



矩陣式紅外測溫模組

BMS26M833

Arduino Library 說明

版本：V1.00 日期：2023-06-08

www.bestmodulescorp.com

目錄

簡介	3
Arduino Lib 函式	3
Arduino Lib 下載及安裝	7
Arduino 範例	8
範例 1：readTemperature	8
範例 2：setInterrupt	10
範例 3：DisplayThermalImagingOnTheTFT	12

簡介

BMS26M833 是倍創推出的 8×8 矩陣式紅外測溫模組，使用 I²C 通訊方式。本文檔對 BMS26M833 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例演示了讀取溫度等功能。

Arduino Lib 函式

Arduino Lib 名稱：BMS26M833		Lib 版本：V1.0.1
構造函式 & 初始化		
1	BMS26M833(uint8_t intPin = 8, TwoWire * theWire = &Wire)	
	描述	構造函式，選擇中斷腳位以及 wire 的介面
	參數	intPin：INT 腳位默認為 D8，連接模組 INT 腳位 * theWire：I ² C 介面默認為 wire
	返回值	—
備註	—	
2	void begin(uint8_t i2c_addr = BMS26M833_IICADDR)	
	描述	模組初始化
	參數	i2c_addr：I ² C 位址，默認為 0x68
	返回值	void
備註	—	
功能函式		
3	void writeReg(uint8_t addr, uint8_t data)	
	描述	寫入數值至特定的暫存器
	參數	addr：暫存器位址 data：要寫入的數據
	返回值	void
備註	暫存器位址請參考 AMG8833 規格書	
4	uint8_t readReg(uint8_t addr)	
	描述	讀取特定暫存器的數值
	參數	addr：暫存器位址
	返回值	暫存器裡的數據
備註	暫存器位址請參考 AMG8833 規格書	
5	void readReg(uint8_t addr, uint8_t rBuf[], uint8_t rLen)	
	描述	讀取暫存器來獲取數據
	參數	addr：暫存器位址 rBuf[]：用於存儲要獲取的數據的變數 rLen：數據的字節
	返回值	void
備註	—	

6	void readPixels(float tempBuff[])	
	描述	獲取 8×8 像素溫度數據
	參數	tempBuff[]：存儲來自傳感器的溫度數據，單位：°C
	返回值	void
	備註	tempBuff[] 建議長度設為 64-byte
7	void readPixelsAndMaximum(float tempBuff[], float &maxVlaue, float &minVlaue)	
	描述	獲取 8×8 像素溫度數據和其中的最值
	參數	tempBuff[]：存儲來自傳感器的溫度數據，單位：°C &maxVlaue：存儲來自傳感器的溫度數據的最大值，單位：°C &minVlaue：存儲來自傳感器的溫度數據的最小值，單位：°C
	返回值	void
	備註	tempBuff[] 長度建議設為 64-byte
8	void getINTTable(uint8_t buf[], uint8_t size = 8)	
	描述	獲取 8×8 中斷表 (註)
	參數	buf[]：存儲 8×8 中斷表數據 size：要獲取的中斷暫存器的個數 (最多為 8，依次為 INT0~INT7)
	返回值	void
	備註	—
9	uint8_t getOperationMode()	
	描述	獲取工作模式
	參數	—
	返回值	工作模式
	備註	—
10	void sleep()	
	描述	進入休眠模式
	參數	—
	返回值	void
	備註	—
11	void reset(uint8_t mode=INITIAL_RESET)	
	描述	設置復位模式
	參數	mode：復位模式選擇 0x30(FLAG_RESET)：標誌位復位 0x3f(INITIAL_RESET)：中斷復位 (默認)
	返回值	void
	備註	—
12	uint8_t getFrameMode()	
	描述	獲取幀模式
	參數	—
	返回值	幀模式 0x01(FPS_1)：1Hz 0x00(FPS_10)：10Hz
	備註	—

13	uint8_t getINT()	
	描述	獲取 INT 腳位準位
	參數	—
	返回值	INT 腳位準位 1 : HIGH · 溫度在所設範圍內 0 : LOW · 溫度超過所設定的溫度上限值或低於溫度的下限值
	備註	—
14	uint8_t getStatus()	
	描述	獲取狀態暫存器
	參數	—
	返回值	狀態暫存器數據
	備註	—
15	uint8_t getAverageOutputMode()	
	描述	獲取平均輸出模式
	參數	—
	返回值	輸出模式 0x20(TWICE_MOVE_AVE_OUTPUT) : 兩次輸出模式 0x00(ONE_MOVE_OUTPUT) : 一次輸出模式
	備註	—
16	float readThermistorTemp()	
	描述	獲取熱敏電阻溫度值
	參數	—
	返回值	溫度值 · 單位 : °C
	備註	此溫度值一般用於測量環境溫度
參數配置		
17	void beginMeasure(uint8_t mode= NORMAL_MODE)	
	描述	開始測量
	參數	mode : 測量模式選擇 0x00(NORMAL_MODE) : 正常模式 (默認) 0x20(STAND_BY_MODE_60SEC) : 60s 測量一次 0x21(STAND_BY_MODE_10SEC) : 10s 測量一次
	返回值	void
	備註	—
18	void setFrameMode(uint8_t mode =FPS_10)	
	描述	設置幀模式
	參數	mode : 幀模式選擇 0x01(FPS_1) : 1Hz 0x00(FPS_10) : 10Hz (默認)
	返回值	void
	備註	—

19	void setINT(uint8_t isEnabled = false)	
	描述	設置中斷狀態
	參數	isEnabled：中斷模式選擇 true：使能 false：除能
	返回值	void
	備註	—
20	void setInterruptLevels(float high, float low)	
	描述	設置中斷的溫度閾值
	參數	high：溫度上限閾值，單位：°C low：溫度下限閾值，單位：°C
	返回值	void
	備註	默認滯後值為 high 的 95%
21	void setInterruptLevels(float high, float low, float hysteresis)	
	描述	設置中斷的溫度閾值
	參數	high：溫度上限閾值，單位：°C low：溫度下限閾值，單位：°C hysteresis：中斷檢測的滯後值，單位：°C
	返回值	void
	備註	解析度為 0.25°C
22	void setStatusClear()	
	描述	清除狀態暫存器
	參數	—
	返回值	void
	備註	同時清除以下標誌位： OVT：熱敏電阻溫度溢出標誌位 (當輸出值 (REG：0x0E~0x0F) 超過 0xFFFF) OVS：像素溫度溢出標誌位 (當輸出值 (REG：0x80~0xFF) 超過 0xFFFF) INT：中斷標誌位
23	void setAverageOutputMode(uint8_t mode=ONE_MOVE_OUTPUT)	
	描述	設置平均輸出模式
	參數	mode：輸出模式選擇 0x20(TWICE_MOVE_AVE_OUTPUT)：兩次輸出模式 0x00(ONE_MOVE_OUTPUT)：一次輸出模式
	返回值	void
	備註	—
24	void setOperationMode(uint8_t mode)	
	描述	設置工作模式
	參數	mode：輸出模式選擇 0x00(NORMAL_MODE)：正常工作模式 0x10(SLEEP_MODE)：睡眠模式 0x20(STAND_BY_MODE_60SEC)：待機模式_60 秒 0x21(STAND_BY_MODE_10SEC)：待機模式_10 秒
	返回值	void
	備註	—

註：8×8 中斷表如下表所示 (與傳感器的像素點一致)，當某個像素溫度超過溫度上限值或低於下限值時，對應的中斷置 1，否則為 0。

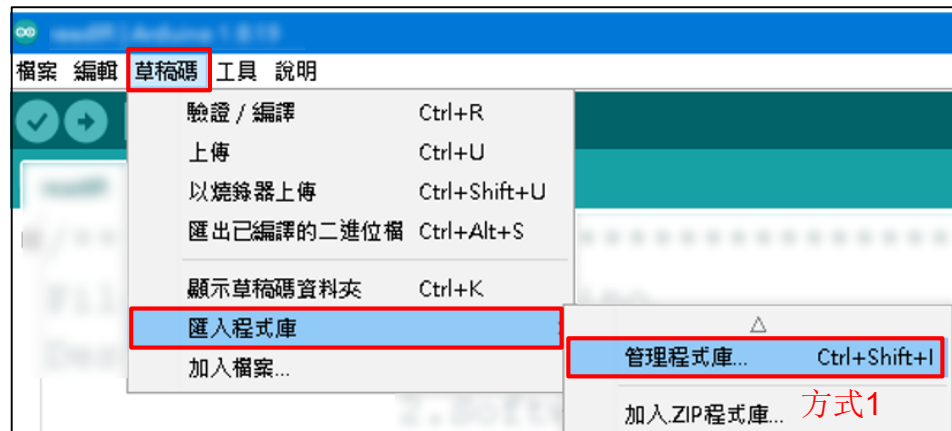
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
byte 0	0	1	0	0	0	0	0	1
byte 1	0	0	0	0	0	0	0	0
byte 2	0	0	0	0	0	0	0	0
byte 3	0	0	0	1	0	0	0	0
byte 4	0	0	0	0	0	0	0	0
byte 5	0	0	0	0	0	0	0	0
byte 6	0	0	0	0	0	1	0	0
byte 7	0	0	0	0	0	0	0	0

Arduino Lib 下載及安裝

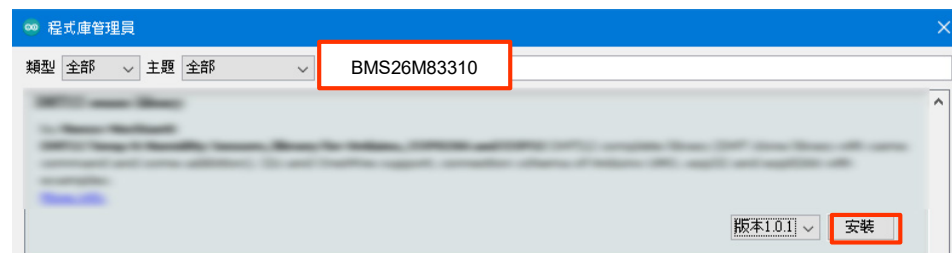
BMS26M833 Arduino Library : 可參考下面兩種方法安裝 BMS26M833 的 Arduino Library

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 BMS26M833 → 安裝



搜索安裝流程 1

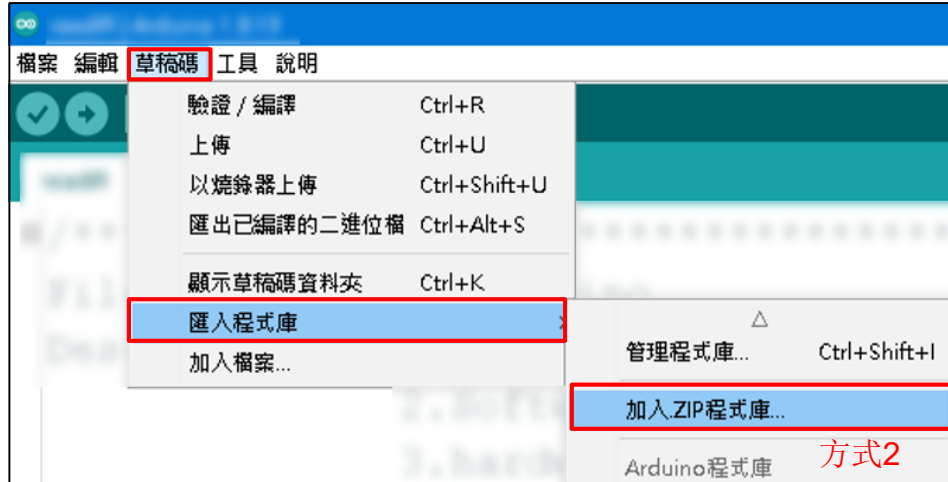


搜索安裝流程 2

方式 2：添加 .ZIP 程式庫 · 需提前下載 .ZIP 程式庫

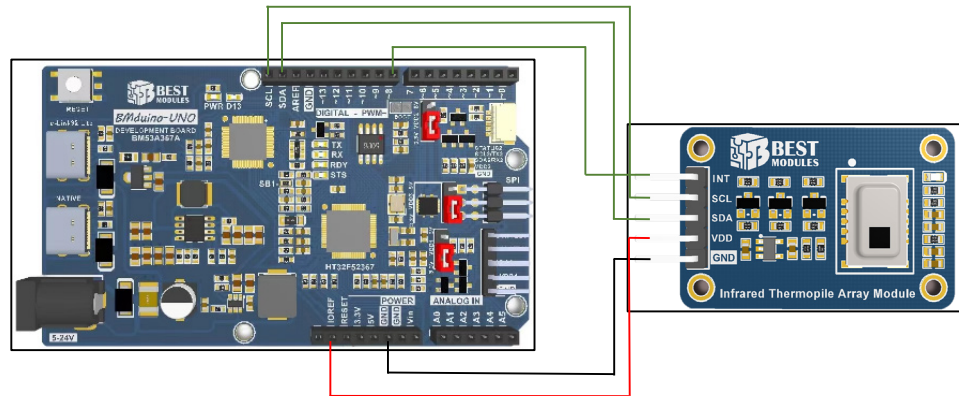
下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bms26m833.html#tab-product2>) · 下載“文件”目錄下的 Arduino 範例程式 (BMS26M833 Library)。

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

範例 1：readTemperature

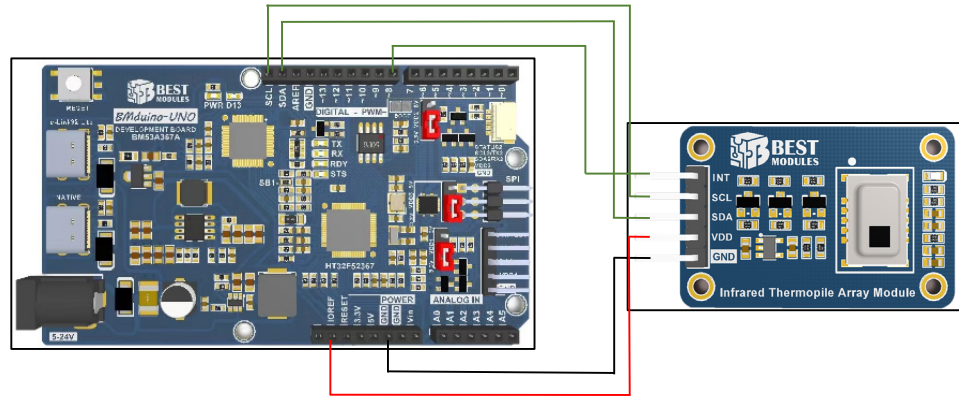


實物連接示意圖

範例 1 實現功能：獲取 8×8 溫度值 · 並在串口監視器上顯示

- 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMS26M833) → 選擇對應範例 (readTemperature)

範例 2 : setInterrupt



實物連接示意圖

範例 2 實現功能：使能模組的中斷功能，獲取 8×8 中斷表，並在串口監視器上顯示

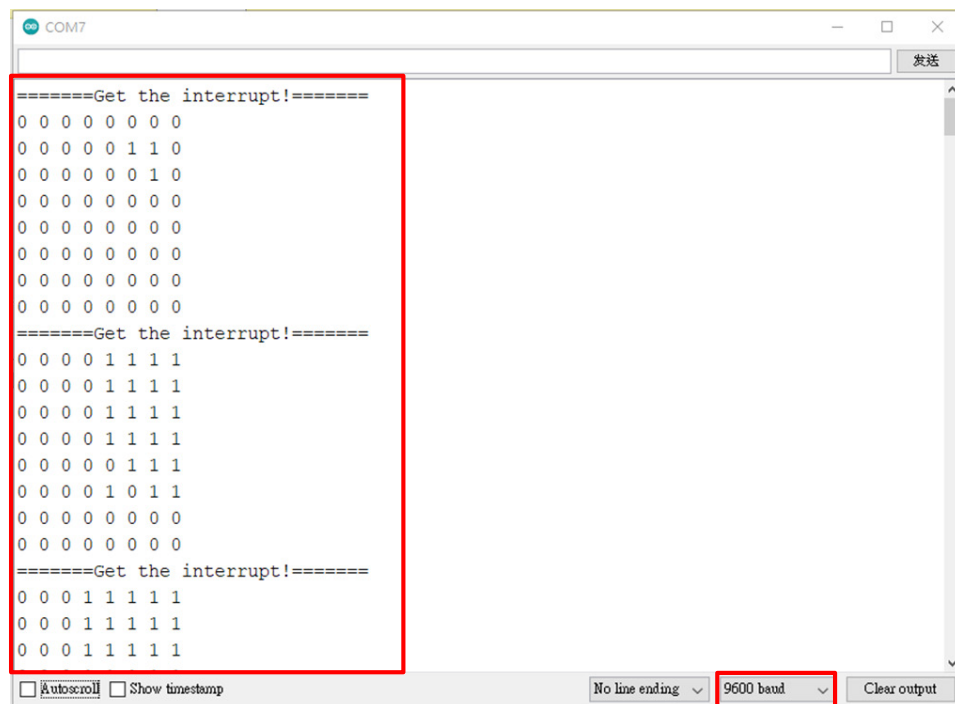
1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMS26M833) → 選擇對應範例 (setInterrupt)
2. 範例說明：
 - a. 創建對象 & 初始化模組

```
#include <BMS26M833.h>
// 設定超溫報警的上限值和下限值，需注意解析度為 0.25°C
#define TEMP_INT_HIGH 30
#define TEMP_INT_LOW 15
BMS26M833 Amg; // 創建對象
uint8_t interruptTable[8];
void setup()
{
  Serial.begin(9600);
  Amg.begin(); // 模組初始化
  Amg.setInterruptLevels(TEMP_INT_HIGH, TEMP_INT_LOW); // 設定上限值與下
  // 限值
  Amg.setINT(true); // 配置中斷使能
}
```

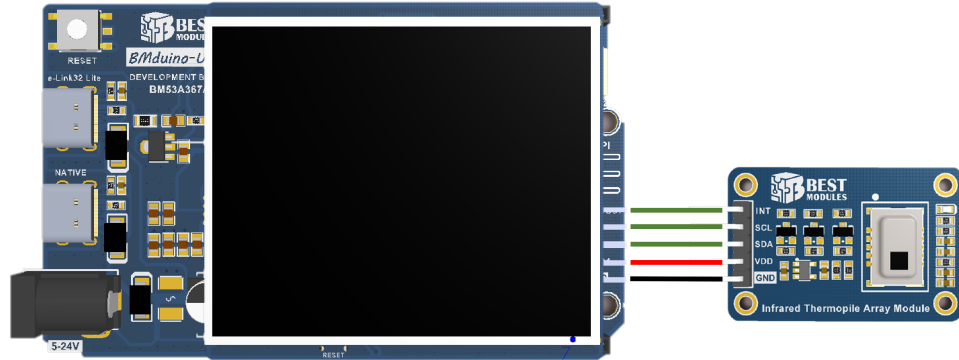
b. 獲取中斷表並在串口監視器中顯示

```
void loop()
{
  if(Amg.getINT() == 0) // 當 INT 腳為低準位時，說明有中斷發生
  {
    Amg.getINTTable(interruptTable); // 獲取中斷表
    Serial.println("====Get the interrupt!====");
    for(int i=0;i<8;i++)
    {
      for(int j=7;j>=0;j--)
      {
        bool intBit = interruptTable[i] & (1<<j);
        Serial.print(intBit);
        Serial.print(" ");
      }
      Serial.println();
    }
    Amg.setStatusClear(); // 清除中斷標誌位以便於接收下一次中斷
  }
}
```

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 : DisplayThermalImagingOnTheTFT



實物連接示意圖

範例 3 實現功能：獲取溫度值，通過相關演算法處理，生成熱成像並配合使用 BMD58T280 TFT 模組顯示圖像

(需要下載 BMD58T280 相關的庫檔案，具體安裝方法：Arduino IDE → 草稿碼 → 匯入程式庫 → 管理程式庫 → 搜索 BMD58T280 → 安裝)

1. 範例打開：Arduino IDE → 檔案 → 範例 → Lib 選擇 (BMS26M833) → 選擇對應範例 (DisplayThermalImagingOnTheTFT)
2. 範例說明：
 - a. 構建 & 初始化模組

```
#include <SPI.h>
#include <BMD58T280.h>
#include "lcdfont.h"
#include <BMS26M833.h>

/* 熱成像演算法採用 c 編譯 */
extern "C" {
    #include "InfraredThermalImaging.h" // 演算法
}

#if !defined(ARDUINO_AVR_UNO) // 選擇 BMduino 開發板
    BMD58T280 TFTscreen; // 選擇 EBI 通訊
#else // 選擇 UNO 開發板
    BMD58T280 TFTscreen(&SPI); // 選擇 SPI 通訊
#endif

/* 宏定義方便修改調試 */
#define OutMatWidth 55 // 輸出熱成像寬度
#define OutMatHeight 55 // 輸出熱成像高度
#define Magnification 3 // 輸出熱成像的放大倍數
#define TempDiffConfig 4 // 當溫差小於此值時，認作無效熱成像，不予顯示，
// 單位：°C

#define BackgroundConfig 20 // 背景去除程度，0 為 DISABLE，值越大，目標與背
// 景之間的過度層越薄，經驗值為 20，範圍為 0-255
#define ColorBarConfig Rainbow2_65K // 提前將調色板 RGB 轉為 65K 值，顯示過程
// 中采用查表方式可以提升前端顯示速度
```

```
ThermalImagingConfig ThermImaConfig;
BMS26M833 amg(22,&Wire1); //AM8833 Sencor 實例化

float TempMat[8 * 8];
uint8_t DataBuff[OutMatWidth * OutMatHeight];
uint16_t timecnt=0;
float TempMax,TempMin;
char AxisPrintout[20];
uint16_t maging_xStart,maging_yStart;
uint32_t Systemtime = 0;

void setup() {
  Serial.begin(9600);           // 串口初始化
  TFTscreen.begin();           //TFT 初始化
  TFTscreen.setRotation(1);    // 設置屏幕顯示方向
  amg.begin();                  // 模組初始
  LCDShowColorbar();
  maging_xStart = 100-Magnification*25; // 定位熱成像畫面 x 座標位置
  maging_yStart = 130-Magnification*25; // 定位熱成像畫面 y 座標位置
  ThermImaConfig.InWidth      = 8;
  ThermImaConfig.InHeight     = 8;
  ThermImaConfig.InMat        = TempMat;
  ThermImaConfig.OutMat       = DataBuff;
  ThermImaConfig.OutWidth     = OutMatWidth;
  ThermImaConfig.OutHeight    = OutMatHeight;
  ThermImaConfig.Background   = BackgroundConfig;
  ThermImaConfig.TempDiff     = TempDiffConfig;
}
```

b. 獲取溫度值並通過演算法處理，顯示在 LCD 屏幕上

```
void loop()
{
  amg.readPixelsAndMaximum(TempMat, TempMax, TempMin); // 獲取溫度以及 // 最值
  if(Temp_Min>0 && Temp_Max<80) // 測溫範圍 0~80°C
  {
    InfraredThermalImaging(&ThermImaConfig); // 演算法處理
    LCDShow(DataBuff,Temp_Max,Temp_Min,OutMatWidth,OutMatHeight,
            TempDiffConfig,Magnification,maging_xStart,maging_yStart); // 在屏幕 // 上顯示
  }
  else
  {
    LCDShow(DataBuff,0,0,OutMatWidth,OutMatHeight,TempDiffConfig,
            Magnification,maging_xStart,maging_yStart );
    delay(30);
  }
}
```

```

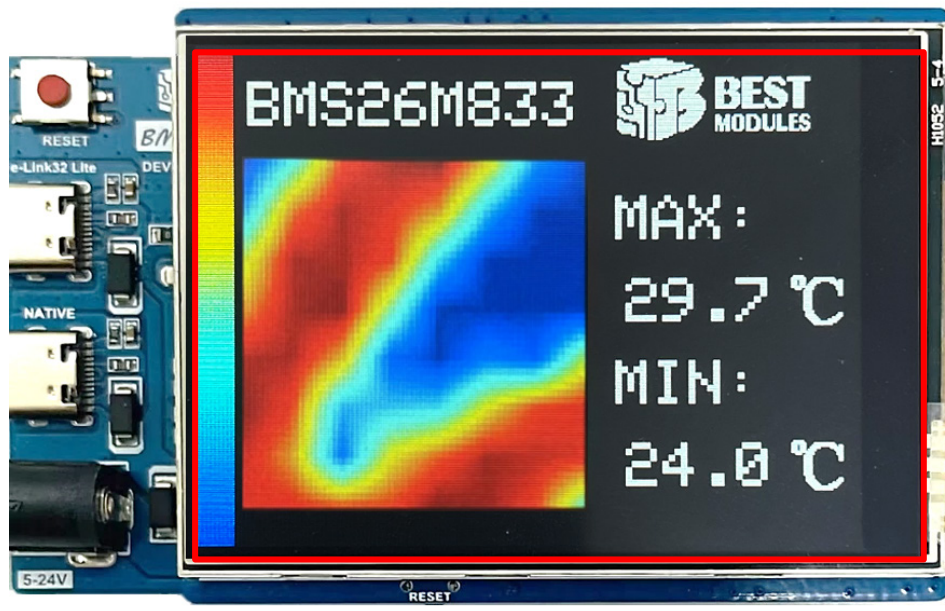
/* 調用此函式：更新溫度值與熱成像畫面在 TFT 屏幕上 */
void LCDShow(uint8_t *Buff,float Max,float Min,uint8_t Width,uint8_t
Height,uint8_t TempDiff,uint8_t Mul,uint16_t xStart,uint16_t yStart)
{
    uint16_t temp;
    uint16_t color;
    uint16_t i,j,k,l,xLeng,yLeng;
    xLeng = Mul * Width;
    yLeng = Mul * Height;
    LCD_ShowFloatNum1(210,110,Max,3,BM_ILI9341::WHITE,BM_ILI9341::
BLACK,3);
    LCD_ShowFloatNum1(210,190,Min,3,BM_ILI9341::WHITE,BM_ILI9341::
BLACK,3);
    if((Max-Min) <= TempDiff)
    {
        color=ColorBarConfig[0];
        TFTscreen.fill(color);
        TFTscreen.noStroke();
        TFTscreen.rect(xStart,yStart,xLeng,yLeng);
        return;
    }
    TFTscreen.setAddrWindow(xStart,yStart,xLeng,yLeng);
    for(i = 0;i < Height;i++)
    {
        for(k = 0;k < Mul;k++)
        {
            for(j = 0;j < Width;j++)
            {
                temp = i * Width + j;
                color=ColorBarConfig[Buff[temp]];
                TFTscreen.writeColor(color,Mul);
            }
        }
    }
}

/* 調用此函式：顯示小數在 TFT 屏幕上 */
void LCD_ShowFloatNum1(uint16_t x,uint16_t y,float num,uint8_t len,
u32 fc,u32 bc,uint8_t sizey)
{
    String strResult = String(num);
    strResult.toCharArray(AxisPrintout, 5);
    TFTscreen.fill(bc);
    TFTscreen.noStroke();
    TFTscreen.rect(x-1, y-2, 70, 26);//background color
    TFTscreen.stroke(fc);//word color
    TFTscreen.setTextSize(sizey);
    TFTscreen.text(AxisPrintout, x, y);//display variable
}

```

```
/* 調用此函式：顯示初始畫面在 TFT 屏幕上 */  
void LCDShowColorbar()  
{  
  TFTscreen.background(BM_ILI9341::BLACK);  
  TFTscreen.stroke(BM_ILI9341::WHITE);  
  TFTscreen.setTextSize(3);  
  TFTscreen.text("MAX:", 205, 70);  
  TFTscreen.text("MIN:", 205, 150);  
  TFTscreen.stroke(BM_ILI9341::WHITE);  
  TFTscreen.setTextSize(3);  
  TFTscreen.text("BMS26M833", 25, 15);  
  TFTscreen.drawImage(288,105,gImage_tempLogo,32,32,0xFFFF);//Display  
  //°C  
  TFTscreen.drawImage(288,185,gImage_tempLogo,32,32,0xFFFF);//Display  
  //°C  
  TFTscreen.drawImage(200,0,gImage_BMLogo,104,44,0xFFFF);//Display  
  //BMLOGO  
  
  uint8_t a=0;  
  uint8_t i;  
  a=0;  
  for(i = 245;i > 0;i--)  
  {  
    TFTscreen.stroke(ColorBarConfig[i]);  
    TFTscreen.line(0,a,18,a);  
    a++;  
  }  
  TFTscreen.noStroke();  
}
```

3 · 按要求連接電路 · LCD 屏幕顯示效果如下：



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。