



第二代接近感应模块

# **BM32S2031-1**

## **Arduino Library V1.0.3 说明**

版本: V1.20 日期: 2025-06-04

[www.bestmodulescorp.com](http://www.bestmodulescorp.com)

## 目录

简介 .....	3
Arduino Lib 函数 .....	3
Arduino Lib 下载及安装 .....	8
Arduino 范例 .....	9
范例：readIR .....	9

## 简介

BM32S2031-1 是倍创推出的第二代接近感应模块，使用 UART 通信方式。本文档对 BM32S2031-1 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明。范例使用 BMS31M001/BMS31M001A 模块，演示了获取版本号和有感应时获取信号值等功能。

适用型号：

型号	说明
BM32S2031-1	第二代接近感应模块
BMS31M001	板载 BM32S2031-1 模块
BMS31M001A	板载 BM32S2031-1 模块

## Arduino Lib 函数

Arduino Lib 名称: BM32S2031-1		Lib 版本: V1.0.3
<b>构造函数 &amp; 初始化</b>		
1	BM32S2031_1(HardwareSerial *theSerial=&Serial)	
	描述	构造函数，使用硬件 Serial 接口
	参数	*theSerial: 选择硬件 Serial 接口 (默认 Serial 接口)
	返回值	—
	备注	—
2	BM32S2031_1(uint8_t rxPin, uint8_t txPin)	
	描述	构造函数，使用软件串行接口
	参数	rxPin: RX 引脚，连接 BM32S2031-1 的 D 引脚或 BMS31M001 的 TX 引脚 txPin: TX 引脚，连接 BM32S2031-1 的 C 引脚或 BMS31M001 的 RX 引脚
	返回值	—
	备注	—
3	void begin(uint16_t baud=BM32S2031_1_BAUD)	
	描述	模块初始化
	参数	baud: 波特率 9600 (BM32S2031_1_BAUD): 固定为 9600bps
	返回值	void
	备注	—
<b>功能函数</b>		

4	uint8_t getIRStatus()	
	描述	获取 IR 感应状态
	参数	—
	返回值	IR 感应状态 0x00: 无触发 0x01: 有触发
	备注	信号值大于接近感应触发阈值时为有触发, 否则为无触发
5	int16_t readIRSignalValue()	
	描述	读取信号值
	参数	—
	返回值	信号值
	备注	信号值 = 环境值 - 参考值
6	uint8_t distanceLearning()	
	描述	距离学习
	参数	—
	返回值	距离学习情况 0x00: 学习成功 0x01: 学习失败
	备注	—
7	uint8_t reset()	
	描述	模块复位
	参数	—
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
8	uint8_t saveSettings()	
	描述	将当前参数保存至 EEPROM
	参数	—
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
9	uint8_t restoreSettings()	
	描述	把存储在 EEPROM 的参数设置为当前参数
	参数	—
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

10	int16_t getFWVer()	
	描述	获取版本号
	参数	—
	返回值	版本号
	备注	举例：返回值为 0x0100，则版本号为 V1.00
<b>设置 &amp; 读取函数</b>		
11	uint8_t getIRThreshold()	
	描述	获取接近感应触发阈值
	参数	—
	返回值	接近感应触发阈值
	备注	—
12	uint8_t getIRDebounce()	
	描述	获取 IR 触发去抖次数
	参数	—
	返回值	IR 触发去抖次数
	备注	—
13	uint8_t getIRResponseTime()	
	描述	获取红外线侦测反应速度
	参数	—
	返回值	侦测反应速度参数 0x00: 8ms 0x01: 16ms 0x02: 32ms 0x03: 64ms 0x04: 128ms 0x05: 256ms 0x06: 0.5s 0x07: 1s 0x08: 快速模式
	备注	—
14	uint8_t getIROutputTime()	
	描述	获取输出延时时间 (只在一般高低电平输出模式有效)
	参数	—
	返回值	输出延时时间参数 0: 不启动 1~255: (1~255) 秒
	备注	—

15	uint8_t getMode()	
	描述	获取模块功能 ( I/O 模式下的 TX 脚位功能 )
	参数	—
	返回值	功能参数 Bit 4~0: 预留 Bit 5: 输出模式选择 0: High/Low ( 一般高低电平输出模式 ) 1: Pulse/Toggle ( 脉冲切换模式 ) Bit 6: Pulse/Toggle 模式选择 ( 仅在 Bit 5=1 时有效 ) 0: Pulse ( 物体靠近后, 每 1 秒钟开输出 200μs ) 1: Toogle ( 物体靠近, 开输出; 物体离开后再次靠近, 关输出 ) Bit 7: 输出电平设定 0: 正常高电平, 低电平有效 1: 正常低电平, 高电平有效
	备注	—
16	uint8_t getIRCurrent()	
	描述	获取发射电流
	参数	—
	返回值	发射电流参数 参数 > 63, 发射电流 = (( 参数 - 64 )×5+5) mA 参数 ≤ 63, 发射电流 = ( 参数 × 5+5 ) mA
	备注	—
17	uint8_t setIRThreshold(uint8_t threshold=16)	
	描述	设置接近感应触发阈值
	参数	threshold: 接近感应触发阈值, 范围 16~180, 默认 16
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
18	uint8_t setIRDebounce(uint8_t debounce=0)	
	描述	设置 IR 触发去抖次数
	参数	debounce: IR 触发去抖次数, 范围 0~15, 默认 0
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

19	uint8_t setIRResponseTime(uint8_t responseTime=SPEED_500_MS)	
	描述	设置红外线侦测反应速度
	参数	responseTime: 侦测反应速度参数, 默认 0x06(SPEED_500_MS) 0x00 (SPEED_8_MS): 8ms 0x01 (SPEED_16_MS): 16ms 0x02 (SPEED_32_MS): 32ms 0x03 (SPEED_64_MS): 64ms 0x04 (SPEED_128_MS): 128ms 0x05 (SPEED_256_MS): 256ms 0x06 (SPEED_500_MS): 500ms 0x07 (SPEED_1_S): 1s 0x08 (SPEED_FASTMODE): 快速模式
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
20	uint8_t setIROutputTime(uint8_t outputTime=0)	
	描述	设置输出延时时间 (只在一般高低电平输出模式有效)
	参数	outputTime: 输出延时时间, 默认 0 0: 不启动 1~255: (1~255) 秒
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
21	uint8_t setMode(uint8_t mode=0b00010011)	
	描述	设置模块功能 (I/O 模式下的 TX 引脚功能)
	参数	mode: 功能参数, 默认 0b00010011 Bit 4~0: 预留 Bit 5: 输出模式选择 0: High/Low (一般高低电平输出模式) 1: Pulse/Toggle (脉冲切换模式) Bit 6: Pulse/Toggle 模式选择 (仅在 Bit 5=1 时有效) 0: Pulse (物体靠近后, 每 1 秒钟开输出 200μs) 1: Toogle (物体靠近, 开输出; 物体离开后再次靠近, 关输出) Bit 7: 输出电平设定 0: 正常高电平, 低电平有效 1: 正常低电平, 高电平有效
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

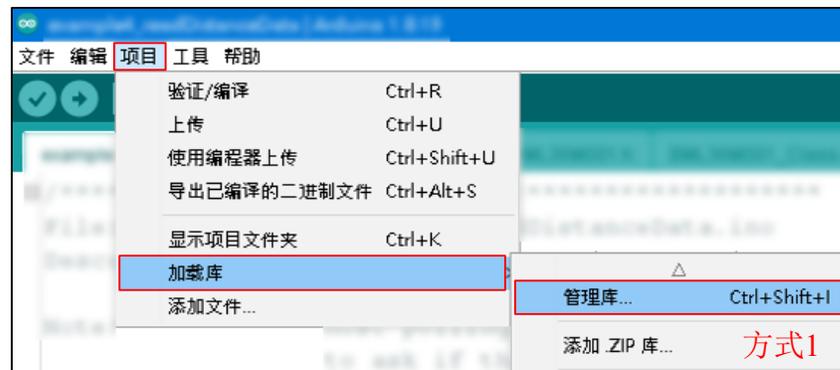
22	uint8_t setIRCurrent(uint8_t current=78)	
	描述	设置发射电流
	参数	current: 发射电流参数, 范围 0~127, 默认 78 current>63, 发射电流 = ((current-64)×5+5) mA current≤63, 发射电流 = (current×5+5) mA
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

## Arduino Lib 下载及安装

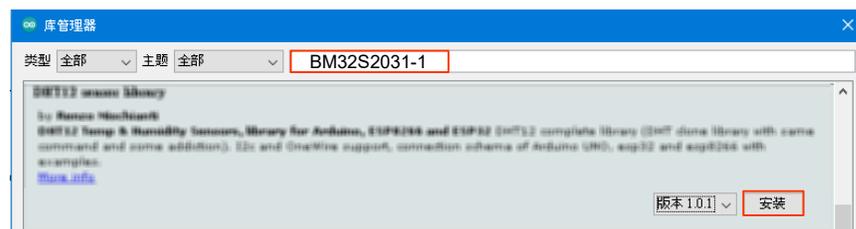
BM32S2031-1 Library: 可参考下面两种方法安装 BM32S2031-1 的 Arduino Library。

### 方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BM32S2031-1 → 安装



搜索安装流程 1

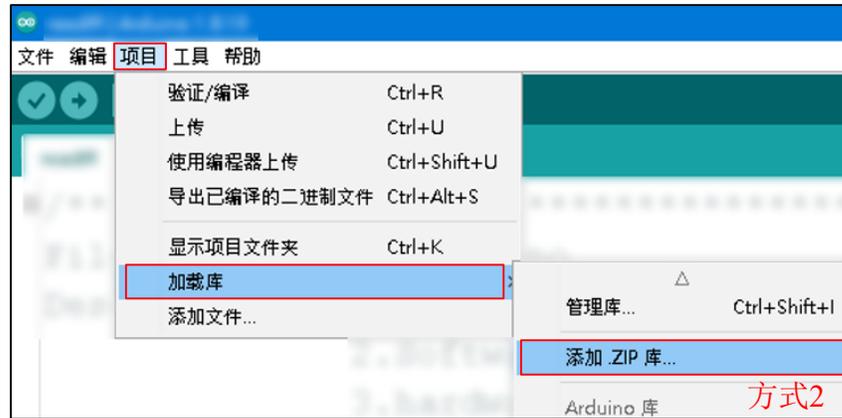


搜索安装流程 2

### 方式 2: 添加 .ZIP 库, 需提前下载 .ZIP 库

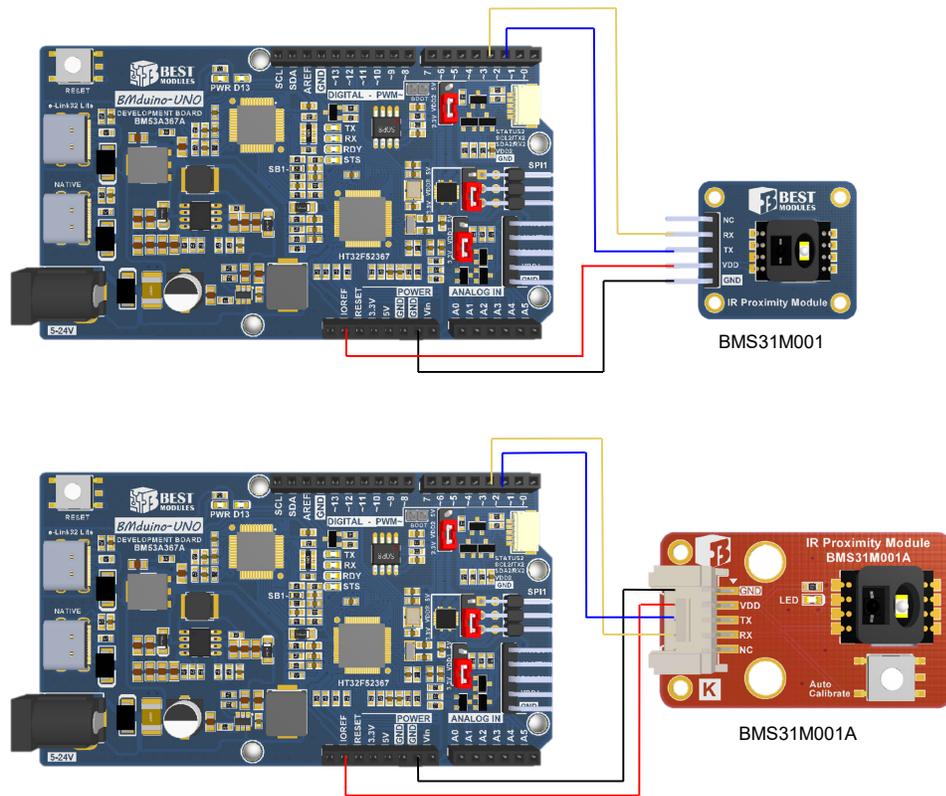
下载方法: 打开倍创官方网站 (<https://www.bestmodulescorp.com/bm32s2031-1.html>), 下载“文件”菜单下的 Arduino 范例程序 (BM32S2031-1 Library)。

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



## Arduino 范例

范例：readIR



实物连接示意图

范例实现功能：有感应时获取当前的信号值，并在串口监视器上显示。

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM32S2031-1) → 选择范例 (readIR)

## 2. 示例说明:

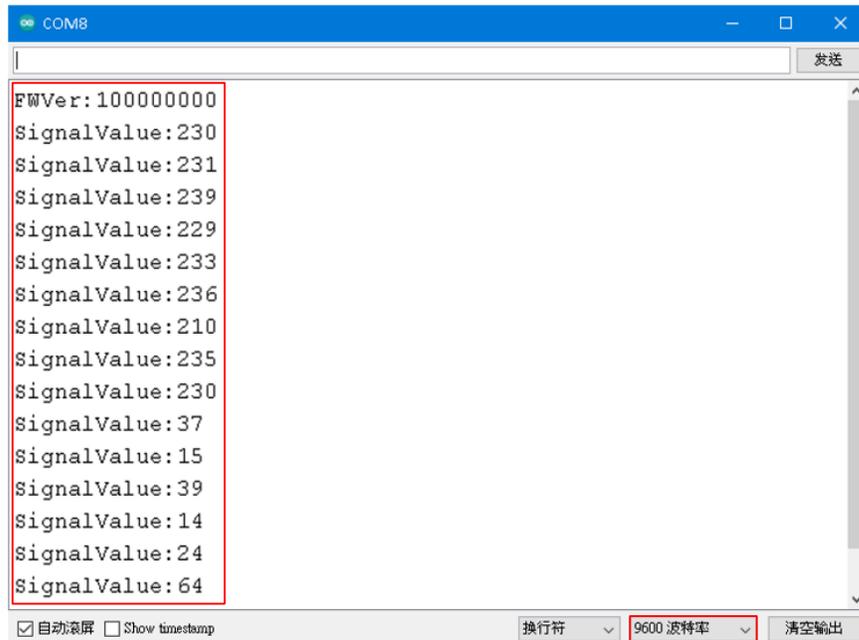
### a. 创建对象 & 模块初始化及配置

```
#include "BM32S2031-1.h"
BM32S2031_1 BMS31(2,3); // 创建对象, TX 引脚连接 D2, RX 引脚连接 D3
int16_t FWVer = 0;
uint8_t irStatus = 0;
int16_t signalValue = 0;
void setup()
{
  BMS31.begin(); // 初始化模块
  Serial.begin(9600); // 配置串口监视器
  FWVer = BMS31.getFWVer(); // 获取版本号
  Serial.print("FWVer:");
  Serial.println(FWVer,BIN);
}
```

### b. 有感应时获取信号值并在串口监视器中显示

```
void loop()
{
  /** 获取感应状态, 判断是否触发 **/
  irStatus = BMS31.getIRStatus(); // 获取感应状态
  if(irStatus) // 判断是否有触发
  {
    /** 获取当前信号值, 并在串口监视器上显示 **/
    signalValue = BMS31.readIRSignalValue(); // 获取信号值
    Serial.print("SignalValue:");
    Serial.println(signalValue); // 在串口监视器中显示当前信号值
  }
}
```

## 3. 打开串口监视器, 波特率选择 9600; 串口监视器显示如下:



```
COM8
FWVer: 100000000
SignalValue: 230
SignalValue: 231
SignalValue: 239
SignalValue: 229
SignalValue: 233
SignalValue: 236
SignalValue: 210
SignalValue: 235
SignalValue: 230
SignalValue: 37
SignalValue: 15
SignalValue: 39
SignalValue: 14
SignalValue: 24
SignalValue: 64
 自动滚屏  Show timestamp 换行符 9600 波特率 清空输出
```

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。