~0x012c, 流光灯变化速度 = 流光灯变化速度参数 × 60, 单位 ms buff [3]: 任意值, 无作用。
	返回值	执行情况: 0: 成功 1: 失败
	备注	—

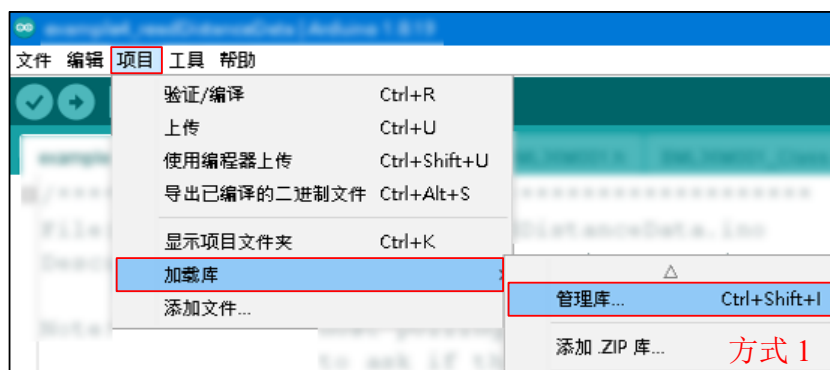
17	uint8_t setTimer(uint16_t time)	
	描述	设置定时时间
	参数	time: 定时时间, 单位 s
	返回值	执行情况: 0: 成功 1: 失败
	备注	—
18	uint8_t sleep()	
	描述	设置模块休眠
	参数	—
	返回值	执行情况: 0: 成功 1: 失败
	备注	—
19	uint8_t reset()	
	描述	设置模块复位
	参数	—
	返回值	执行情况: 0: 成功 1: 失败
	备注	—

Arduino Lib 下载及安装

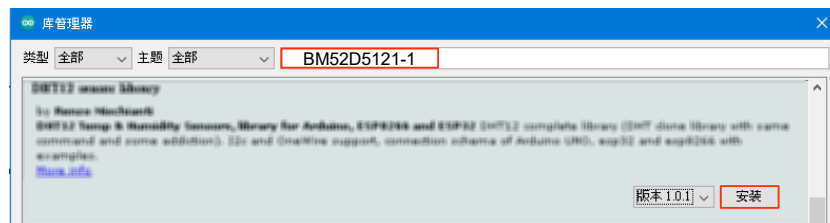
BM52D5121-1 Library: 可参考下面两种方法安装 BM52D5121-1 的 Arduino Library

方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BM52D5121-1 → 安装



搜索安装流程 1

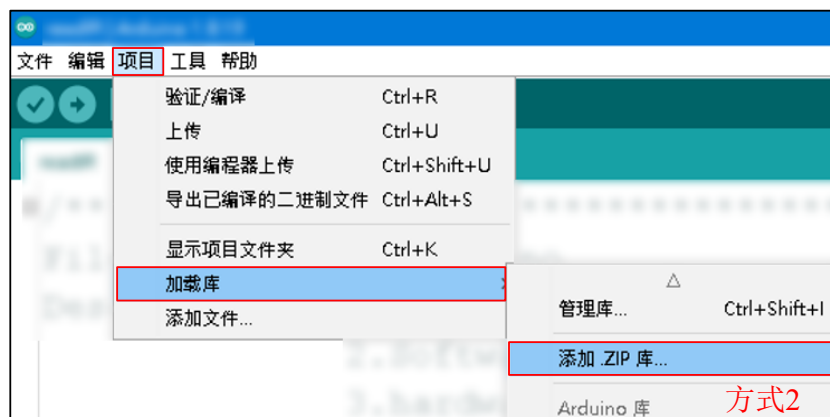


搜索安装流程 2

方式 2：添加 .ZIP 库，需提前下载 .ZIP 库

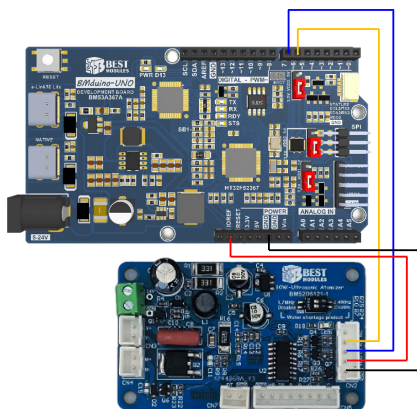
下载方法：打开倍创官方网站 (<https://www.bestmodulescorp.com/bm52d5121-1.html#tab-product2>) 文件目录下的 Arduino 范例程序 (BM52D5121-1 Library)。

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



Arduino 范例

范例 1：setAtoAndFan



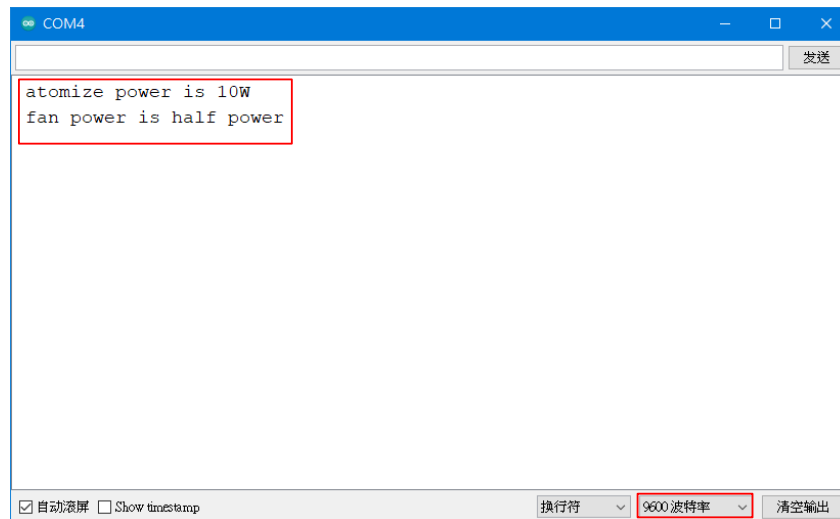
范例实现功能：配置雾化功率为10W，待追频(3.2s)完成后，配置风扇为半功率。

注意：上电前，将模块的二位拨码开关拨至检水 disable

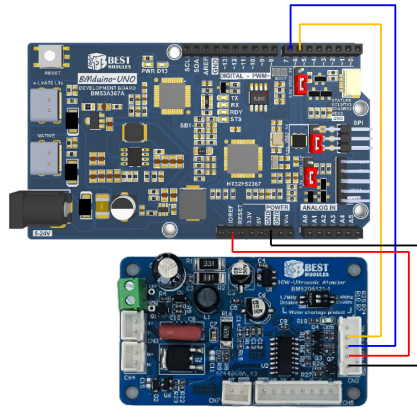
1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM52D5121-1) → 选择范例 (setAtoAndFan)
2. 示例说明：
 - a. 构建对象 & 模块初始化及配置

```
#include <BM52D5121-1.h>
BM52D5121_1 Atomize(7,6);           // 使用软件 UART
void setup() {
    Atomize.begin(9600);
    Serial.begin(9600);
    If(Atomize.setAtoPower(0x20)==0) // 设置雾化功率
    {
        Serial.println("atomize power is 10W");
    }
    delay(3200);
    if(Atomize.setFanPower(0x01)==0) // 设置风扇功率
    {
        Serial.println("fan power is half power");
    }
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示如下



范例 2: setRGB



范例实现功能：配置雾化功率为 7.5W，配置风扇功率为半功率，三种 RGB 模式循环切换，正常灯 → 呼吸灯 → 流光灯，模式切换间隔 30s

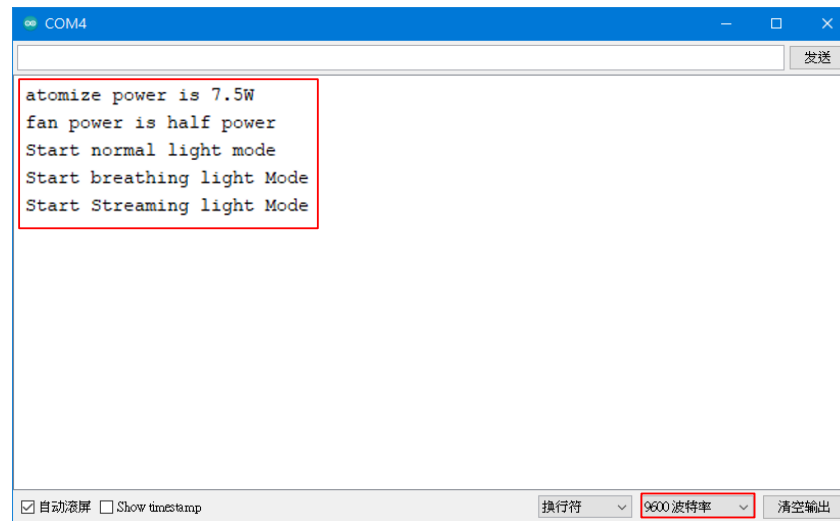
1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM52D5121-1) → 选择范例 (setRGB)
2. 示例说明：
 - a. 构建 & 模块初始化及配置

```
#include <BM52D5121-1.h>
BM52D5121_1 Atomize(7,6);                                     // 使用软件 UART
uint8_t normal[4]={0x03,0x00,0x80,0x80};                     // 正常灯
uint8_t breathe[4]={0x32,0x80,0x40,0x00};                     // 呼吸灯
uint8_t streamer[4]={0x01,0x00,0x74,0x00};                     // 流光灯
void setup() {
    Atomize.begin(9600);
    Serial.begin(9600);
    if(Atomize.setAtoPower(0x11)==0){
        Serial.println("atomize power is 7.5W"); // 配置雾化功率为 7.5W
    }
    delay(3200);
    if(Atomize.setFanPower(0x01)==0){
        Serial.println("fan power is half power"); // 配置风扇功率为半功率
    }
}
```

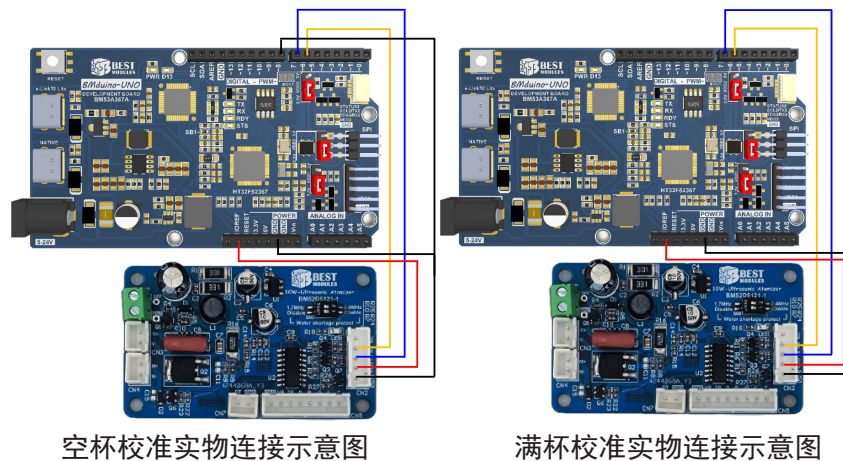
- b. 三种 RGB 模式循环切换，正常灯 → 呼吸灯 → 流光灯，模式切换间隔 30s

```
void loop() {
    if(Atomize.setRGB(normal)==0){
        Serial.println("Start normal light mode"); // 正常灯
    }
    delay(30000);
    if(Atomize.setRGB(breathe)==0){
        Serial.println("Start breathing light Mode"); // 呼吸灯
    }
    delay(30000);
    if(Atomize.setRGB(streamer)==0){
        Serial.println("Start Streaming light Mode"); // 流光灯
    }
    delay(30000);
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示如下



范例 3: calibrateWaterLevel



范例实现功能：先进行空杯校准，等待 LED 闪烁即空杯校准完成，加水至满杯再进行满杯校准。

注意：上电前，模块两位拨码开关拨至 Enable 处

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM52D5121-1) → 选择范例 (calibrateWaterLevel)

2. 示例说明:

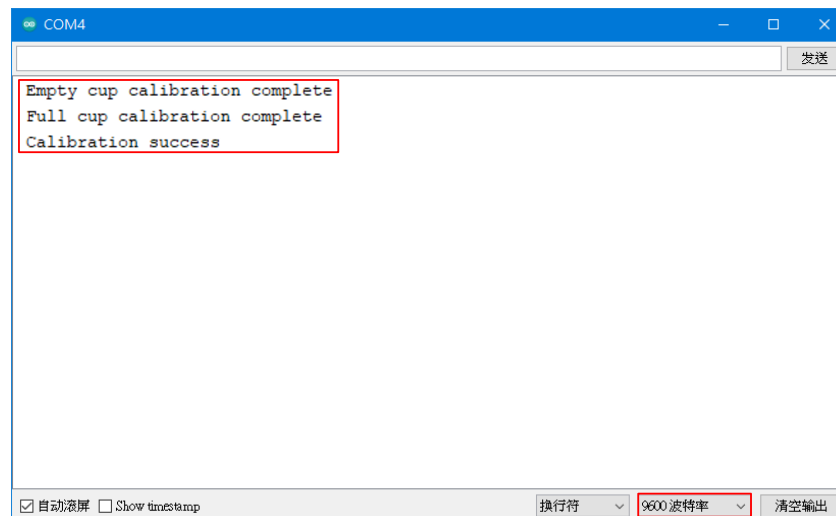
a. 构建 & 初始化模块

```
#include <BM52D5121-1.h>
BM52D5121_1 Atomize(7,6);
uint8_t i=2;
uint8_t flag=0;
void setup() {
  Atomize.begin(9600);
  Serial.begin(9600);
  pinMode(8,INPUT);
}
```

b. 轮询判断 PIN8 接收到的高低电平，进行对应的检水校准动作

```
void loop() {
  i=pinRead(8);
  if((i==0)&&(flag!=1)){ // 连接 PIN8 和 GND
    if(Atomize.calibrateWaterLevel(0x01)==0) { // 空杯校准
      Serial.println("Empty cup calibration complete");
      flag=1;
    }
  }
  if((i==1)&&(flag==1)){ // 断开 PIN8 和 GND
    Atomize.calibrateWaterLevel(0x02); // 满杯校准
    flag=2;
    Serial.println("Full cup calibration complete");
    delay(1000);
    i=Atomize.getModuleStatus(); // 获取模块状态
    uint8_t a=bitRead(i,1); // 判断检水校准标志位是否失败
    if(a==1){
      Serial.println("Calibration failed");
    }
    else{
      Serial.println("Calibration success");
    }
  }
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示如下



Copyright© 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版时倍创已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。倍创不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。倍创就文中提到的信息及该信息之应用，不承担任何法律责任。此外，倍创并不推荐将倍创的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。倍创特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用倍创产品的风险完全由买方承担，如因该等使用导致倍创遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使倍创免受损害。倍创 (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。倍创在此并未明示或暗示授予任何知识产权。倍创拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。