



1D 红外线手势控制数字模块

BM32S3021-1

Arduino Library V1.0.4 说明

版本: V1.30 日期: 2025-05-28

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: getGesture	8

简介

BM32S3021-1 是倍创推出的 1D 红外线手势控制数字模块，使用 UART 通信方式。本文档对 BM32S3021-1 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例使用 BMS31M002/BMS31M002A 模块，演示了识别手势是左滑还是右滑的功能。

适用型号：

型号	说明
BM32S3021-1	1D 红外线手势控制数字模块
BMS31M002	板载 BM32S3021-1 模块
BMS31M002A	板载 BM32S3021-1 模块

Arduino Lib 函数

Arduino Lib 名称: BM32S3021-1		Lib 版本: V1.0.4
构造函数 & 初始化		
1	BM32S3021_1(uint8_t intPin, HardwareSerial *theSerial=&Serial)	
	描述	构造函数，使用硬件 Serial 接口
	参数	intPin: INT 连接的引脚 *theSerial: 选择硬件 Serial 接口 (默认 Serial 接口)
	返回值	—
	备注	—
2	BM32S3021_1(uint8_t intPin, uint8_t rxPin, uint8_t txPin)	
	描述	构造函数，使用软件串行接口
	参数	intPin: INT 引脚，连接 BM32S3021-1 的 INT 引脚或 BMS31M002 的 INT 引脚 rxPin: RX 引脚，连接 BM32S3021-1 的 D 引脚或 BMS31M002 的 TX 引脚 txPin: TX 引脚，连接 BM32S3021-1 的 C 引脚或 BMS31M002 的 RX 引脚
	返回值	—
	备注	—
3	void begin(uint16_t baud=9600)	
	描述	模块初始化
	参数	baud: 波特率，固定为 9600bps
	返回值	void
	备注	—
功能函数		

4	uint8_t getINT()	
	描述	获取 INT 引脚电平
	参数	—
	返回值	INT 引脚电平 0x00: 低电平, 检测到手势 0x01: 高电平, 没检测到手势
	备注	—
5	uint8_t getIRStatus()	
	描述	获取手势感应状态
	参数	—
	返回值	手势感应状态 Bit 3: 校准状态 0: 一般模式 1: 校准中 Bit 2: 左滑状态 0: 左滑结束 1: 左滑成立 Bit 1: 右滑状态 0: 右滑结束 1: 右滑成立 Bit 0: 接近感应状态 0: 无接近 1: 接近
	备注	—
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 getIRGestureNum()	
	描述	获取滑动次数
	参数	—
	返回值	滑动次数 (-127~128) -n = 右滑次数 n = 左滑次数
	备注	—
8	uint16_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
9	uint8_t reset()	
	描述	复位模块
	参数	—
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

设置 & 读取函数		
10	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	—
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 getIRQTrigerTime()	
	描述	获取手势成立时 IRQ 触发时间
	参数	—
	返回值	返回值：手势成立时 IRQ 触发时间参数 IRQ 触发时间 = (返回值 × 4) ms
	备注	—
13	uint8_t getIRContinityGestureTime()	
	描述	获取累计连续滑动时长
	参数	—
	返回值	返回值：累计连续滑动时长参数 累计连续滑动时长 = (返回值 × 64) ms
	备注	—
14	uint8_t getIRFastestGestureTime()	
	描述	获取最快手势判断时间
	参数	—
	返回值	返回值：最快手势判断时间参数 最快手势判断时间 = (20 + (返回值 × 4)) ms
	备注	—
15	uint8_t getIRSlowestGestureTime()	
	描述	获取最慢手势判断时间
	参数	—
	返回值	返回值：最慢手势判断时间参数 最慢手势判断时间 = (返回值 × 16) ms
	备注	—
16	uint8_t setIRDebounce(uint8_t debounce=7)	
	描述	设置 IR 触发去抖次数
	参数	debounce: IR 触发去抖次数, 范围 0~255, 默认 7
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

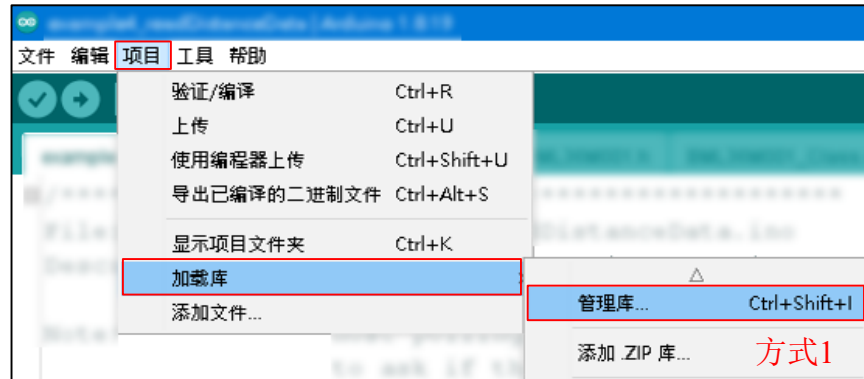
17	uint8_t setIRThreshold(uint8_t threshold=16)	
	描述	设置手势触发阈值
	参数	threshold: 手势触发阈值, 范围 10~200, 默认 16
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
18	uint8_t setIRQTrigerTime(uint8_t irqTime=50)	
	描述	设置手势成立时 IRQ 触发时间
	参数	irqTime: IRQ 触发时间参数, 范围 0~255, 默认 50 IRQ 触发时间 = (irqTime×4) ms
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
19	uint8_t setIRContinutyGestureTime(uint8_t irTime=30)	
	描述	设置累计连续滑动时长
	参数	irqTime: 累计连续滑动时长参数, 范围 0~255, 默认 30 累计连续滑动时长 = (irqTime×64) ms
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
20	uint8_t setIRFastestGestureTime(uint8_t irTime=0)	
	描述	设置最快手势判断时间
	参数	irqTime: 最快手势判断时间参数, 范围 0~200, 默认 0 最快手势判断时间 = (20+(irqTime×4)) ms
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—
21	uint8_t setIRSlowestGestureTime(uint8_t irTime=80)	
	描述	设置最慢手势判断时间
	参数	irqTime: 最慢手势判断时间参数, 范围 0~200, 默认 80 最慢手势判断时间 = (irqTime×16) ms
	返回值	执行情况 0x00: 成功 0x01: 失败
	备注	—

Arduino Lib 下载及安装

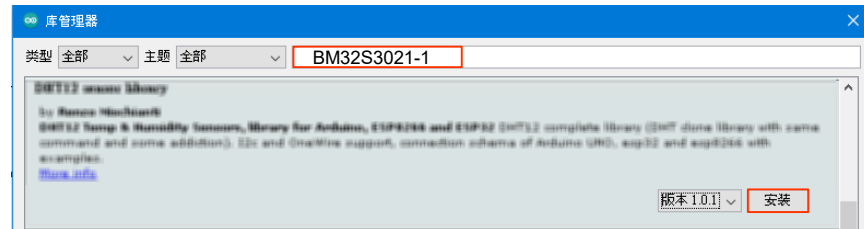
BM32S3021-1 Library: 可参考下面两种方法安装 BM32S3021-1 的 Arduino Library。

方式 1: 搜索安装

搜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BM32S3021-1 → 安装



搜索安装流程 1

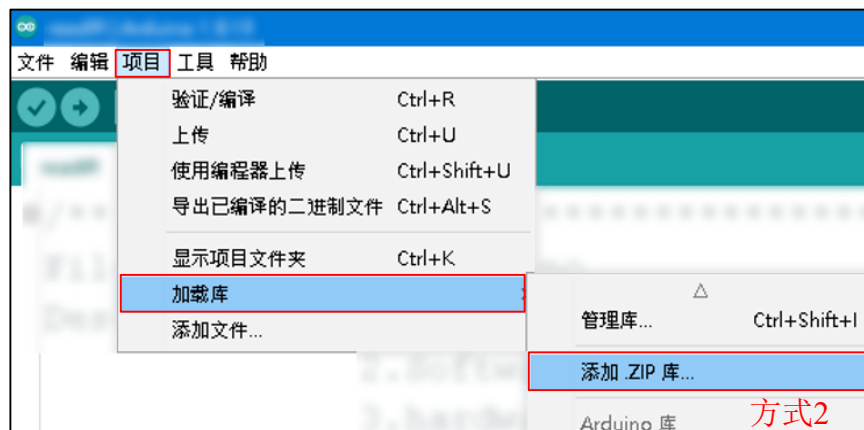


搜索安装流程 2

方式 2: 添加 .ZIP 库, 需提前下载 .ZIP 库

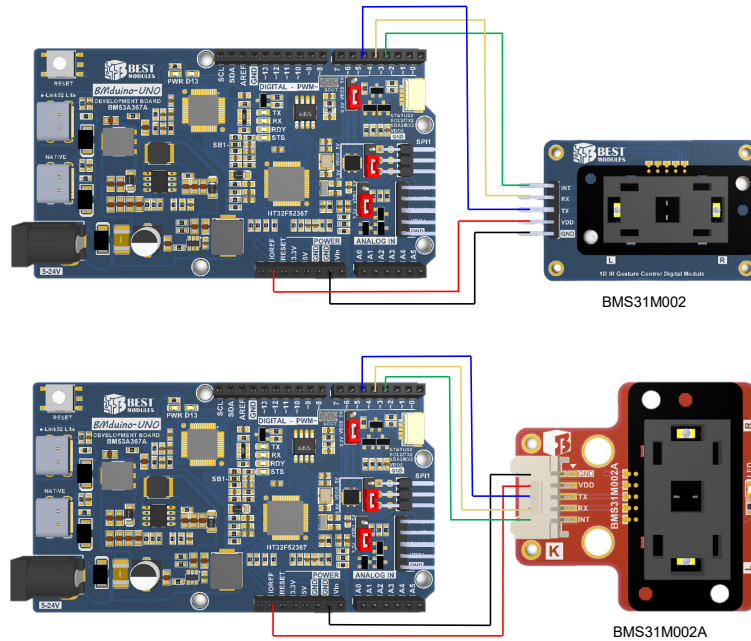
下载方法: 打开倍创官方网站 (<https://www.bestmodulescorp.com/bm32s3021-1.html>) “文件” 目录下的 Arduino 范例程序 (BM32S3021-1 Library)。

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



Arduino 范例

范例：getGesture



实物连接示意图

范例实现功能：有感应时获取手势感应状态，判断是左滑还是右滑，并在串口监视器上显示。

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM32S3021-1) → 选择范例 (getGesture)

2. 示例说明：

a. 创建对象 & 模块初始化及配置

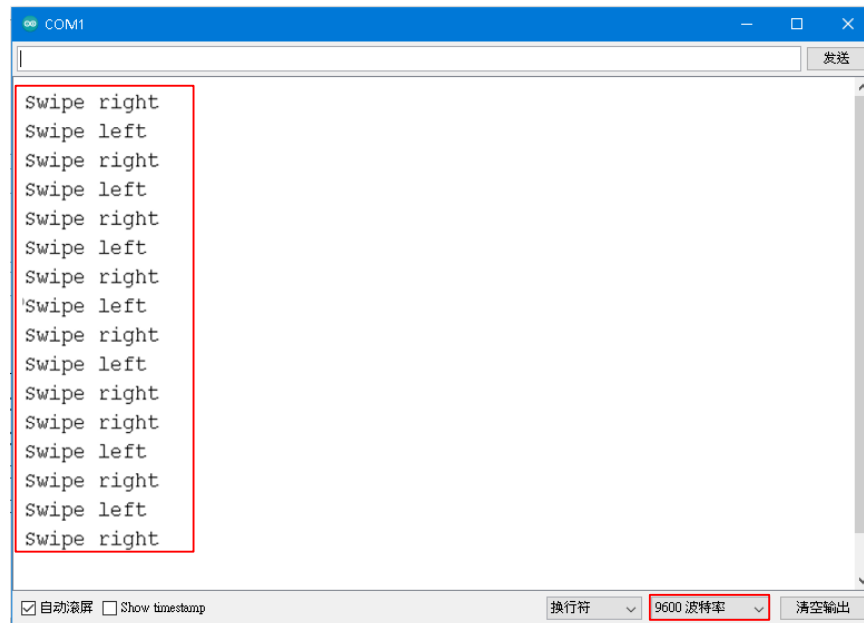
```
#include "BM32S3021-1.h"
BM32S3021_1 myGesture(3,5,4); // 创建对象，INT 引脚连接开发板 D3，TX 引脚连接
// 开发板 D5，RX 引脚连接开发板 D4

uint8_t irStatus = 0;
void setup()
{
    myGesture.begin(); // 初始化模块
    Serial.begin(9600); // 配置串口监视器
}
```

b. 有触发时判断手势，并在串口监视器中显示


```
void loop()
{
  if(!myGesture.getINT ())           // 获取 INT 线状态，判断是否触发
  {
    /** 判断手势，并在串口监视器上显示 **/
    irStatus = myGesture.getIRStatus();// 获取手势感应状态
    if(!(irStatus&0x08))
    {
      if(irStatus&0x02)               // 判断手势是否为右滑
      {
        Serial.println("Swipe right");// 在串口监视器中显示 Swipe right
      }
      else if(irStatus&0x04)          // 判断手势是否为左滑
      {
        Serial.println("Swipe left");// 在串口监视器中显示 Swipe left
      }
    }
  }
}
```

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。