



中功率雾化模块

BMN31K522

Arduino Library 说明

版本: V1.00 日期: 2023-05-31

www.bestmodulescorp.com

目录

简介	3
Arduino Lib 函数	3
Arduino Lib 下载及安装	4
Arduino 范例	5
范例 : controlAtomization	5

简介

BMN31K522 是倍创推出的中功率雾化模块，使用 UART 通信方式。本文档对 BMN31K522 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示雾化功能和雾化档位切换功能。

Arduino Lib 函数

Arduino Lib 名称: BMN31K522		Lib 版本: V1.0.1
构造函数 & 初始化		
1	BMN31K522(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	BMN31K522(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()	
	描述	模块初始化
	参数	—
	返回值	void
	备注	波特率固定为 9600
功能函数		
4	uint8_t setAtomizationRank(uint8_t rank)	
	描述	设置雾化档位
	参数	rank: 雾化档位, 范围 0~255 (PWM 脉宽)
	返回值	执行情况: 0x00: 成功 0xFF: 失败
	备注	—
5	uint8_t setAtomizationTime(uint8_t times)	
	描述	设置雾化定时时间
	参数	times: 定时时间, 范围 0~255, 单位: min
	返回值	执行情况: 0x00: 成功 0xFF: 失败
	备注	—

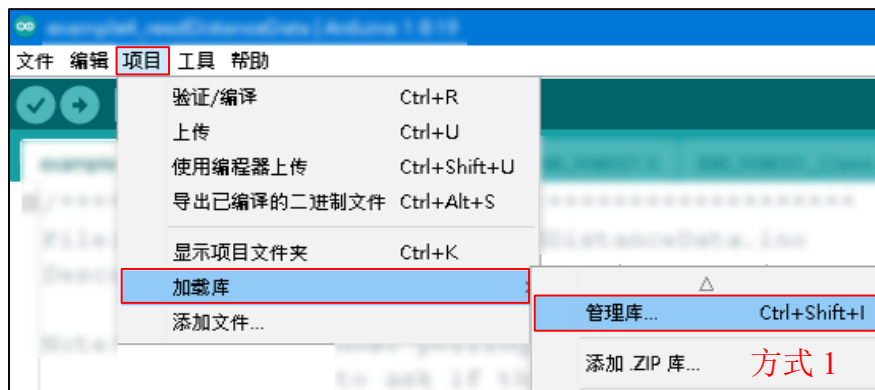
6	uint8_t getAtomizationRank()	
	描述	获取雾化档位值
	参数	—
	返回值	雾化档位 0~255
7	uint8_t getAtomizationTime()	
	描述	获取雾化定时剩余时间
	参数	—
	返回值	定时剩余时间 (单位: min)
8	uint8_t getWaterStatus()	
	描述	获取水位状态
	参数	—
	返回值	水位状态: 0: 缺水 1: 有水
	备注	—

Arduino Lib 下载及安装

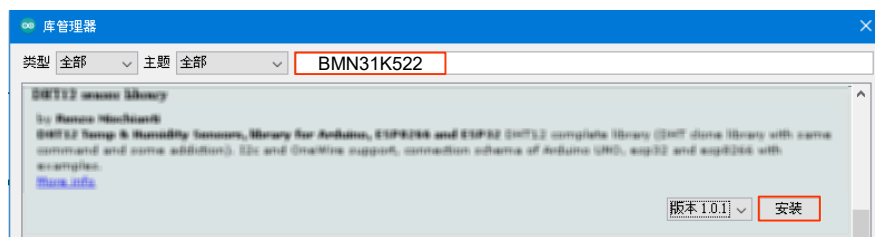
BMN31K522 Library: 可参考下面两种方法安装 BMN31K522 的 Arduino Library

方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BMN31K522 → 安装



搜索安装流程 1

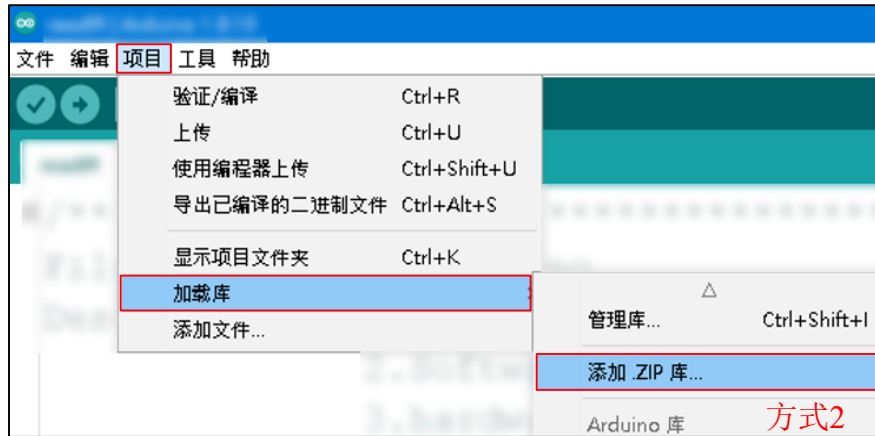


搜索安装流程 2

方式 2：添加 .ZIP 库，需提前下载 .ZIP 库

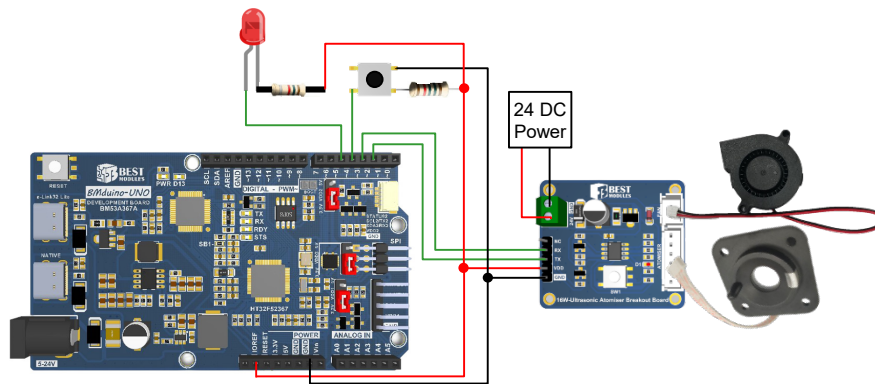
下载方法：打开倍创官方网站 (<https://www.bestmodulescorp.com/bmn31k522.html#tab-product2>) 文件目录下的 Arduino 范例程序 (BMN31K522 Library)。

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



Arduino 范例

范例 : controlAtomization



实物连接示意图

范例功能描述：检水有水的条件下，按键作为开关，控制雾化模块定时工作（雾化档位、定时时间固定）。LED 亮表示雾化开启；LED 灭表示雾化停止。

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BMN31K522) → 选择范例 (controlAtomization)

2. 示例说明:

a. 构建对象 & 初始化模块

```

#include <BMN31K522.h>
#define SERIAL_DEBUG (1) // 0 - 禁用串行调试输出; 1 - 启用串行调试输出
#define ATOMIZER_RANK (255)
#define ATOMIZER_TIME (1)
BMN31K522 myAtomizer(2, 3); // 创建对象
byte atomizerLED = 5; // 雾化器工作 LED
byte atomizerSwitch = 4; // 雾化器开关 PIN
unsigned long TimeCount_1s = 0; // 1 秒的时间计数
byte waterStatus = 0; // 缺水状态: 0 - 缺水; 1 - 有水
unsigned long switchOnTime = 0; // 开关时间计数
bool atomizerOnFlag = false; // 雾化器工作标志
void setup()
{
    if (SERIAL_DEBUG){
        Serial.begin(9600);
    } myAtomizer.begin(); // 模块初始化
    pinMode(atomizerLED,OUTPUT);
    pinMode(atomizerSwitch,INPUT);
    digitalWrite(atomizerLED,HIGH);
    TimeCount_1s = millis(); // 开始计时
}
    
```

- b. 间隔 1s 获取一次检水数据，并在串口监视器上显示检水状态；扫描按键是否按下，判断是否开启 / 关闭雾化工作；当雾化开启时，间隔 1s 获取雾化定时剩余时间，当时间为 0 时，定时结束雾化停止；LED 显示雾化工作状态，雾化开启 - 亮，雾化停止 - 灭。

```

void loop() {
    byte switchStatus;
    uint8_t getTime;
    /***** 间隔 1 秒获取检水状态 & 剩余时间 *****/
    if (millis() - TimeCount_1s >= 1000)
    {
        TimeCount_1s = millis();
        /***** 获取检水状态 *****/
        waterStatus=myAtomizer.getWaterStatus();
        if (SERIAL_DEBUG)
        {
            if (waterStatus)
            {
                Serial.println("Atomizer' s water status is: Have water");
                // 串口监视器显示检水状态
            }
            else
            {
                Serial.println("Atomizer' s water status is: No water");
                // 串口监视器显示检水状态
            }
        }
        /***** 获取剩余时间 *****/
        if (atomizerOnFlag)
        {
            getTime=myAtomizer.getAtomizationTime();
            if (getTime == 0)
                
```

```
    {
        atomizerOnFlag = false;
        digitalWrite(atomizerLED,HIGH);    // LED 指示雾化工作状态
    }
    if (SERIAL_DEBUG)
    {
        Serial.print("Atomizer work remaining time is:");
        // 串口监视器显示雾化定时剩余时间
        Serial.println(getTime);
    }
}
}
/***** 按键扫描 *****/
if (waterStatus == 1)
{
    switchStatus = digitalRead(atomizerSwitch);
    if (!switchStatus)
    {
        if ((millis() - switchOnTime) > 150) // 开关去抖
        {
            switchOnTime = millis();
            if (!atomizerOnFlag)
            {
                atomizerOnFlag = true;
                digitalWrite(atomizerLED,LOW);
                myAtomizer.setAtomizationRank(ATOMIZER_RANK); // 设定雾化档位
                myAtomizer.setAtomizationTime(ATOMIZER_TIME); // 设定雾化时间
            }
            else
            {
                atomizerOnFlag = false;
                digitalWrite(atomizerLED,HIGH); // LED 指示雾化工作状态
                myAtomizer.setAtomizationRank(0); // 关闭雾化工作
                myAtomizer.setAtomizationTime(0);
            }
        }
    }
    else
    {
        switchOnTime = millis();
    }
}
/***** 缺水处理 *****/
else
{
    if (atomizerOnFlag)
    {
        atomizerOnFlag = false;
        digitalWrite(atomizerLED,HIGH);
    }
}
}
```

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。倍创不承担任何明示、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。