接近 & 环境光侦测模块

# BMS33M332
# Arduino Library 说明

版本：V1.00　日期：2023-05-22

www.bestmodulescorp.com

# 目录

# 简介

BMS33M332 是倍创推出的接近 & 环境光侦测模块，使用 I²C 通信方式。本文档对 BMS33M332 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了读取接近感应以及环境光的 A/D 值等功能。

# Arduino Lib 函数

| Arduino Lib 名称：BMS33M332 | | Lib 版本：V1.0.1 |
|---|---|---|
| **构造函数 & 初始化** | | |
| 1 | BMS33M332(uint8_t intPin, TwoWire *theWire=&Wire) | |
| | 描述 | 构造函数 |
| | 参数 | intPin：INT 引脚<br>*theWire：I²C 通信接口选择 |
| | 返回值 | — |
| | 备注 | — |
| 2 | void begin(uint8_t addr=BMS33M332_IICADDR) | |
| | 描述 | 模块初始化 |
| | 参数 | Addr：从机地址，0x47 |
| | 返回值 | void |
| | 备注 | — |
| **功能函数** | | |
| 3 | uint16_t readRawProximity() | |
| | 描述 | 获取接近感应 AD 值 |
| | 参数 | — |
| | 返回值 | 接近感应 AD 值 |
| | 备注 | — |
| 4 | uint16_t readRawAmbient() | |
| | 描述 | 获取环境光 AD 值 |
| | 参数 | — |
| | 返回值 | 环境光 AD 值 |
| | 备注 | 环境光值 = AD 值 × (0.8204/alsIt/alsGain)，单位：LUX。<br>其中 alsIt=2^time，可通过 setALSIntegrationTime 函数设置；alsGain 可通过 setALSGain 函数设置。 |
| 5 | float readAmbient() | |
| | 描述 | 获取环境光值 |
| | 参数 | — |
| | 返回值 | 环境光值，单位 LUX |
| | 备注 | — |

| | | |
|---|---|---|
| 6 | uint8_t getPDTID() | |
| | 描述 | 获取产品的 ID 号 |
| | 参数 | — |
| | 返回值 | 产品的 ID 号 |
| | 备注 | — |
| 7 | void setINT(uint16_t thdh, uint16_t thdl, bool isEnable=true) | |
| | 描述 | 设置中断触发，设置阈值以及使能。 |
| | 参数 | thdh：PS 上阈值<br>thdl：PS 下阈值<br>isEnable：<br>true：使能 (default)<br>false：除能 |
| | 返回值 | void |
| | 备注 | — |
| 8 | uint8_t getINT() | |
| | 描述 | 获取 INT 引脚电平 |
| | 参数 | — |
| | 返回值 | INT 引脚电平<br>0：LOW<br>1：HIGH |
| | 备注 | INT 引脚电平在接近感应 AD 值超出高阈值时 , 变为 LOW；并在回到低阈值以下时，恢复为 HIGH |
| 9 | uint8_t getPositionStatus() | |
| | 描述 | 获取位置状态，检测是否有满足条件的触发 |
| | 参数 | — |
| | 返回值 | 位置状态<br>0：中断触发<br>1：无中断触发 |
| | 备注 | INT 引脚电平在接近感应 AD 值超出高阈值时 , 变为 LOW；并在回到低阈值以下时，恢复为 HIGH |
| 10 | void reset() | |
| | 描述 | 复位 |
| | 参数 | — |
| | 返回值 | void |
| | 备注 | — |
| 11 | void writeReg(uint8_t addr, uint8_t data) | |
| | 描述 | 写寄存器 |
| | 参数 | addr：寄存器地址<br>data：要写入的数据 |
| | 返回值 | void |
| | 备注 | — |

| | | |
|---|---|---|
| 12 | uint8_t readReg(uint8_t addr) | |
| | 描述 | 读寄存器 |
| | 参数 | addr：寄存器地址 |
| | 返回值 | 寄存器值 |
| | 备注 | — |
| 13 | void readReg(uint8_t addr, uint8_t rBuf[], uint8_t rLen) | |
| | 描述 | 连读寄存器，rLen 字节，存储于数组 rBuf[] |
| | 参数 | addr：寄存器地址<br>rBuf[]：用于存储读取的数据<br>rLen：连读的字节数 |
| | 返回值 | void |
| | 备注 | — |
| **参数读取 & 配置** | | |
| 14 | uint8_t getLEDcurrent() | |
| | 描述 | 获取 LED 的恒定电流参数 |
| | 参数 | — |
| | 返回值 | LED 的恒定电流参数<br>0：3.125mA<br>1：6.25mA<br>2：12.5mA<br>3：25mA<br>4：50mA<br>5：100mA<br>6：150mA |
| | 备注 | — |
| 15 | uint8_t getMeasureIntervalTime() | |
| | 描述 | 获取测量间隔时间参数 |
| | 参数 | — |
| | 返回值 | 测量间隔时间参数 time |
| | 备注 | 测量间隔时间 = (time+1)×1.54 ms |
| 16 | uint16_t getPSHighThreshold() | |
| | 描述 | 获取 PS 上阈值 |
| | 参数 | — |
| | 返回值 | PS 上阈值 |
| | 备注 | — |
| 17 | uint16_t getPSLowThreshold() | |
| | 描述 | 获取 PS 下阈值 |
| | 参数 | — |
| | 返回值 | PS 下阈值 |
| | 备注 | — |

| | | |
|---|---|---|
| 18 | uint16_t getALSHighThreshold() | |
| | 描述 | 获取 ALS 上阈值 |
| | 参数 | — |
| | 返回值 | ALS 上阈值 |
| | 备注 | — |
| 19 | uint16_t getALSLowThreshold() | |
| | 描述 | 获取 ALS 下阈值 |
| | 参数 | — |
| | 返回值 | ALS 下阈值 |
| | 备注 | — |
| 20 | void setLEDcurrent(uint8_t current) | |
| | 描述 | 设置 LED 恒定电流 |
| | 参数 | current：LED 恒定电流选择<br>0(CURRENT_3_125MA)：3.125mA<br>1(CURRENT_6_25MA)：6.25mA<br>2(CURRENT_12_5MA)：12.5mA<br>3(CURRENT_25MA)：25mA<br>4(CURRENT_50MA)：50mA<br>5(CURRENT_100MA)：100mA<br>6(CURRENT_150MA)：150mA |
| | 返回值 | void |
| | 备注 | — |
| 21 | void setMeasureIntervalTime(uint8_t time, bool isEnable=true) | |
| | 描述 | 设置测量间隔时间 |
| | 参数 | time：测量间隔时间参数，测量间隔时间 = (time+1)×1.54 ms<br>isEnable：<br>ture：使能 (default)<br>false：除能 |
| | 返回值 | void |
| | 备注 | — |
| 22 | void setPSHighThreshold(uint16_t thdh) | |
| | 描述 | 设置 PS 上阈值 |
| | 参数 | thdh：PS 上阈值 |
| | 返回值 | void |
| | 备注 | — |
| 23 | void setPSLowThreshold(uint16_t thdl) | |
| | 描述 | 设置 PS 下阈值 |
| | 参数 | thdl：PS 下阈值 |
| | 返回值 | void |
| | 备注 | — |

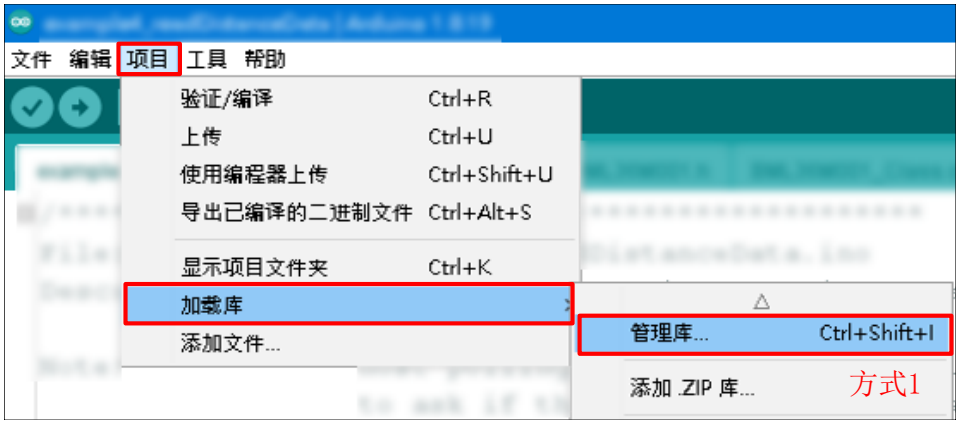
| | | |
|---|---|---|
| 24 | void setALSHighThreshold(uint16_t thdh) | |
| | 描述 | 设置 ALS 上阈值 |
| | 参数 | thdh：ALS 上阈值 |
| | 返回值 | void |
| | 备注 | — |
| 25 | void setALSLowThreshold(uint16_t thdl) | |
| | 描述 | 设置 ALS 下阈值 |
| | 参数 | thdl：ALS 下阈值 |
| | 返回值 | void |
| | 备注 | — |

# Arduino Lib 下载及安装

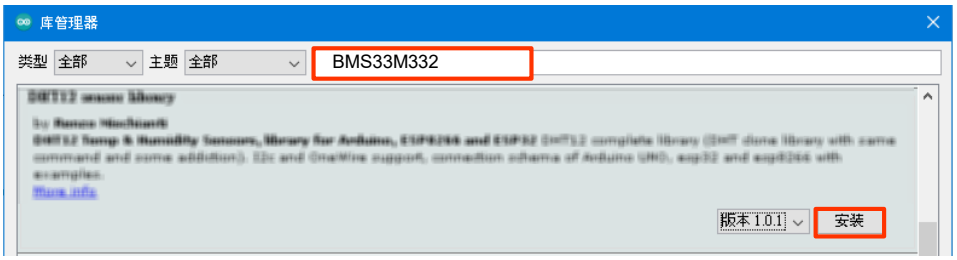
BMS33M332 Library：可参考下面两种方法安装 BMS33M332 的 Arduino Library

### 方式 1：搜索安装

搜索安装：Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BMS33M332 → 安装




搜索安装流程 **1**

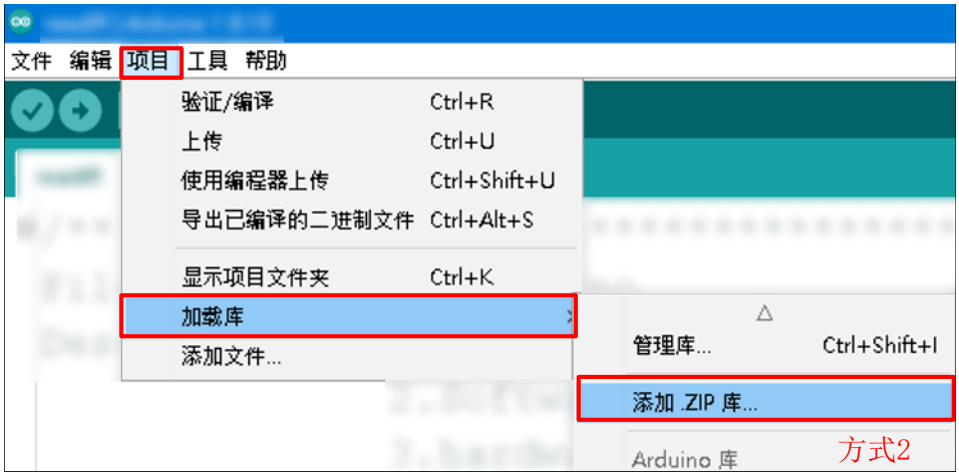



搜索安装流程 **2**

### 方式 2：添加 .ZIP 库，需提前下载 .ZIP 库

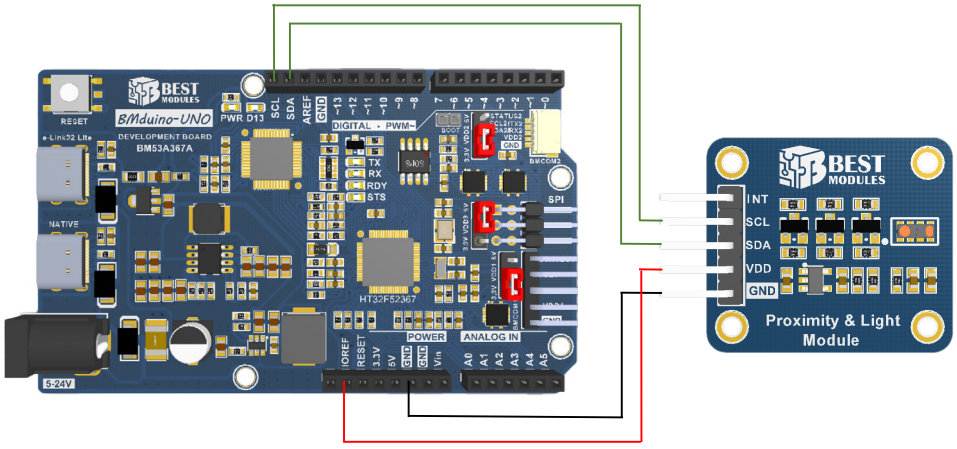
下载方法：打开倍创官方网站 (https://www.bestmodulescorp.com/bms33m332.html#tab-product2) 文件目录下的 Arduino 范例程序 (BMS33M332 Library)。

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




# Arduino 范例

## 范例 1：readAmbientAndProximity




实物连接示意图

范例 1 实现功能：以 I²C 方式与模块通信，获取接近感应以及环境光的 A/D 值，并在串口监视器上显示。

1. 范例打开：

   文件 → 示例 → Lib 选择 (BMS33M332) → 选择范例 (readAmbientAndProximity)

2. 示例说明：

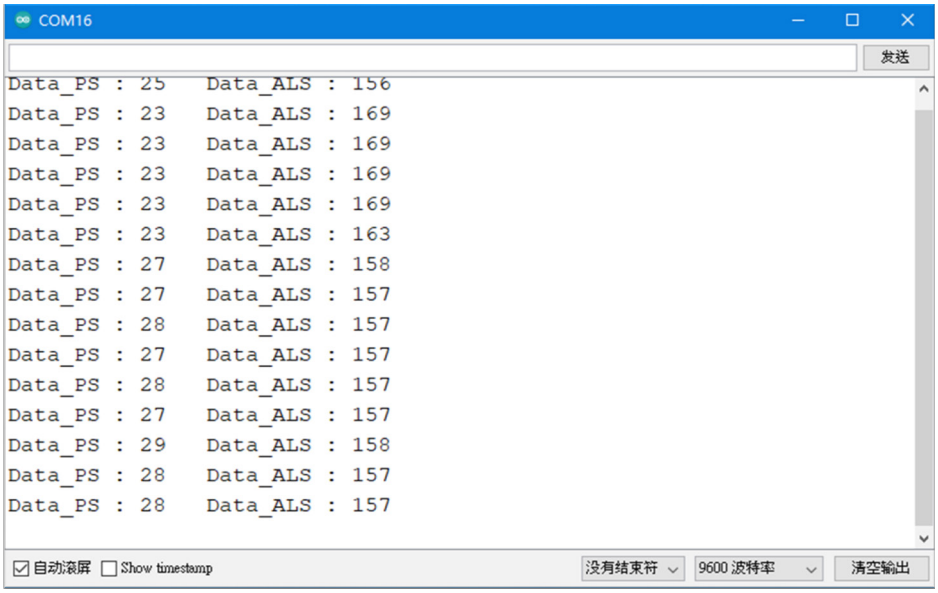
a. 构建 & 初始化对象

```
#include "BMS33M332.h"
BMS33M332  Alsps(8);    // 选择 Pin8 作为 INT 引脚
uint16_t alsValue;
uint16_t psValue;
void setup()
{
  Serial.begin(9600);   // 串口监视器初始化，波特率为 9600
  Alsps.begin();        // 模块初始化
}
```

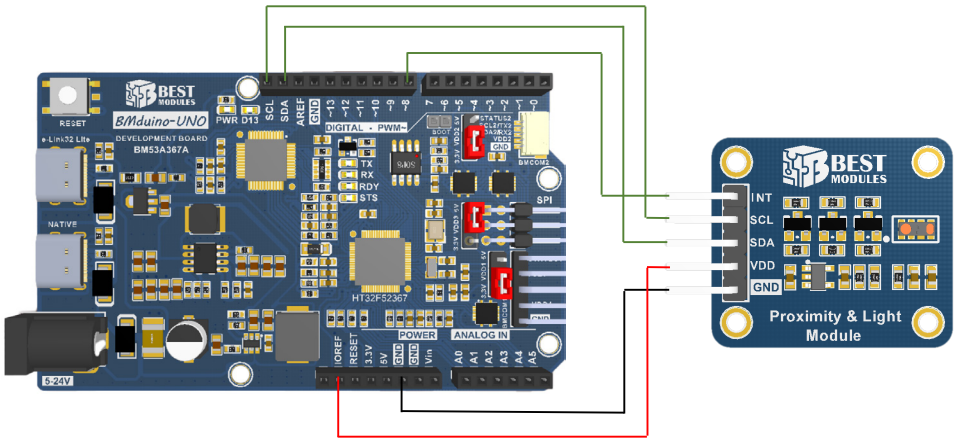
b. 获取接近感应以及环境光的 A/D 值并在串口监视器中显示

```
void loop()
{
  psValue = Alsps.readProximity();   // 获取接近感应的 A/D 值
  Serial.print("Data_PS : ");
  Serial.print(psValue);
  alsValue = Alsps.readAmbient();    // 获取环境光的 A/D 值
  Serial.print("Data_ALS : ");
  Serial.print(alsValue);
  Serial.println();
  delay(1000);
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示数据如下：



## 范例 2：getPositionStatus




实物连接示意图

范例 2 实现功能：以 I²C 方式与模块通信，判断是否有物体靠近，如果有符合设置的动作，结果会通过串口监视器显示。

1. 范例打开：

   文件 → 示例 → Lib 选择 (BMS33M332) → 选择范例 (getPositionStatus)

2. 示例说明：

   a. 构建 & 初始化对象

```
#include "BMS33M332.h"
BMS33M332  Alsps(8);      // 选择 Pin8 作为 INT 引脚
void setup()
{
  Serial.begin(9600);     // 串口监视器初始化，波特率为 9600
  Alsps.begin();          // 模块初始化
  Alsps.setINT(400,200); // 设置接近感应阈值：thdh=400，thdl=200
}
```

   b. 获取状态标志位判断是否有物体靠近，并在串口监视器中显示

```
void loop()
{
  if(Alsps.getPositionStatus() == 1)   // 获取状态标志位
  {
    Serial.println("No objects found ):");
  }
  else Serial.println("Find object!");
  delay(1000);
}
```

3. 打开串口监视器，波特率选择 9600；串口监视器显示数据如下：