



土壤温湿度检测模块

# **BM25S2621-1**

## **Arduino Library V1.0.2 说明**

版本: V1.10 日期: 2026-05-28

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

## 目录

简介 .....	3
Arduino Lib 函数 .....	3
Arduino Lib 下载及安装 .....	5
Arduino 范例 .....	6
范例 1: setID .....	6
范例 2: readTemperatureAndMoisture .....	8
范例 3: getData .....	10

## 简介

BM25S2621-1 是倍创推出的土壤温湿度检测模块，搭配 RS485 转 UART 后，使用 UART 通信方式。本文档对 BM25S2621-1 模块的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例使用 BME34K262 / BME34K262A 模块，演示了读取土壤温湿度等功能。

适用型号：

型号	说明
BM25S2621-1	土壤温湿度检测模块 ( 需 RS485 转 UART )
BME34K262	包括转接板 BMB22M301、BM25S2621-1
BME34K262A	包括转接板 BMB22M301A、BM25S2621-1

## Arduino Lib 函数

Arduino Lib 名称: BM25S2621-1		Lib 版本: V1.0.2
<b>构造函数 &amp; 初始化</b>		
1	<b>BM25S2621_1(uint8_t staPin, HardwareSerial* theSerial=&amp;Serial)</b>	
	描述	构造函数，选择硬件 UART 通信
	参数	staPin: RS485 收发器通信控制引脚 * theSerial: 选择硬件 UART 接口 ( 默认 Serial 接口 )
	返回值	—
	备注	staPin: 输入高电平为写模式 staPin: 输入低电平为读模式
2	<b>BM25S2621_1(uint8_t staPin, uint8_t rxPin, uint8_t txPin)</b>	
	描述	构造函数，选择软件 UART 通信
	参数	staPin: RS485 Transceiver 通信控制引脚 rxPin: RX 引脚，连接 BME34K262/BME34K262A 或 BM25S2621-1 转 UART 后的 TX 引脚 txPin: TX 引脚，连接 BME34K262/BME34K262A 或 BM25S2621-1 转 UART 后的 RX 引脚
	返回值	—
	备注	staPin: 输入高电平为写模式 staPin: 输入低电平为读模式
3	<b>void begin()</b>	
	描述	模块初始化
	参数	—
	返回值	void
	备注	—

功能函数	
4	uint8_t setID(uint8_t currentID, uint8_t newID, bool broadcast=false)
	描述 设置模块 ID
	参数 currentID: 模块当前 ID newID: 模块新 ID broadcast: 选择是否广播 TRUE: 广播 FALSE: 不广播 (默认)
	返回值 执行状态: 0 – 执行成功; 1 – 执行失败
	备注 —
5	uint8_t restoreDefault(uint8_t id, bool broadcast = false)
	描述 将模块恢复至出厂状态
	参数 id: 模块当前 ID broadcast: 选择是否广播 TRUE: 广播 FALSE: 不广播 (默认)
	返回值 执行状态: 0 – 执行成功; 1 – 执行失败
	备注 —
参数获取	
6	uint8_t getID()
	描述 读取模块 ID
	参数 —
	返回值 当前模块 ID
	备注 —
7	uint8_t readTemperature(uint8_t id, bool broadcast=false)
	描述 获取温度测量值
	参数 id: 模块当前 ID broadcast: 选择是否广播 TRUE: 广播 FALSE: 不广播 (默认)
	返回值 温度值, 单位 °C
	备注 —
8	uint8_t readMoisture(uint8_t id, broadcast=false)
	描述 获取湿度测量值
	参数 id: 模块当前 ID broadcast: 选择是否广播 TRUE: 广播 FALSE: 不广播 (默认)
	返回值 湿度测量值, 单位 %RH
	备注 —

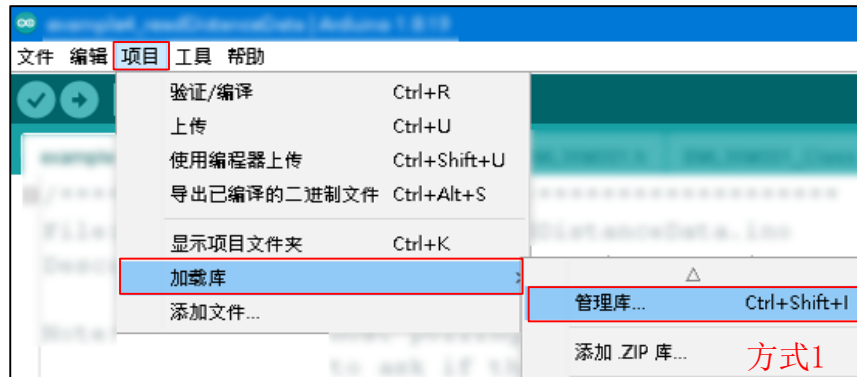
	uint8_t readTemperatureAndMoisture(uint8_t id, uint8_t *temp, uint8_t *mois, bool broadcast=false)	
	描述	获取温度、湿度测量值
9	参数	id: 模块当前 ID *temp: 用于存储获取的温度值, 单位 °C *mois: 用于存储获取的湿度值, 单位 %RH broadcast: 选择是否广播 TRUE: 广播 FALSE: 不广播 (默认)
	返回值	执行状态: 0 - 执行成功; 1 - 执行失败
	备注	—

## Arduino Lib 下载及安装

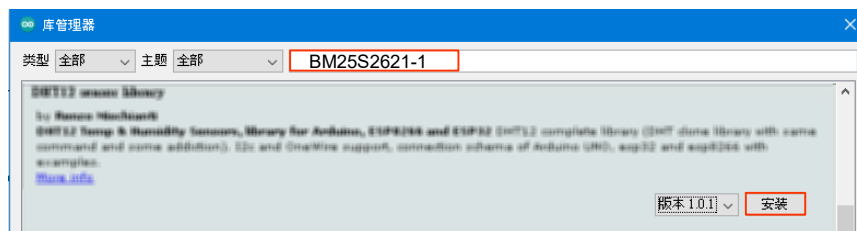
BM25S2621-1 Library: 可参考下面两种方法安装 BM25S2621-1 的 Arduino Library。

### 方式 1: 搜索安装

Arduino IDE → 项目 → 加载库 → 管理库 ... → 搜索 BM25S2621-1 → 安装



搜索安装流程 1

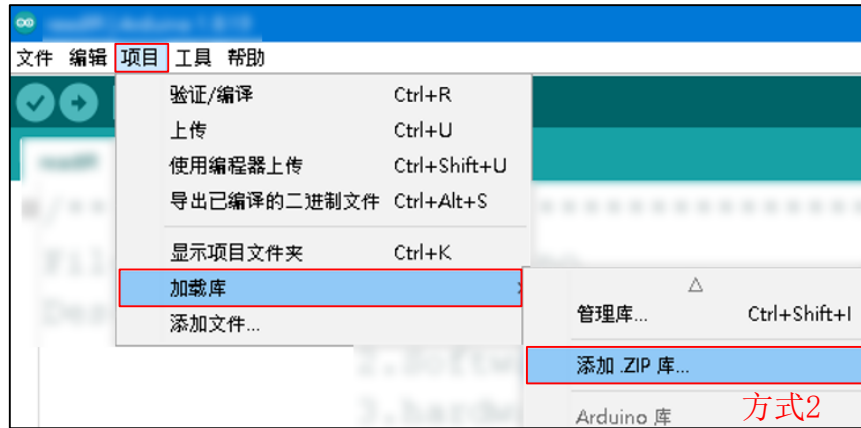


搜索安装流程 2

### 方式 2: 添加 .ZIP 库, 需提前下载 .ZIP 库

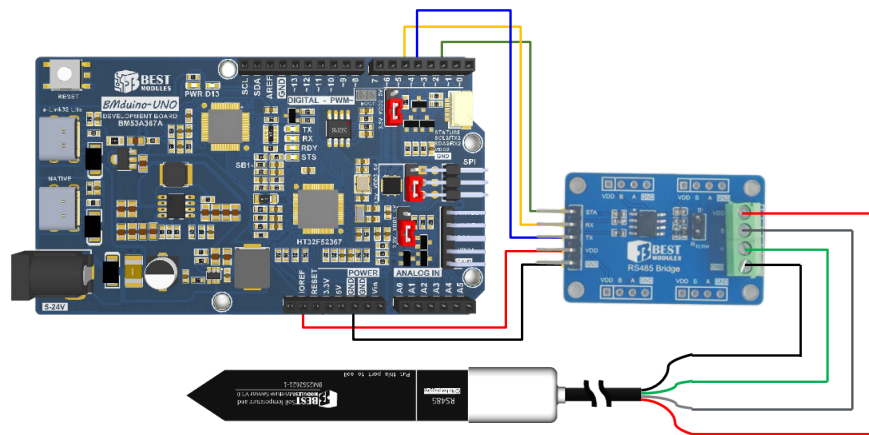
下载方法: 打开倍创官网 (<https://www.bestmodulescorp.com.cn/bm25s2621-1.html>), 下载“文件”菜单下的 Arduino 范例程序 (BM25S2621-1 Library)。

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

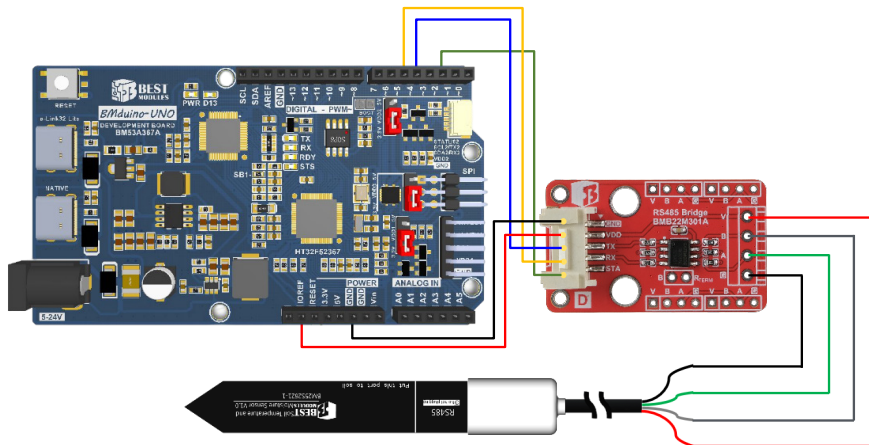


## Arduino 范例

### 范例 1: setID



BME34K262 实物连接示意图



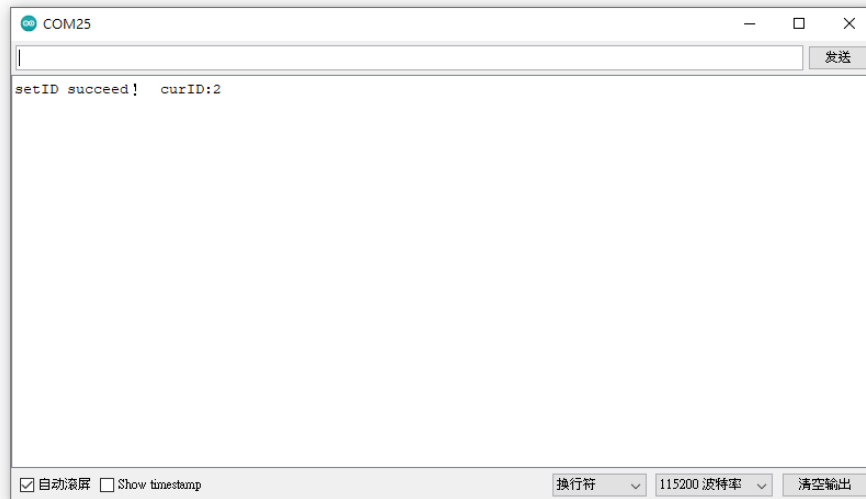
BME34K262A 实物连接示意图

范例实现功能：设置单个模块的 ID 值。

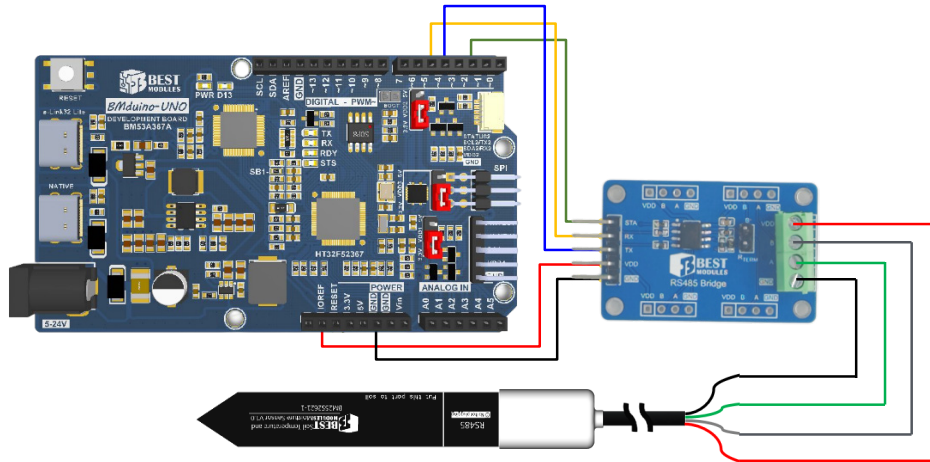
1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM25S2621-1) → 选择范例 (setID)
2. 示例说明：
  - a. 构建对象 & 初始化模块

```
#include "BM25S2621-1.h"
#define STATUS_PIN 2
#define RX_PIN 4
#define TX_PIN 5
BM25S2621_1 module (STATUS_PIN, RX_PIN, TX_PIN); // 创建对象,
// STA 引脚连接开发板 D2, 模块 TX 引脚连接开发板 D4, 模块 RX 引脚连接开发板 D5
Target_ID = 2; // 目标 ID
void setup()
{
  Serial.begin(115200);
  module.begin();
  if(module.setID(1,Target_ID,true)) // 广播命令,
  // 注意总线上每次只能接一个模块
  {
    Serial.print("setID succeed! curID:");
    Serial.println(Target_ID);
  }
  else
  {
    Serial.println("setID fail!");
  }
}
```

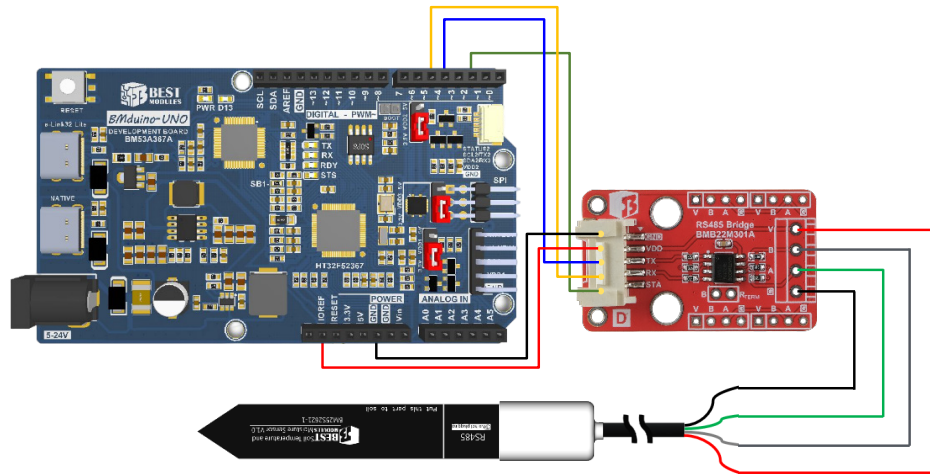
3. 打开串口监视器，波特率选择 115200；串口监视器显示数据如下：



## 范例 2: readTemperatureAndMoisture



BME34K262 实物连接示意图



BME34K262A 实物连接示意图

范例实现功能：每隔 1 秒获取一次模块的温湿度测量值，并在串口监视器上显示。

1. 范例打开：Arduino IDE → 文件 → 示例 → Lib 选择 (BM25S2621-1) → 选择范例 (readTemperatureAndHumidity)
2. 示例说明：
  - a. 构建对象 & 初始化模块

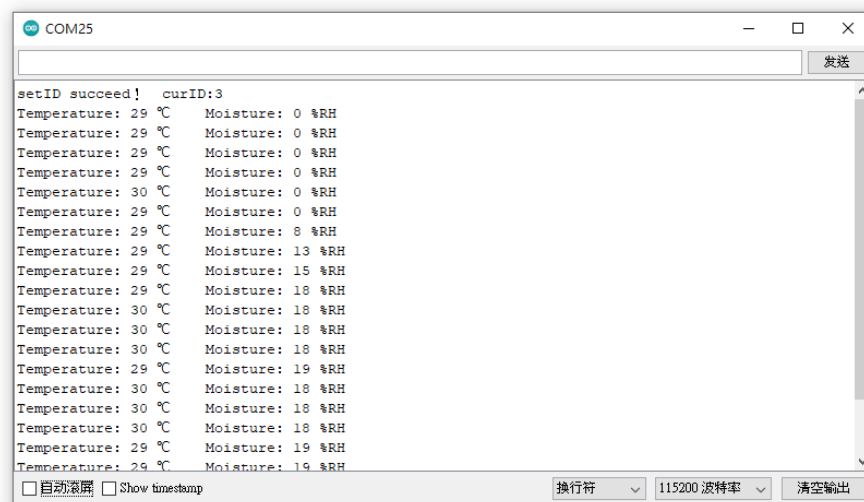
```
#include "BM25S2621-1.h"
#define STATUS_PIN 2
#define RX_PIN 4
#define TX_PIN 5
BM25S2621_1 module (STATUS_PIN, RX_PIN, TX_PIN); // 创建对象,
// STA 引脚连接开发板 D2, 模块 TX 引脚连接开发板 D4, 模块 RX 引脚连接开发板 D5
uint8_t Target_ID = 3; // 目标 ID
uint8_t mois = 0; // 储存湿度值
uint8_t temp = 0; // 储存温度值
void setup()
```

```
{  
  Serial.begin(115200);  
  module.begin();  
  // 通过广播命令设置模块 ID 为指定 ID  
  if(module.setID(1,Target_ID,true))  
  {  
    Serial.print("setID succeed! curID:");  
    Serial.println(Target_ID);  
  }  
  else  
  {  
    Serial.println("setID fail!");  
  }  
}
```

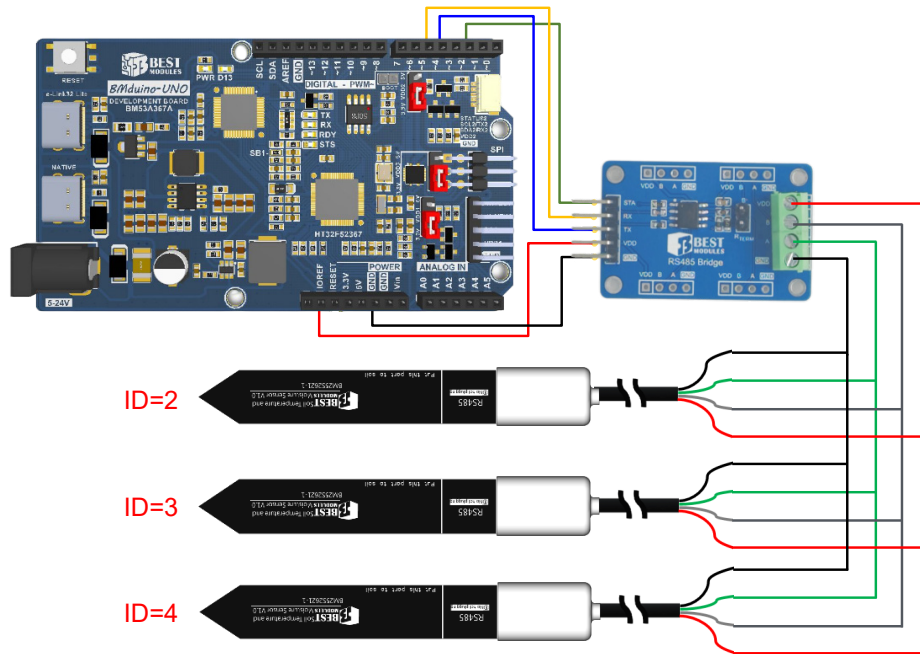
b. 每隔一秒获取温湿度测量值，并在串口监视器中显示

```
void loop()  
{  
  if( module.readTemperatureAndMoisture(Target_ID,&temp,&mois,false))  
  // 获取温湿度  
  {  
    Serial.print("Temperature:");  
    Serial.print(temp);  
    Serial.print("°C");  
    Serial.print("Moisture:");  
    Serial.print(mois);  
    Serial.println("%RH");  
  }  
  else  
  {  
    Serial.println("read temperature and moisture fail!");  
  }  
  delay(1000);  
}
```

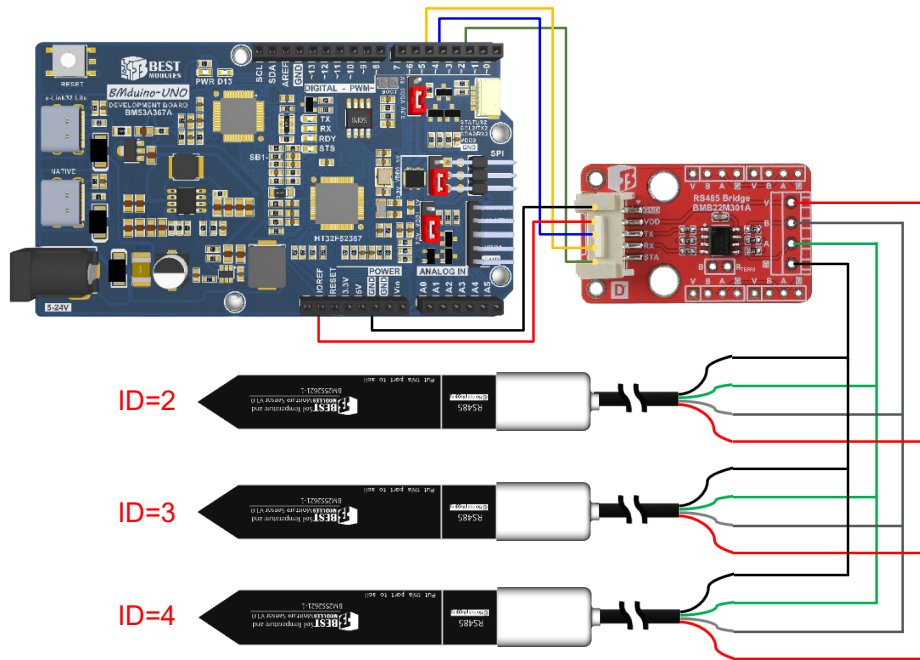
3. 打开串口监视器，波特率选择 115200；将探头插入土壤中，串口监视器显示数据如下：



### 范例 3: getData



BME34K262 实物连接示意图



BME34K262A 实物连接示意图

范例实现功能：将 3 个模块设置为指定 ID 并获取模块的温湿度测量值，并在串口监视器上显示。

注：先用 setID 范例，将 3 个模块 ID 分别设置为 2、3、4，再使用本范例。

1. 范例打开: Arduino IDE → 文件 → 示例 → Lib 选择 (BM25S2621-1) → 选择范例 (getData)

2. 示例说明:

a. 构建对象 & 初始化模块

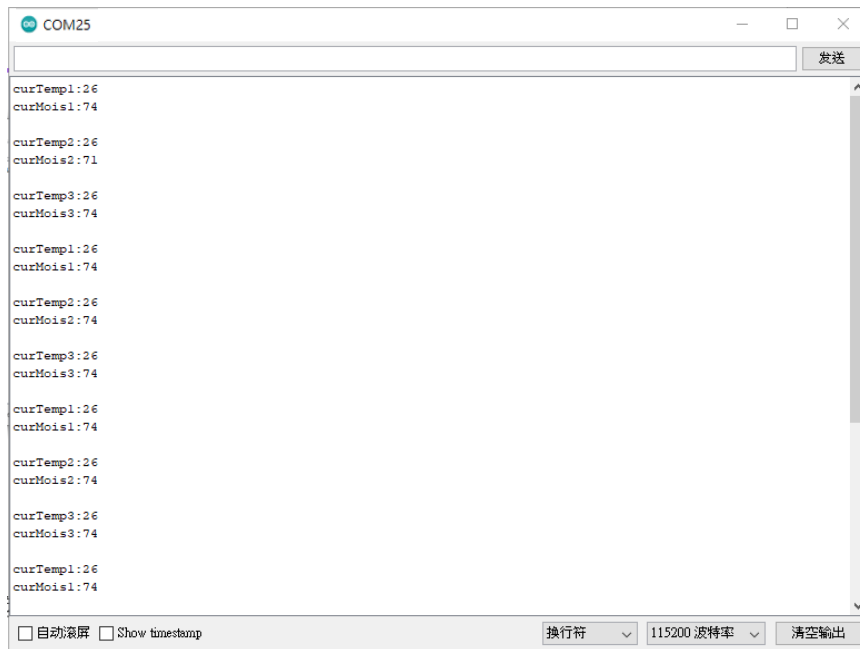
```
#include "BM25S2621-1.h"
#define STATUS_PIN 2
#define RX_PIN 4
#define TX_PIN 5
BM25S2621_1 module (STATUS_PIN, RX_PIN, TX_PIN); // 创建对象,
// STA 引脚连接开发板 D2, 模块 TX 引脚连接开发板 D4, 模块 RX 引脚连接开发板 D5
uint8_t mois1, mois2, mois3; // Soil moisture value
uint8_t temp1, temp2, temp3; // Soil temperature value
uint8_t module1_id = 2; // Module 1 ID
uint8_t module2_id = 3; // Module 2 ID
uint8_t module3_id = 4; // Module 3 ID
void setup()
{
  Serial.begin(115200);
  module.begin();
}
```

b. 每隔一秒获取全部模块温湿度测量值, 并在串口监视器中显示

```
void loop()
{
  if(module.readTemperatureAndMoisture(module1_id,&temp1,&mois1,
  false))// 获取第一个模块的土壤温湿度
  {
    Serial.print("curTemp1:");
    Serial.println(temp1);
    Serial.print("curMois1:");
    Serial.println(mois1);
  }
  else
  {
    Serial.println("module1 getTemperature_Moisture error");
  }
  delay(10);
  if(module.readTemperatureAndMoisture (module2_id,&temp2,&mois2,
  false))// 获取第二个模块的土壤温湿度
  {
    Serial.print("curTemp2:");
    Serial.println(temp2);
    Serial.print("curMois2:");
    Serial.println(mois2);
  }
  else
  {
    Serial.println("module2 getTemperature_Moisture error");
  }
  delay(10);
}
```

```
if(module. readTemperatureAndMoisture (module3_id,&temp3,&mois3,
false))// 获取第三个模块的土壤温湿度
{
    Serial.print("curTemp3:");
    Serial.println(temp3);
    Serial.print("curMois3:");
    Serial.println(mois3);
}
else
{
    Serial.println("module3 getTemperature_Moisture error");
}
Serial.println();
delay(1000);
}
```

3. 打开串口监视器，波特率选择 115200；将探头插入土壤中，串口监视器显示数据如下：



Copyright® 2026 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。