



酒精数字传感器

BM22S3421-1 Arduino Library 说明

版本: V1.11 日期: 2024-08-23

www.bestmodulescorp.com

目录

简介	3
Arduino Lib 函数	3
Arduino Lib 下载及安装	6
Arduino 范例	8
范例: readAlcLevelAndADValue	8

简介

BM22S3421-1 是倍创推出的一款酒精数字传感器，使用 UART 通信方式。本文档对 BM22S3421-1 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明。范例使用了 BMA34M421 模块，演示了读取酒精气体浓度信息的功能。

适用型号：

型号	说明
BM22S3421-1	酒精数字传感器
BMA34M421	板载 BM22S3421-1 传感器

Arduino Lib 函数

Arduino Lib 名称：BM22S3421-1		Lib 版本：V1.0.1
构造函数 & 初始化		
1	BM22S3421_1(uint8_t statusPin, HardwareSerial *theSerial=&Serial)	
	描述	构造函数，使用硬件 UART 通信
	参数	statusPin: STATUS 引脚，连接 BM22S3421-1 的 STATUS 引脚或 BMA34M421 的 STA 引脚 *theSerial: 选择硬件 UART 接口 (默认 Serial 接口)
	返回值	—
	备注	—
2	BM22S3421_1(uint8_t statusPin, uint8_t rxPin, uint8_t txPin)	
	描述	构造函数，使用软件 UART 通信
	参数	statusPin: STATUS 引脚，连接 BM22S3421-1 的 STATUS 引脚或 BMA34M421 的 STA 引脚 rxPin: RX 引脚，连接 BM22S3421-1 或 BMA34M421 的 TX 引脚 txPin: TX 引脚，连接 BM22S3421-1 或 BMA34M421 的 RX 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	—
参数查询		
4	uint8_t getSTATUS()	
	描述	获取 STATUS 引脚电平
	参数	—
	返回值	STATUS 引脚电平： 0: 低电平 1: 高电平
	备注	高 / 低电平指示模块是否报警，报警时的电平可设 (默认高电平)

5	uint16_t readADValue()	
	描述	读取酒精气体的实时 A/D 值
	参数	—
	返回值	12-bit A/D 值
	备注	A/D 值越大，代表酒精气体浓度越大 一般用于主动获取数据，建议在除能串口自动输出功能后使用 ⁽²⁾
6	uint8_t readAlcLevel()	
	描述	读取酒精气体浓度等级
	参数	—
	返回值	酒精气体浓度等级： 1：等级 1 (低浓度) 2：等级 2 3：等级 3 4：等级 4 (高浓度)
	备注	一般用于主动获取数据，建议在除能串口自动输出功能后使用 ⁽²⁾
7	uint8_t readParam(uint8_t cmd, uint8_t addr)	
	描述	读取模块参数
	参数	cmd: 指令码 addr: 地址码
	返回值	模块参数
	备注	查阅 BM22S3421-1 数据手册的指令表 (特殊查询指令)
8	bool isInfoAvailable()	
	描述	查询是否接收到模块自动输出的信息 (14 字节)
	参数	—
	返回值	接收情况： true: 已收到 false: 未收到
	备注	需在使能串口自动输出功能后使用 ⁽²⁾ ，模块出厂默认使能。
9	void readInfoPackage(uint8_t array[])	
	描述	读取模块自动输出的信息
	参数	array[0]~array[4]: 固定数据 array[5]: 酒精气体实时 A/D 值高字节 array[6]: 酒精气体实时 A/D 值低字节 array[7]: 酒精气体浓度等级，有效值 1~4，代表等级 1~等级 4 array[8]: 传感器状态 bit 0=1: 正在标定 bit 1~bit 4: 保留 bit 5=1 标定完成 bit 6=1: 模块故障 bit 7=1: 模块报警 array[9]: 标定倒计时，默认标定时间 60s，为 0 代表标定完成 array[10]: 上电预热倒计时，预热时间 180s，为 0 代表预热完成 array[11]: 软件版本号高字节；如 0x11，表示 1.1.x array[12]: 软件版本号低字节；如 0x02，表示 x.x.2 array[13]: 校验码；前 13 字节数据之和取低 8 位，取反再加一
	返回值	void
	备注	此函数需在“if (isInfoAvailable() == true)”后使用

参数设置函数		
10	uint8_t writeCommand(uint8_t cmd, uint8_t addr, uint8_t param)	
	描述	写入参数至模块
	参数	cmd: 指令码 addr: 地址码 param: 需要写入的参数
	返回值	执行情况 ⁽¹⁾
	备注	查阅 BM22S3421-1 数据手册的指令表 (特殊修改指令)
通用函数		
11	uint8_t reset()	
	描述	复位模块
	参数	—
	返回值	执行情况 ⁽¹⁾
	备注	—
12	uint8_t requestInfoPackage(uint8_t array[])	
	描述	主动获取模块的所有信息
	参数	array[0]~array[4]: 固定数据 array[5]: 酒精气体实时 A/D 值高字节 array[6]: 酒精气体实时 A/D 值低字节 array[7]: 酒精气体浓度等级, 有效值 1~4, 代表等级 1~等级 4 array[8]: 传感器状态 bit 0=1: 正在标定 bit 1~bit 4: 保留 bit 5=1 标定完成 bit 6=1: 模块故障 bit 7=1: 模块报警 array[9]: 标定倒计时, 默认标定时间 60s, 为 0 代表标定完成 array[10]: 上电预热倒计时, 预热时间 180s, 为 0 代表预热完成 array[11]: 软件版本号高字节; 如 0x11, 表示 1.1.x array[12]: 软件版本号低字节; 如 0x02, 表示 x.x.2 array[13]: 校验码; 前 13 字节数据之和取低 8 位, 取反再加一
	返回值	执行情况 ⁽¹⁾
	备注	一般用于主动获取数据, 建议在除能串口自动输出功能后使用 ⁽²⁾
13	uint8_t restoreDefault()	
	描述	将模块的参数恢复至出厂设置
	参数	—
	返回值	执行情况 ⁽¹⁾
	备注	—

14	uint8_t calibrateModule(uint8_t calibrateMode)	
	描述	酒精浓度等级标定
	参数	calibrateMode: 标定功能选择 0x01: 触发标定点 1 (等级 1、2 的判断阈值) 0x02: 触发标定点 2 (等级 2、3 的判断阈值) 0x03: 触发标定点 3 (等级 3、4 的判断阈值) 0x06: 触发标定检验, 更新标定点 1/2/3 的校验值
	返回值	执行情况 ⁽¹⁾
	备注	1. 用户可以通过此函数自定义浓度等级; 2. 标定点 1/2/3 的标定无固定顺序, 可以任意顺序标定; 且可以选择标定其中的一个 / 二个 / 三个标定点; 3. 标定点必须按照大小设置, 且标定点 1 < 标定点 2 < 标定点 3 4. 在标定点完成后, 需要使用标定校验指令, 从而使设备进入标定完成状态;

注: 1. 0 – 指令执行成功; 1 – 模块回复的数据错误; 2 – 通信超时

2. 串口自动输出功能:

使能: 可使用 writeCommand(0xe0,0x1e, AUTO_MODE)

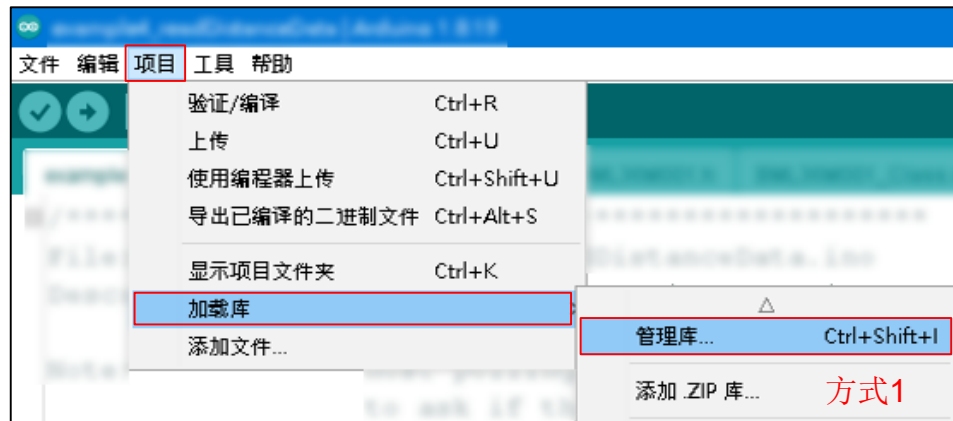
除能: 可使用 writeCommand(0xe0,0x1e, CMD_MODE)

Arduino Lib 下载及安装

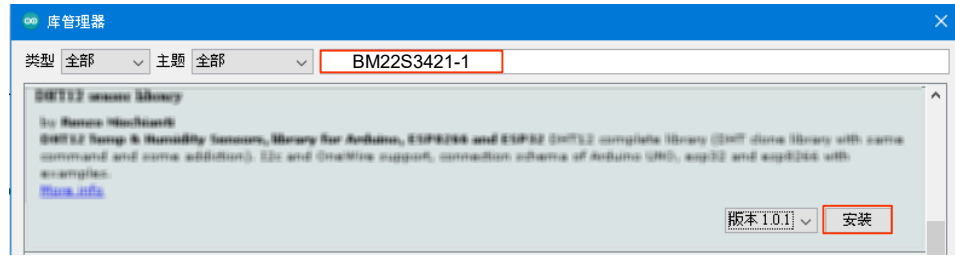
BM22S3421-1 Library: 可参考下面两种方法安装 BM22S3421-1 的 Arduino Library。

方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BM22S3421-1 → 安装



搜索安装流程 1

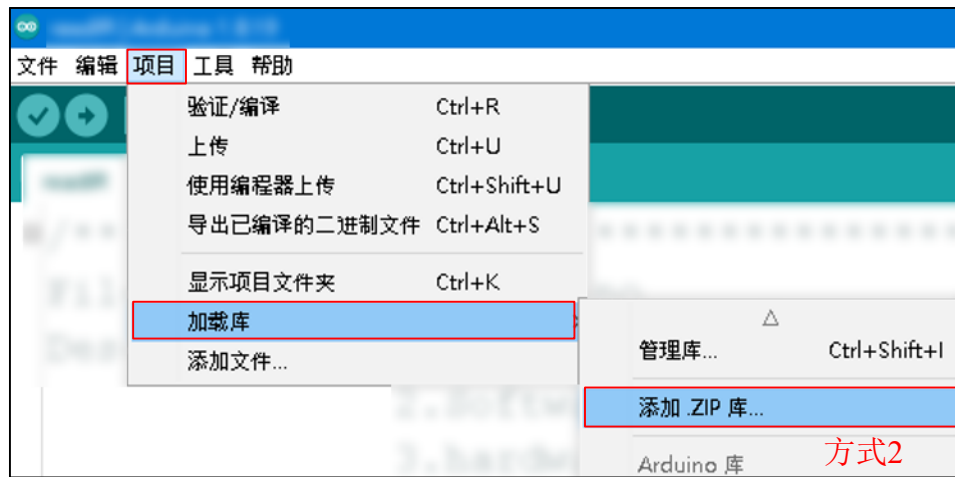


搜索安装流程 2

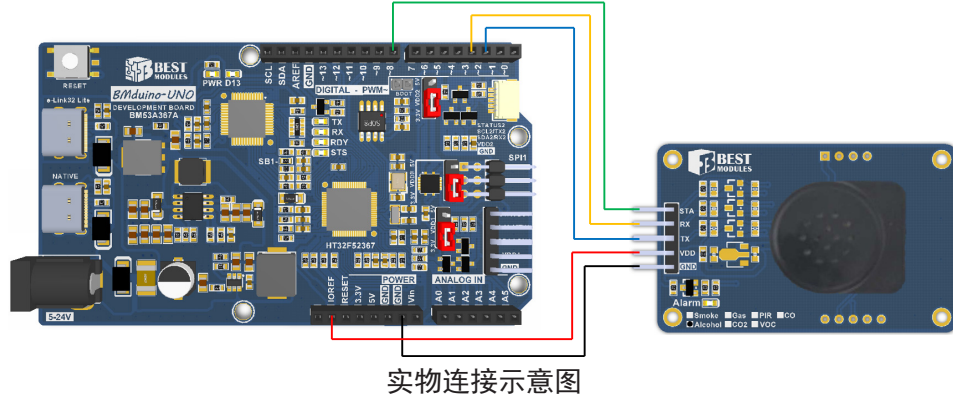
方式 2：添加 .ZIP 库，需提前下载 .ZIP 库。

下载方法：打开倍创官网 (<https://www.bestmodulescorp.com/bm22s3421-1.html>) 文件目录下的 Arduino 范例程序 (BM22S3421-1 Library)。

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



Arduino 范例



范例：readAlcLevelAndADValue

范例实现功能：实现功能：接收模块每秒自动输出的信息，并将酒精气体的浓度等级和 A/D 值打印到串口监视器

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM22S3421-1) → 选择范例 (readAlcLevelAndADValue)
2. 示例说明：
 - a. 构建对象 & 初始化模块

```
#include < BM22S3421-1.h> // 调用 BM22S3421-1 库
/* 创建数组、变量用于存放数据 */
uint8_t moduleInfo[14] = {0};
uint16_t ADValue, AlcLevel;
/* 创建对象 & 设置 Software serial 引脚 */
BM22S3421_1 Alc(8, 2, 3); // Software serial: D8:STATUS, D2:RX, D3:TX
void setup()
{
    Alc.begin(); // 模块初始化
    Serial.begin(9600); // 串口监视器初始化, 波特率 9600bps
    /* 等待模块预热结束 */
    Serial.println("Module preheating...(about 3 mins)");
    preheatCountdown(); // 等待模块预热结束
    Serial.println("End of module preheating.");
    Serial.println();
}
```

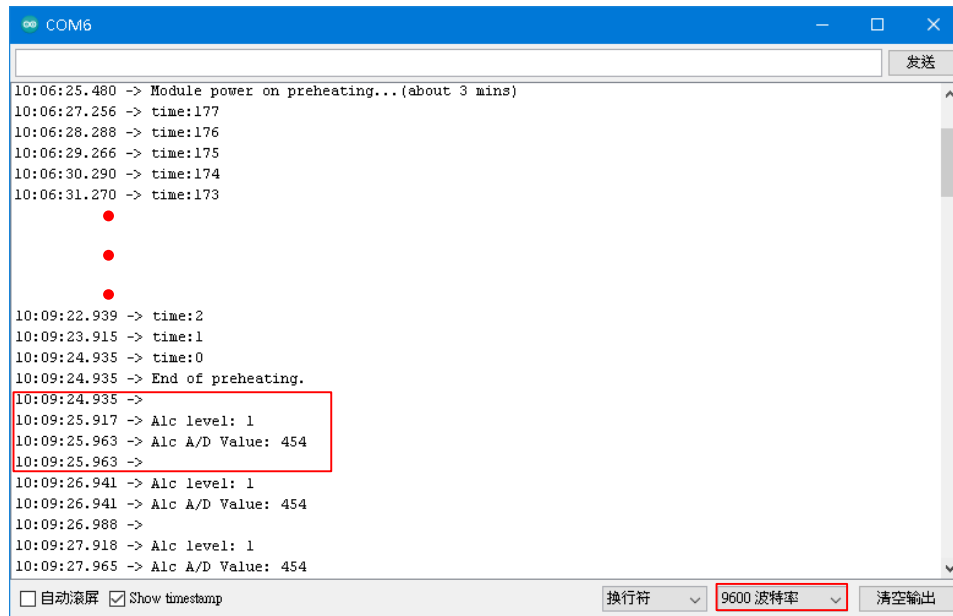
- b. 接收模块自动发送的数据

```
void loop()
{
    if (Alc.isInfoAvailable() == true) // 轮询是否接收到模块发出的数据
    {
        Alc.readInfoPackage(moduleInfo); // 读取模块发送的数据到moduleInfo[]
        printInfo(); // 打印模块的部分信息
    }
}
```


c. 根据接收到的数据，打印部分信息

```
void printInfo()
{
  /* 打印酒精浓度等级 */
  Serial.print("Alcohol level: ");
  AlcLevel = moduleInfo[7];
  Serial.println(AlcLevel);
  /* 打印当前酒精气体浓度的 A/D 值 (12-bit) */
  Serial.print("Alcohol A/D Value: ");
  ADValue = ((uint16_t)moduleInfo[5] << 8) + moduleInfo[6];
  Serial.println(ADValue);
  Serial.println();
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示如下：



Copyright® 2024 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。