



感煙探測數位傳感器

BM22S2021-1 Arduino Library 說明

版本：V1.00 日期：2023-05-30

www.bestmodulescorp.com

目錄

簡介	3
Arduino Lib 函式	3
Arduino Lib 下載及安裝	8
Arduino 範例	9
範例 1: readStatusPin	9
範例 2: readSmokeValue	11
範例 3: airCalibration	12

簡介

BM22S2021-1 是倍創推出的感煙探測數位傳感器。使用 UART 通訊方式。本文檔對 BM22S2021-1 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例使用 BMA26M202 模組，演示了報警檢測、煙霧值讀取、空氣校準等功能。

適用型號：

型號	說明
BM22S2021-1	感煙探測數位傳感器
BMA26M202	板載 BM22S2021-1 傳感器

Arduino Lib 函式

Arduino Lib 名稱：BM22S2021-1		Lib 版本：V1.0.1
構造成函式 & 初始化		
1	BM22S2021_1(uint8_t statusPin,HardwareSerial*theSerial)	
	描述	構造成函式 硬體 UART
	參數	statusPin：STATUS 腳位，連接 BM22S2021-1 的 STATUS 腳位 或 BMA26M202 的 STA 腳位 * theSerial：硬體 UART 通訊介面選擇
	返回值	—
	備註	—
2	BM22S2021_1(uint8_t statusPin,uint8_t rxPin, uint8_t txPin)	
	描述	構造成函式 軟體 UART
	參數	statusPin：STATUS 腳位，連接 BM22S2021-1 的 STATUS 腳位 或 BMA26M202 的 STA 腳位 rxPin：RX 腳位，連接 BM22S2021-1 或 BMA26M202 的 TX 腳位 txPin：TX 腳位，連接 BM22S2021-1 或 BMA26M202 的 RX 腳位
	返回值	—
	備註	—
3	void begin()	
	描述	模組初始化
	參數	—
	返回值	void
	備註	波特率固定為 9600
功能函式		
4	uint16_t getSTATUS()	
	描述	獲取 STATUS 腳位準位
	參數	—
	返回值	STATUS 腳位準位： 0：低準位 1：高準位
	備註	—

5	uint16_t getFWVer()	
	描述	獲取模組軟體版本
	參數	—
	返回值	模組軟體版本 FWVer
	備註	—
6	uint8_t getProDate(uint8_t buff[])	
	描述	獲取模組生產日期
	參數	buff[]：用於存儲生產日期，buff[0]/buff[1]/buff[2]：年/月/日
	返回值	執行情況： 0：成功 1：失敗
	備註	如：0x21,0x06,0x2 表示 2021/6/2
7	uint8_t requestInfoPackage(uint8_t buff[])	
	描述	獲取設備當前所有數據
	參數	buff[]：存放讀取到的數據包 (41 bytes)
	返回值	執行情況： 0：成功 1：失敗
	備註	主動發送指令獲取數據，數據內容參考 BM22S2021-1 規格書
8	bool isInfoAvailable()	
	描述	判斷是否是模組自動輸出的數據包
	參數	—
	返回值	判斷結果： true：是 false：不是
	備註	—
9	void readInfoPackage(uint8_t array[])	
	描述	讀取 TX 腳位自動輸出的字節數據包
	參數	array[]：存放讀取到的數據包
	返回值	—
	備註	數據包大小：41 字節 /21 字節，數據內容參考 BM22S2021-1 規格書
10	uint8_t readRegister(uint8_t addr)	
	描述	讀取指定暫存器的數據
	參數	addr：暫存器位址
	返回值	暫存器數據
	備註	暫存器內容參考 BM22S2021-1 規格書
11	uint8_t readRunningVariables(uint8_t addr)	
	描述	讀取指定位址的運行變數
	參數	addr：地址
	返回值	運行變數
	備註	運行變數，請參考 BM22S2021-1 規格書的運行變數清單

12	uint8_t getAutoTx()	
	描述	獲取 TX 腳位自動輸出的使能模式
	參數	—
	返回值	使能模式： 0x80：詳細輸出 0x81：簡單輸出 0x00：不輸出
	備註	—
13	uint8_t getStatusPinActiveMode()	
	描述	獲取報警時 STATUS 腳位的輸出準位
	參數	—
	返回值	報警準位： 0x80：高準位 0x00：低準位
	備註	—
14	uint8_t getT0ATopLimit()	
	描述	獲取 T0A 空氣標定上限參數
	參數	—
	返回值	T0A 空氣標定上限參數
	備註	煙霧值上限 = 標定上限參數 × 16。
15	uint8_t getT0ABottomLimit()	
	描述	獲取 T0A 空氣標定下限參數
	參數	—
	返回值	T0A 空氣標定下限參數
	備註	煙霧值下限 = 標定下限參數 × 16。
16	uint8_t getT0BTopLimit()	
	描述	獲取 T0B 空氣標定上限參數
	參數	—
	返回值	T0B 空氣標定上限參數
	備註	煙霧值上限 = 標定上限參數 × 16。
17	uint8_t getT0BBottomLimit()	
	描述	獲取 T0B 空氣標定下限參數
	參數	—
	返回值	T0B 空氣標定下限參數
	備註	煙霧值下限 = 標定下限參數 × 16。
18	uint16_t getT0AThreshold()	
	描述	獲取 T0A 報警閾值
	參數	—
	返回值	T0A 報警閾值
	備註	12-bit 有效數據

19	uint16_t getT0BThreshold()	
	描述	獲取 T0B 報警閾值
	參數	—
	返回值	T0B 報警閾值
	備註	12-bit 有效數據
20	uint8_t getDetectCycle()	
	描述	獲取設備待機時的檢測週期
	參數	—
	返回值	檢測週期 · 單位 s
	備註	無煙環境下設備待機時的檢測週期
21	uint8_t calibrateModule()	
	描述	進行空氣標定
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	函式運行至結束需要 8s
22	uint8_t resetModule()	
	描述	復位模組
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	—
23	uint8_t restoreDefault ()	
	描述	使模組恢復出廠設置
	參數	—
	返回值	執行情況： 0：成功 1：失敗
	備註	所有參數恢復默認
配置參數		
24	uint8_t writeRegister(uint8_t addr,uint8_t data)	
	描述	向指定暫存器寫入數據
	參數	addr：地址 data：需要寫入的數據
	返回值	執行情況： 0：成功 1：失敗
	備註	—

25	uint8_t setAutoTx(uint8_t mode)	
	描述	設置 TX 腳位自動輸出的使能模式
	參數	mode：使能模式： 0x80：詳細輸出 0x81：簡易輸出 0x00：不輸出
	返回值	執行情況： 0：成功 1：失敗
	備註	詳細輸出數據包為 41 字節；簡單輸出數據包為 21 字節
26	uint8_t setStatusPinActiveMode(uint8_t state)	
	描述	設置報警時 STATUS 腳位輸出準位
	參數	報警準位： 0x80：高準位 0x00：低準位
	返回值	執行情況： 0：成功 1：失敗
	備註	默認 0x80
27	uint8_t setT0ACalibrateRange(uint8_t toplimit,uint8_t bottomlimit)	
	描述	設置 T0A 標定上下限參數
	參數	toplimit：標定上限參數 bottomlimit：標定下限參數
	返回值	執行情況： 0：成功 1：失敗
	備註	toplimit 默認 0xC8，bottomlimit 默認 0x19 煙霧值上限 = toplimit×16，煙霧值下限 = bottomlimit×16。 空氣標定（當前環境）的煙霧值需要在標定上下限內才能成功。
28	uint8_t setT0BcalibrateRange (uint8_t toplimit,uint8_t bottomlimit)	
	描述	設置 T0B 標定上下限參數
	參數	toplimit：標定上限參數 bottomlimit：標定下限參數
	返回值	執行情況： 0：成功 1：失敗
	備註	toplimit 默認 0xC8，bottomlimit 默認 0x19 煙霧值上限 = toplimit×16，煙霧值下限 = bottomlimit×16。 空氣標定（當前環境）的煙霧值需要在標定上下限內才能成功。
29	uint8_t setT0AThreshold(uint16_t value)	
	描述	設置 T0A 報警閾值
	參數	value：報警閾值，默認為 0x015e
	返回值	執行情況： 0：成功 1：失敗
	備註	12-bit 有效數據

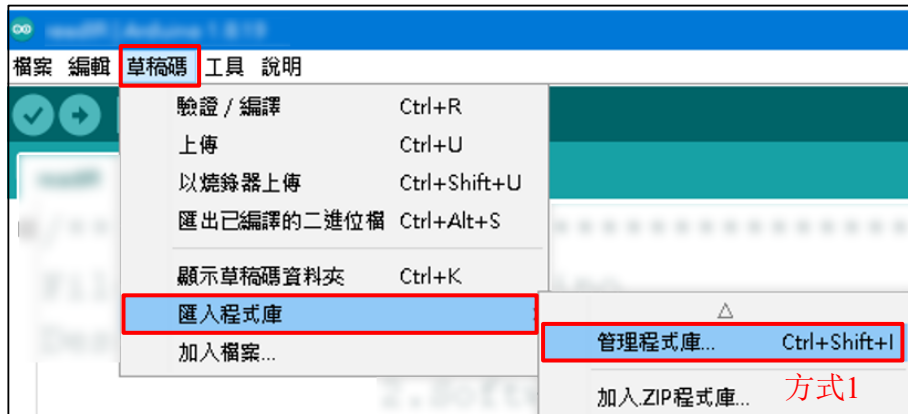
30	uint8_t setT0BThreshold(uint16_t value)	
	描述	設置 T0B 報警閾值
	參數	value：報警閾值，默認 0x015e
	返回值	執行情況： 0：成功 1：失敗
	備註	12-bit 有效數據
31	uint8_t setDetecteCycle(uint8_t Cycle)	
	描述	設置設備待機時的檢測週期
	參數	Cycle：檢測週期，預設 0x08，單位 s
	返回值	執行情況： 0：成功 1：失敗
	備註	無煙環境下設備待機時的檢測週期

Arduino Lib 下載及安裝

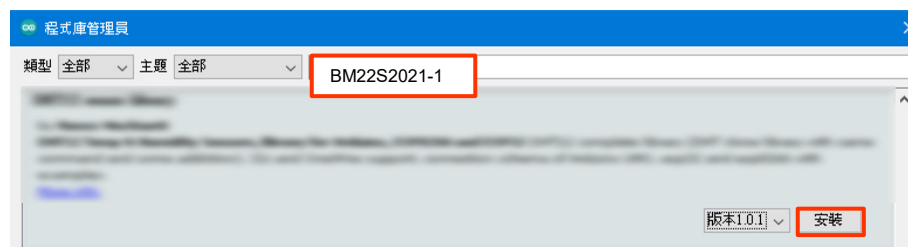
BM22S2021-1 Library：可參考下面兩種方法安裝 BM22S2021-1 的 Arduino Library

方式 1：搜索安裝

搜索安裝：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BM22S2021-1 → 安裝



搜索安裝流程 1

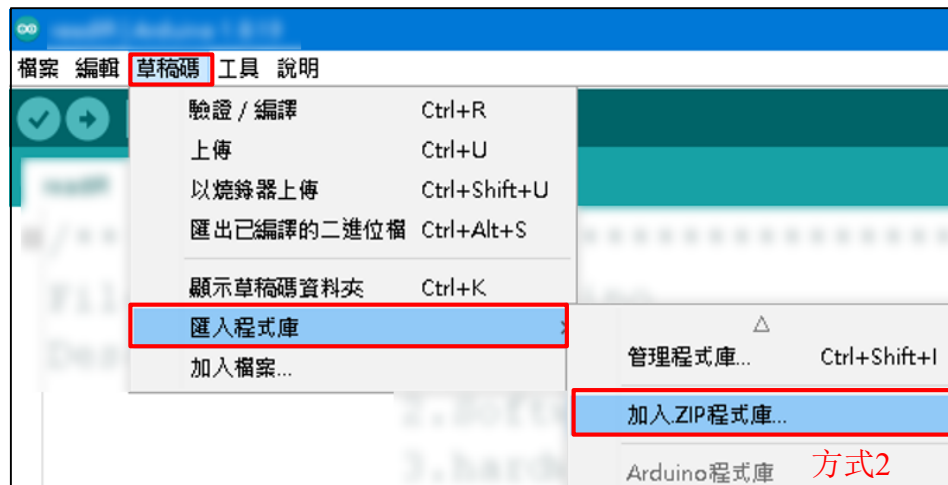


搜索安裝流程 2

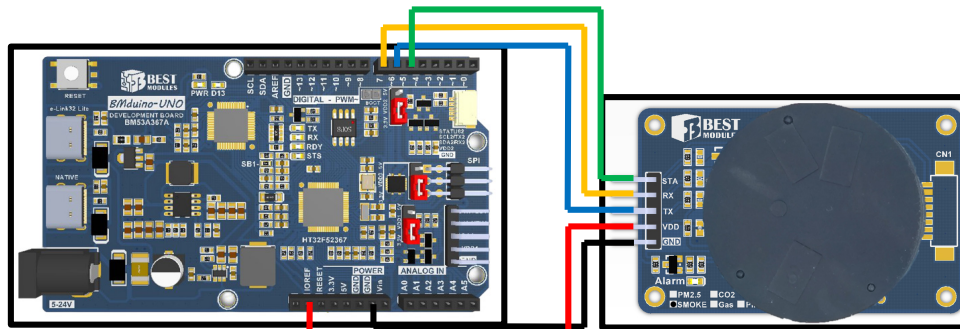
方式 2：添加 .ZIP 程式庫 · 需提前下載 .ZIP 程式庫

下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bm22s2021-1.html#tab-product2>) 文件目錄下的 Arduino 範例程式 (BM22S2021-1 Library)。

添加 .ZIP 程式庫：Arduino IDE → 草稿碼 → 匯入程式庫 → 加入 .ZIP 程式庫 ...



Arduino 範例



實物連接示意圖

範例 1: readStatusPin

範例實現功能：模組正常狀態下 LED13 緩慢閃爍，檢測到模組報警則 LED13 常亮。

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM22S2021-1) → 選擇範例 (readStatusPin)

2. 範例說明：

a. 創建對象

```
#include <Arduino.h>
#include "BM22S2021-1.h"
uint8_t STATUS=5;           // 選擇 statusPin
uint8_t flag=3;
BM22S2021_1 SMOKE(STATUS,6,7); // software Serial
```

b. 模組初始化 配置 I/O

```
void setup()
{
  Serial.begin(9600);
  SMOKE.begin();           // 模組初始化 波特率為 9600
  pinMode(STATUS, INPUT);
  pinMode(13, OUTPUT);
  delay(8000);
}
```

c. 迴圈檢測模組狀態 · 正常 / 報警 LED 閃亮狀態不同

```
void loop()
{
  if(flag!=0&&SMOKE.getSTATUS() != HIGH)
  {
    flag=0;
    Serial.println("Module normal;No alarm"); // 觸發報警狀態
  }
  if(SMOKE.getSTATUS() ==HIGH&&flag!=1)
  {
    Serial.println("Alarm! Smoke concentration too high"); // 模組當前
                                                            // 正常
    flag=1;
  }
  switch(flag)
  {
    case 0: // 正常
      digitalWrite(13, HIGH);
      delay(100);
      digitalWrite(13, LOW);
      delay(7900);
      break;
    case 1: // 報警
      digitalWrite(13, HIGH);
      delay(8000);
      break;
  }
}
```

範例 2: readSmokeValue

範例實現功能：獲取模組自動輸出的數據包，將數據包內的煙霧值列印至串口監視器。

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM22S2021-1) → 選擇範例 (readSmokeValue)
2. 範例說明：
 - a. 構建對象

```
#include <Arduino.h>
#include "BM22S2021-1.h"
uint8_t STATUS=5; // 選擇 statusPIN
uint16_t smokeA;
uint16_t smokeB;
uint8_t Data[41];
BM22S2021-1 SMOKE (STATUS,6,7); // software Serial
```

- b. 初始化模組

```
void setup()
{
  Serial.begin(9600);
  SMOKE.begin(); // 模組初始化 波特率為 9600
  pinMode(STATUS, INPUT);
  pinMode(13, OUTPUT);
}
```

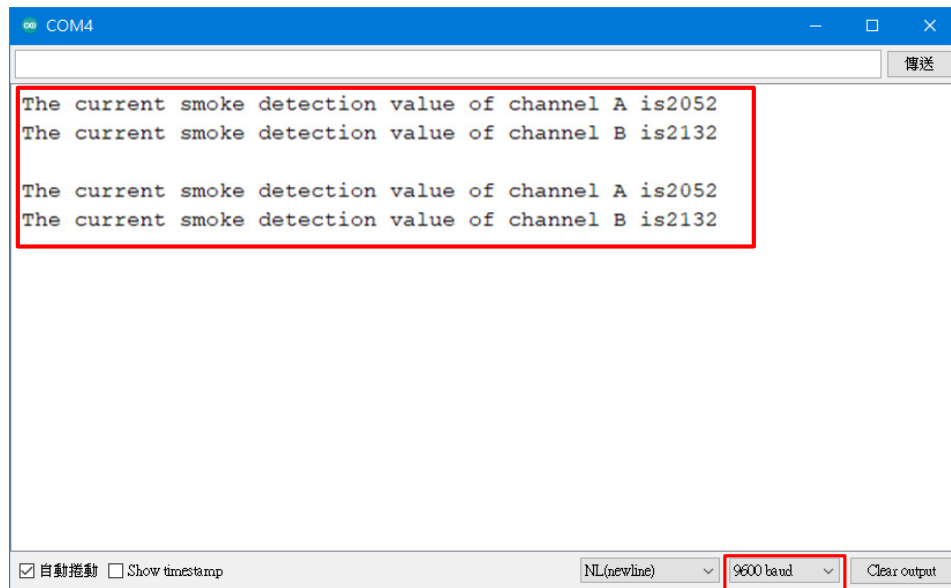
- c. 輪詢判斷是否有數據包，並列印煙霧值。

```
void loop()
{
  if(SMOKE.isInfoAvailable()==1) // 判斷是否有數據包
  {
    SMOKE.readInfoPackage(Data); // 接收模組發送的數據包
    printInfo(); // 列印煙霧檢測值函式
  }
}

void printInfo()
{
  /* 列印通道 A 的煙霧檢測值 */
  smokeA=(Data[17]<<8 | Data[16]);
  Serial.print("The current smoke detection value of channel A is ");
  Serial.println(smokeA,DEC);

  /* 列印通道 B 的煙霧檢測值 */
  smokeB=(Data[19]<<8 | Data[18]);
  Serial.print("The current smoke detection value of channel B is ");
  Serial.println(smokeB,DEC);
  Serial.println();
}
```

3. 打開串口監視器，波特率選擇 9600；串口監視器顯示如下



範例 3: airCalibration

範例實現功能：關閉自動輸出模式，進行空氣標定。

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM22S2021-1) → 選擇範例 (airCalibration)
2. 範例說明：
 - a. 構建對象

```
#include <Arduino.h>
#include "BM22S2021-1.h"
uint8_t i;
uint8_t STATUS=5;
BM22S2021_1 SMOKE(STATUS,6,7); // software Serial
```

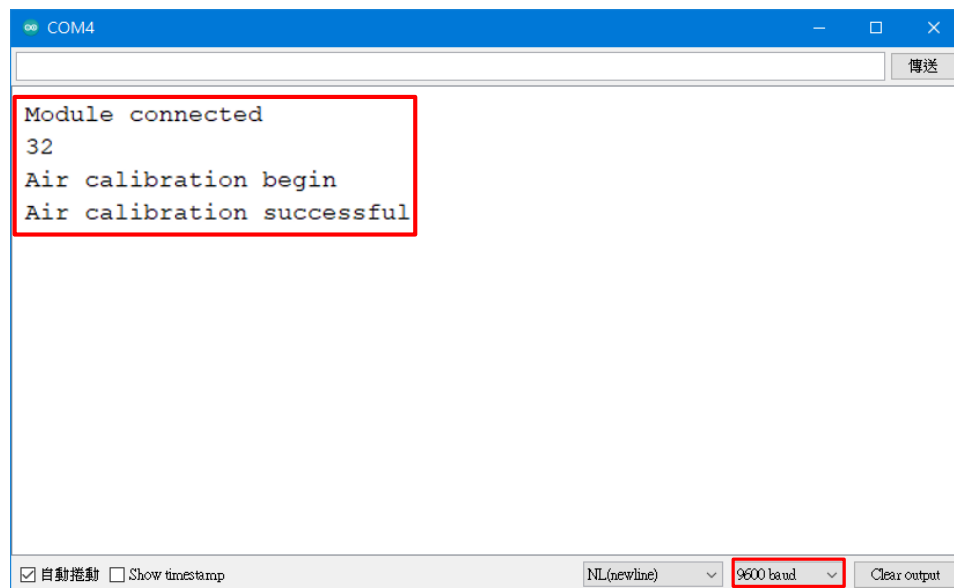
- b. 初始化模組，關閉自動輸出

```
void setup()
{
  Serial.begin(9600); // 串口監視器波特率 9600
  SMOKE.begin();
  i=SMOKE.setAutoTx(0x00); // 關閉 TX 自動輸出 不影響標定
  if (i==0) // 驗證是否設置成功
  {
    Serial.println("Module connected");
  }
}
```

c. 輪詢串口數據 · 執行對應動作。

```
void loop()
{
  if(Serial.available() > 0) // 獲取串口監視器發送的數據
  {
    i=Serial.read();
    Serial.println(i,HEX);
  }
  if(i==0x32) // 發送 2 進行空氣標定
  {
    Serial.println( "Air calibration begin" );
    i=SMOKE.calibrateModule();//Wait for 8s
    if(i==0)
    {
      Serial.println( "Air calibration successful" );
    }
    else
    {
      Serial.println( "Air calibration failed" );
    }
  }
  i=0;
}
```

3. 打開串口監視器 · 波特率選擇 9600；串口監視器顯示如下



Copyright® 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版時倍創已針對所載資訊為合理注意，但不保證資訊準確無誤。文中提到的資訊僅是提供作為參考，且可能被更新取代。倍創不擔保任何明示、默示或法定的，包括但不限於適合商品化、令人滿意的品質、規格、特性、功能與特定用途、不侵害第三人權利等保證責任。倍創就文中提到的資訊及該資訊之應用，不承擔任何法律責任。此外，倍創並不推薦將倍創的產品使用在會因故障或其他原因而可能會對人身安全造成危害的地方。倍創特此聲明，不授權將產品使用於救生、維生或安全關鍵零組件。在救生 / 維生或安全應用中使用倍創產品的風險完全由買方承擔，如因該等使用導致倍創遭受損害、索賠、訴訟或產生費用，買方同意出面進行辯護、賠償並使倍創免受損害。倍創 (及其授權方，如適用) 擁有本文件所提供資訊 (包括但不限於內容、資料、示例、材料、圖形、商標) 的智慧財產權，且該資訊受著作權法和其他智慧財產權法的保護。倍創在此並未明示或暗示授予任何智慧財產權。倍創擁有不事先通知而修改本文件所載資訊的權利。如欲取得最新的資訊，請與我們聯繫。