



CO 探測數位傳感器

# BM22S3221-1 Arduino Library 說明

版本：V1.00 日期：2023-08-18

[www.bestmodulescorp.com](http://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：readCOValue .....	11

## 簡介

BM22S3221-1 是倍創推出的 CO 探測數位傳感器，使用 UART 通訊方式。本文檔對 BM22S3221-1 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明。範例使用 BMA56M322 模組，演示了報警信號輸出、讀取 CO 濃度的功能。

適用型號：

型號	說明
BM22S3221-1	CO 探測數位傳感器
BMA56M322	板載 BM22S3221-1 傳感器

## Arduino Lib 函式

Arduino Lib 名稱：BM22S3221-1		Lib 版本：V1.0.1
<b>構造成函式 &amp; 初始化</b>		
1	BM22S3221_1(uint8_t statusPin, HardwareSerial *theSerial = &Serial)	
	描述	構造成函式 (硬體 UART)
	參數	statusPin：STATUS 腳位，連接 BM22S3221-1 或 BMA56M322 的 STATUS 腳位 *theSerial：硬體介面
	返回值	—
	備註	—
2	BM22S3221_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 BM22S3221-1 的 STATUS 腳位 或 BMA56M322 的 STA 腳位 rxPin：RX 腳位，連接 BM22S3221-1 或 BMA56M322 的 TX 腳位 txPin：TX 腳位，連接 BM22S3221-1 或 BMA56M322 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	—
4	void preheatCountdown()	
	描述	等待完成模組預熱
	參數	—
	返回值	void
	備註	1. 此函式運行結束，代表預熱完成； 2. 當模組上電時，此函式判斷其自動輸出功能是否使能 a. 使能：將一直讀取自動輸出的資訊包中抓取預熱剩餘時間，當該時間為零時退出函式； b. 除能：延時約 120s 後退出函式

功能函式		
5	uint8_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	—
6	uint8_t getWorkStatus()	
	描述	獲取模組當前的工作狀態
	參數	—
	返回值	設備狀態 (1 字節)： Bit 0 為 1：表示標定中 Bit 1~2：保留 Bit 3 為 1：表示標定動作完成 Bit 4 為 1：表示零點標定完成 Bit 5 為 1：表示帶氣標定完成 Bit 6 為 1：表示傳感器故障狀態 Bit 7 為 1：表示報警狀態
	備註	—
7	uint16_t readCOValue()	
	描述	讀取 CO 濃度值
	參數	—
	返回值	CO 濃度值 · 單位：ppm
	備註	最大有效值 10000ppm
8	uint16_t readADValue()	
	描述	讀取當前 CO 濃度對應的 A/D 值
	參數	—
	返回值	12-bit A/D 值
	備註	—
9	uint16_t readRefValue()	
	描述	讀取上電基準值
	參數	—
	返回值	12-bit A/D 值
	備註	—
10	uint8_t requestInfoPackage(uint8_t array[])	
	描述	獲取模組的所有資訊
	參數	array[]：用於存儲模組資訊 (34 字節)
	返回值	執行情況 <sup>(1)</sup>
	備註	1. 建議在除能串口自動輸出功能後使用 2. 每個字節的含義請見 Datasheet

11	<b>bool isInfoAvailable()</b>	
	描述	查詢是否接收到模組自動輸出的資訊
	參數	—
	返回值	接收到模組自動輸出的資訊情況 true：已收到 false：未收到
	備註	在使能串口自動輸出功能後使用
12	<b>void readInfoPackage(uint8_t array[])</b>	
	描述	讀取模組自動輸出的資訊
	參數	array[]：用於存儲模組資訊 ( 34 字節 )
	返回值	—
	備註	1. 此函式在 "isInfoAvailable()==true" 後使用 2. 每個字節的含義請見 Datasheet
13	<b>uint8_t resetModule()</b>	
	描述	復位模組
	參數	—
	返回值	執行情況 <sup>(1)</sup>
	備註	此函式執行後模組進入預熱狀態
14	<b>uint8_t restoreDefault()</b>	
	描述	將模組的參數恢復至出廠設置
	參數	—
	返回值	執行情況 <sup>(1)</sup>
	備註	此函式執行後模組進入預熱狀態
<b>參數查詢</b>		
15	<b>uint16_t getFWVer()</b>	
	描述	查詢韌體版本
	參數	—
	返回值	韌體版本號
	備註	0x0105 表示 V1.5
16	<b>uint8_t getProDate(uint8_t array[])</b>	
	描述	查詢生產日期
	參數	array[]：用於存儲生產日期 array[0]/array[1]/array[2]：年 / 月 / 日
	返回值	0：執行成功 1：數據校驗錯誤 2：通訊超時
	備註	如：0x21, 0x06, 0x2 表示 2021/6/2
17	<b>bool isAutoTx()</b>	
	描述	查詢串口自動輸出功能是否使能
	參數	—
	返回值	串口自動輸出功能情況 true：使能自動輸出 false：除能自動輸出
	備註	—

18	uint8_t getStatusPinActiveMode()	
	描述	查詢模組報警時，STATUS 腳位輸出高準位或低準位
	參數	—
	返回值	報警準位： 0x08：輸出高準位 0x00：輸出低準位
	備註	—
19	uint16_t getAlarmThreshold()	
	描述	查詢報警的 CO 濃度閾值
	參數	—
	返回值	CO 報警閾值，單位：ppm
	備註	—
20	uint16_t getExitAlarmThreshold()	
	描述	查詢退出報警的 CO 濃度閾值
	參數	—
	返回值	退出 CO 報警的閾值，單位：ppm
	備註	—
<b>參數設置</b>		
21	uint8_t setAutoTx(uint8_t autoTx= AUTO)	
	描述	設置模組是否通過 TX 腳位自動輸出所有的模組數據
	參數	autoTx：自動輸出情況 0x08 (AUTO)：自動輸出 ( 默認 ) 0x00 (PASSIVE)：不自動輸出
	返回值	執行情況
	備註	1. 資訊輸出週期為 1 秒 2. 在資訊發送期間 ( 約 40ms )，模組無法接收任何指令
22	uint8_t setStatusPinActiveMode(uint8_t statusMode = HIGH_LEVEL)	
	描述	設置模組報警時，STATUS 腳位的輸出準位
	參數	statusMode：STATUS 腳位的輸出準位情況 0x08 (HIGH_LEVEL)：高準位 ( 默認 ) 0x00 (LOW_LEVEL)：低準位
	返回值	執行情況 <sup>(1)</sup>
	備註	—
23	uint8_t setAlarmThreshold(uint16_t alarmThreshold = 180)	
	描述	設置 CO 報警的濃度值
	參數	alarmThreshold：CO 報警的濃度值 ( 默認 180ppm )，單位：ppm
	返回值	執行情況 <sup>(1)</sup>
	備註	—
24	uint8_t setExitAlarmThreshold(uint8_t exitAlarmThreshold = 55)	
	描述	設置退出 CO 報警的濃度值
	參數	exitAlarmThreshold：CO 報警的濃度值 ( 默認 55 )，單位：ppm
	返回值	執行情況 <sup>(1)</sup>
	備註	該值需小於 CO 報警的濃度值

25	uint8_t calibrateModule()	
	描述	啟動帶氣標定功能
	參數	—
	返回值	執行情況 <sup>(1)</sup>
	備註	<ol style="list-style-type: none"> <li>1. 標定期間不可向模組發送指令</li> <li>2. 標定需在標準氣體下進行，詳見 Datasheet</li> <li>3. 啟動後需等待模組標定完成，可使用 calibrateCountdown()</li> </ol>
26	void calibrateCountdown()	
	描述	等待模組標定完成
	參數	—
	返回值	—
	備註	<ol style="list-style-type: none"> <li>1. 此函式運行結束，代表標定完成；</li> <li>2. 當模組上電時，此函式判斷其自動輸出功能是否使能 <ol style="list-style-type: none"> <li>a. 使能：將一直讀取自動輸出的資訊包中抓取標定剩餘時間，當該時間為零時退出函式；</li> <li>b. 除能：延時約 120s 後退出函式</li> </ol> </li> </ol>

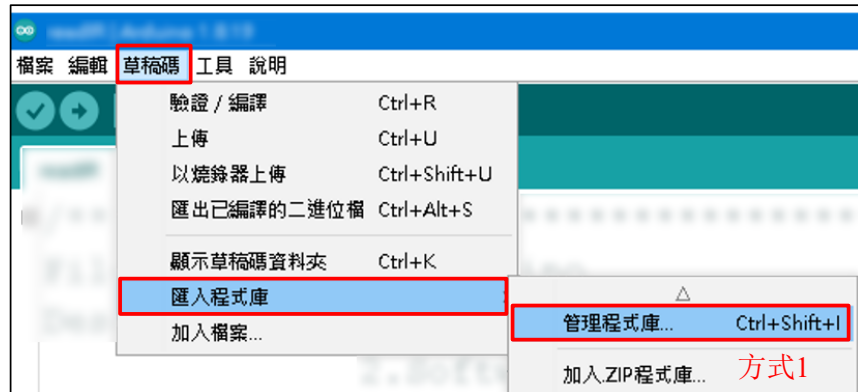
註 1：0 – 指令執行成功；1 – 模組應答的數據校驗錯誤；2 – 通訊超時

## Arduino Lib 下載及安裝

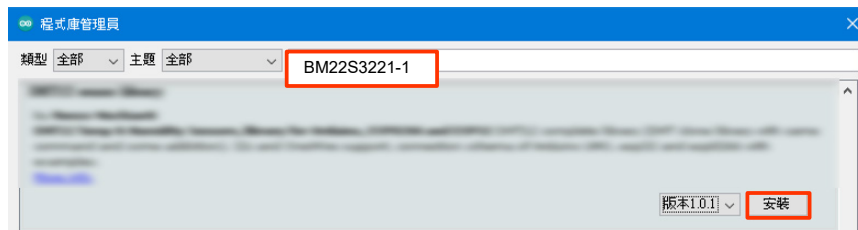
BM22S3221-1 Library：可參考下面兩種方法安裝 BM22S3221-1 的 Arduino Library。

### 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 BM22S3221-1 → 安裝



搜索安裝流程 1

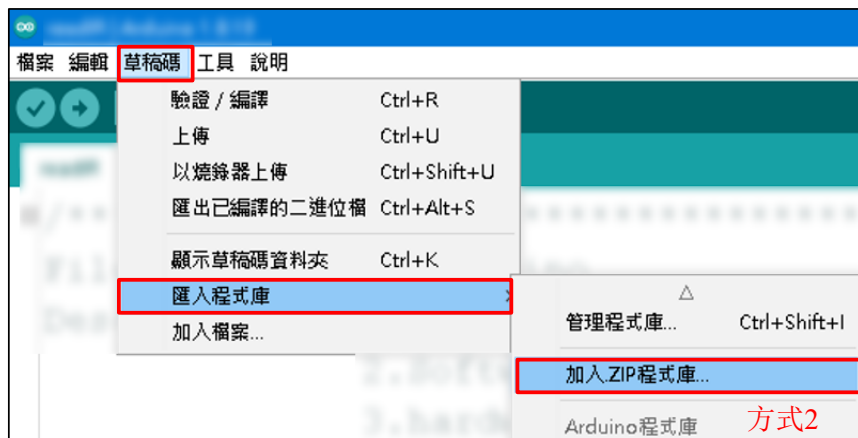


搜索安裝流程 2

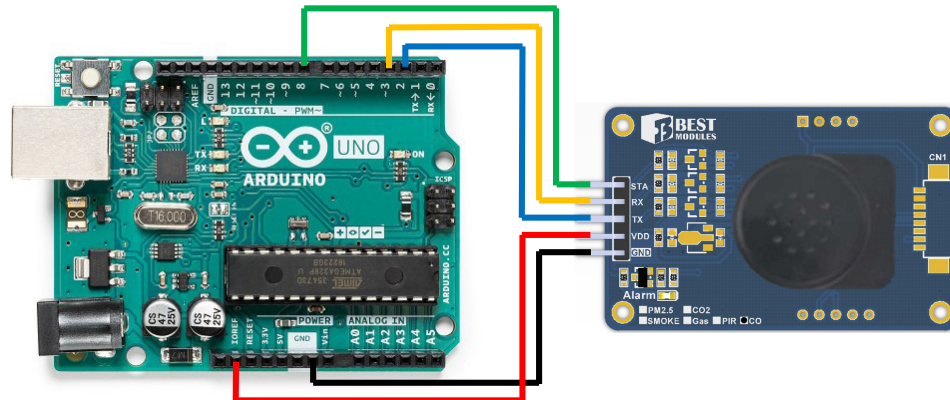
### 方式 2：加入 .ZIP 程式庫，需提前下載 .ZIP 程式庫。

下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bm22s3221-1.html>) 文件目錄下的 Arduino 範例程式 (BM22S3221-1 Library)。

添加 .ZIP 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：通過查詢 STATUS 腳位狀態來判斷是否有報警

- 當模組無報警時 · LED(D13) 閃爍週期 1 秒
- 當模組報警時 · LED(D13) 閃爍週期 0.2 秒

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM22S3221-1) → 選擇範例 (readStatusPin)

2. 範例說明：

a. 構建對象 & 初始化模組

```
#include < BM22S3221-1.h> // 調用 BM22S3221-1 庫

#define LED (13)           // LED 控制引腳 :13

/* 創建對象 & 設置 Software serial 腳位 */
BM22S3221_1 CO(8, 2, 3); // Software serial: 8→STATUS, 2→RX, 3→TX
void setup()
{
  /* 初始化模組 */
  CO.begin(); // Software serial 初始化 (波特率: 9600bps) · 設置 8 號腳位
              // 為輸入模式

  /* 配置序列埠監控視窗 */
  Serial.begin(9600); // Serial 初始化, 波特率 9600bps

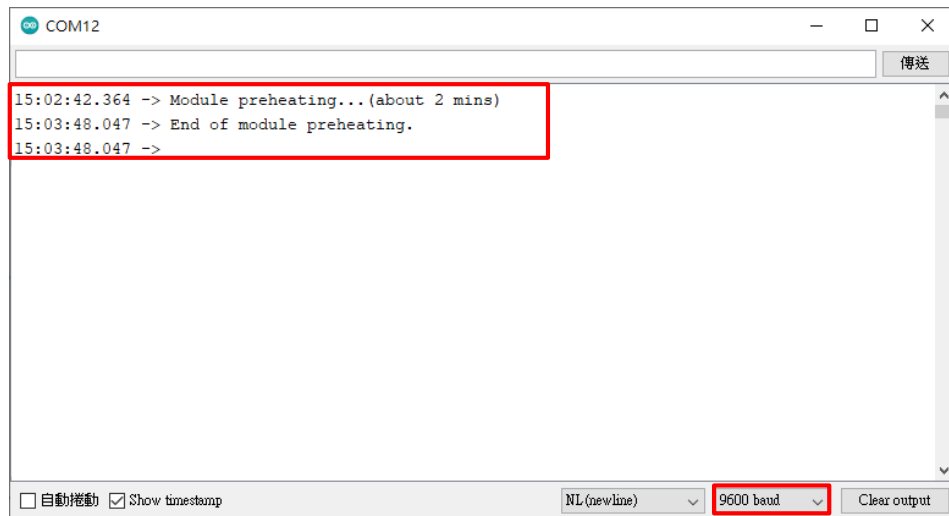
  /* LED 控制腳位初始化 */
  pinMode(LED, OUTPUT);
  digitalWrite(LED, LOW);

  /* 等待模組預熱結束 */
  Serial.println("Module preheating...(about 2 mins)");
  CO.preheatCountdown(); // 等待模組預熱結束
  Serial.println("End of module preheating.");
  Serial.println();
}
```

b. 根據 STATUS 狀態 ( 模組有無報警 ) 改變 LED 閃爍週期

```
void loop()
{
  if (CO.getStatus() == 0)
  {
    /* 無報警 · LED 閃爍週期為 1 秒 */
    digitalWrite(LED, HIGH);
    delay(500);
    digitalWrite(LED, LOW);
    delay(500);
  }
  if (CO.getStatus() == 1)
  {
    /* 報警 · LED 閃爍週期為 0.2 秒 */
    digitalWrite(LED, HIGH);
    delay(100);
    digitalWrite(LED, LOW);
    delay(100);
  }
}
```

3. 打開序列埠監控視窗 · 波特率選擇 9600 · 序列埠監控視窗顯示如下 ; 提示預熱結束後觀察 LED 閃爍情況。



## 範例 2 : readCOValue

範例實現功能：實現功能：接收模組每秒自動輸出的資訊，並將 CO 的相關資訊列印到序列埠監控視窗

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BM22S3221-1) → 選擇範例 (readCOValue)
2. 範例說明：
  - a. 構建對象 & 初始化模組

```
#include < BM22S3221-1.h> // 調用 BM22S3221-1 庫

/* 創建陣列、變數用於存放數據 */
uint8_t moduleInfo[32] = {0};
uint16_t ADValue, COValue, COAlarmThreshold;

/* 創建對象 & 設置 Software serial 腳位 */
BM22S3221_1 CO(8, 2, 3); // Software serial: 8-STATUS, 2-RX, 3-TX
void setup()
{
  /* 初始化模組 */
  CO.begin(); // Software serial 初始化 (波特率: 9600bps) · 設置 8 號腳位
              // 為輸入模式

  /* 配置序列埠監控視窗 */
  Serial.begin(9600); // Serial 初始化 · 波特率 9600bps

  /* 等待模組預熱結束 */
  Serial.println("Module preheating...(about 2 mins)");
  CO.preheatCountdown(); // 等待模組預熱結束
  Serial.println("End of module preheating.");
  Serial.println();
}
```

- b. 接收模組自動發送的數據

```
void loop()
{
  if (CO.isInfoPackage() == true) // 輪詢是否接收到模組發出的數據
  {
    CO.readInfoPackage(moduleInfo); // 讀取模組發送的數據到 moduleInfo[]
    printInfo(); // 列印模組的部分資訊
  }
}
```

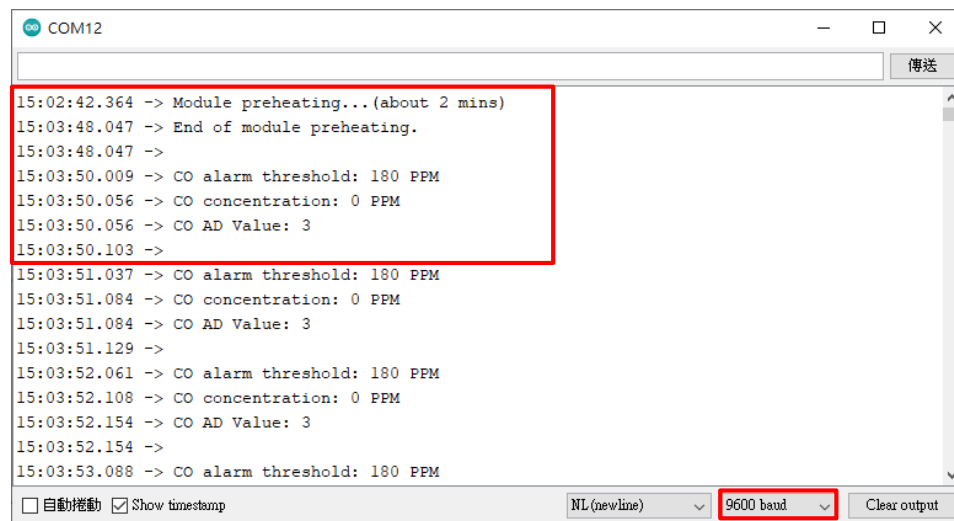
c. 根據接收到的數據 · 列印部分資訊至序列埠監控視窗

```
void printInfo()
{
  /* 列印 CO 報警閾值 (PPM) */
  Serial.print("CO alarm threshold: ");
  COAlarmThreshold = (moduleInfo[21] << 8) + moduleInfo[22];
  Serial.print(COAlarmThreshold);
  Serial.println(" PPM");

  /* 列印當前 CO 濃度 (PPM) */
  Serial.print("CO concentration: ");
  COValue = (moduleInfo[9] << 8) + moduleInfo[10];
  Serial.print(COValue);
  Serial.println(" PPM");

  /* 列印即時 A/D 值 */
  Serial.print("CO AD Value: ");
  ADValue = (moduleInfo[5] << 8) + moduleInfo[6];
  Serial.println(ADValue);
}
```

3. 打開序列埠監控視窗 · 波特率選擇 9600；序列埠監控視窗顯示如下：



```
COM12
15:02:42.364 -> Module preheating...(about 2 mins)
15:03:48.047 -> End of module preheating.
15:03:48.047 ->
15:03:50.009 -> CO alarm threshold: 180 PPM
15:03:50.056 -> CO concentration: 0 PPM
15:03:50.056 -> CO AD Value: 3
15:03:50.103 ->
15:03:51.037 -> CO alarm threshold: 180 PPM
15:03:51.084 -> CO concentration: 0 PPM
15:03:51.084 -> CO AD Value: 3
15:03:51.129 ->
15:03:52.061 -> CO alarm threshold: 180 PPM
15:03:52.108 -> CO concentration: 0 PPM
15:03:52.154 -> CO AD Value: 3
15:03:52.154 ->
15:03:53.088 -> CO alarm threshold: 180 PPM
```

自動捲動  Show timestamp
 
 NL (newline) 9600 baud Clear output

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。