



0.96" OLED 顯示模組

BMD31M090

Arduino Library V1.0.1 說明

版本：V1.20 日期：2024-01-23

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：display	9

簡介

BMD31M090 是倍創推出的一款 0.96 寸 OLED 顯示模組，使用 I²C 通訊方式。本文檔對 BMD31M090 的 Arduino Lib 函式、Arduino Lib 安裝方式進行說明；範例演示 OLED 顯示。

Arduino Lib 函式

Arduino Lib 名稱：BMD31M090		Lib 版本：V1.0.1
構造函式 & 初始化		
1	BMD31M090(uint8_t width, uint8_t height, TwoWire *theWire=&Wire)	
	描述	構造函式，設定 Display width，height 以及 wire 介面
	參數	width：像素顯示寬度 height：像素顯示高度 * theWire：wire 參數
	返回值	—
	備註	—
2	void begin(uint8_t i2c_addr=BMD31M090_DEVICEADDR0, uint32_t clkFrequency=BMD31M090_CLKFREQ)	
	描述	模組初始化，設定 I ² C 設備位址和 I ² C 通訊速率
	參數	i2c_addr：I ² C 設備位址，預設為 0x3C clkFrequency：I ² C 通訊速率，預設為 400kHz
	返回值	void
	備註	—
功能函式		
3	void clearDisplay(void)	
	描述	清除顯示緩衝區的內容 (將所有像素設定為關閉)
	參數	void
	返回值	void
	備註	需搭配 display() 使用
4	void display(void)	
	描述	顯示當前緩衝區的內容
	參數	void
	返回值	void
	備註	—
5	void drawPixel(uint8_t x, uint8_t y, uint8_t pixelColor)	
	描述	畫一個像素點
	參數	x：x 座標，範圍 0~127 y：y 座標，範圍 0~63 pixelColor：像素顏色 0 (pixelColor_BLACK)：像素顏色為黑 1 (pixelColor_WHITE)：像素顏色為白 2 (pixelColor_INVERSE)：像素顏色翻轉
	返回值	void
	備註	需搭配 display() 使用

6	void drawLine(uint8_t x_Start, uint8_t y_Start, uint8_t x_End, uint8_t y_End, uint8_t pixelColor)	
	描述	畫線
	參數	x_Start：起點 x 座標 y_Start：起點 y 座標 x_End：終點 x 座標 y_End：終點 y 座標 pixelColor：像素顏色 0 (pixelColor_BLACK)：像素顏色為黑 1 (pixelColor_WHITE)：像素顏色為白 2 (pixelColor_INVERSE)：像素顏色翻轉
	返回值	void
	備註	需搭配 display() 使用
7	void drawFastHLine(uint8_t x, uint8_t y, uint8_t width, uint8_t pixelColor)	
	描述	畫水平線
	參數	x：x 座標，範圍 0~127 y：y 座標，範圍 0~63 width：寬度，width(w)≤128 pixelColor：像素顏色 0 (pixelColor_BLACK)：像素顏色為黑 1 (pixelColor_WHITE)：像素顏色為白 2 (pixelColor_INVERSE)：像素顏色翻轉
	返回值	void
	備註	需搭配 display() 使用
8	void drawFastVLine(uint8_t x, uint8_t y, uint8_t height, uint8_t pixelColor)	
	描述	畫垂直線
	參數	x：x 座標，範圍 0~127 y：y 座標，範圍 0~63 height：高度，height(h)≤64 pixelColor：像素顏色 0 (pixelColor_BLACK)：像素顏色為黑 1 (pixelColor_WHITE)：像素顏色為白 2 (pixelColor_INVERSE)：像素顏色翻轉
	返回值	void
	備註	需搭配 display() 使用
9	void drawChar(uint8_t x, uint8_t row, uint8_t chr)	
	描述	寫字元
	參數	x：x 座標，範圍 0~127 row：列，範圍 0~7 0 (displayROW0)：列 0 1 (displayROW1)：列 1 2 (displayROW2)：列 2 3 (displayROW3)：列 3 4 (displayROW4)：列 4 5 (displayROW5)：列 5 6 (displayROW6)：列 6 7 (displayROW7)：列 7 chr：字體表上的字元
	返回值	void
	備註	—

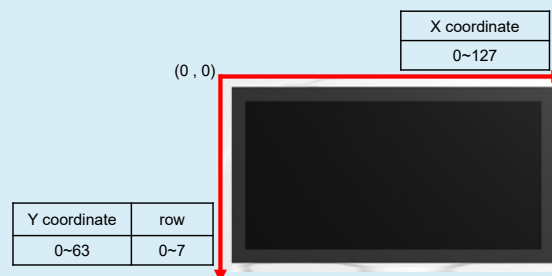
10	void drawString(uint8_t x, uint8_t row, uint8_t *str)	
	描述	寫字串
	參數	x : x 座標 · 範圍 0~127 row : 列 · 範圍 0~7 0 (displayROW0) : 列 0 1 (displayROW1) : 列 1 2 (displayROW2) : 列 2 3 (displayROW3) : 列 3 4 (displayROW4) : 列 4 5 (displayROW5) : 列 5 6 (displayROW6) : 列 6 7 (displayROW7) : 列 7 *str : 字體表上的字串
	返回值	void
	備註	—
11	void drawNum(uint8_t x, uint8_t row, uint32_t num, uint8_t numLen)	
	描述	寫整型數字
	參數	x : x 座標 · 範圍 0~127 row : 列 · 範圍 0~7 0 (displayROW0) : 列 0 1 (displayROW1) : 列 1 2 (displayROW2) : 列 2 3 (displayROW3) : 列 3 4 (displayROW4) : 列 4 5 (displayROW5) : 列 5 6 (displayROW6) : 列 6 7 (displayROW7) : 列 7 num : 字體表上的數字 numLen : 數字長度
	返回值	void
	備註	—
12	void drawBitmap(int8_t x, int8_t y, const uint8_t *Bitmap, uint8_t w, uint8_t h, uint8_t pixelColor)	
	描述	畫圖
	參數	x : x 座標 · 範圍 0~127 y : y 座標 · 範圍 0~63 *Bitmap : 點陣圖名稱 w : 點陣圖寬度 h : 點陣圖高度 pixelColor : 像素顏色 0 (pixelColor_BLACK) : 像素顏色為黑 1 (pixelColor_WHITE) : 像素顏色為白 2 (pixelColor_INVERSE) : 像素顏色翻轉
	返回值	void
	備註	需搭配 display() 使用

13	void startScrollRight(uint8_t startRow, uint8_t endRow, uint8_t scrollSpeed, uint8_t scrollVDirection=SCROLLV_NONE)	
	描述	開始向右滾動
	參數	startRow：起始列位址・範圍 0~7 endRow：結束列位址・範圍 0~7 scrollSpeed：滾動速度 0x07 (SCROLL_2FRAMES)：滾動 2 幀 0x04 (SCROLL_3FRAMES)：滾動 3 幀 0x05 (SCROLL_4FRAMES)：滾動 4 幀 0x00 (SCROLL_5FRAMES)：滾動 5 幀 0x06 (SCROLL_25FRAMES)：滾動 25 幀 0x01 (SCROLL_64FRAMES)：滾動 64 幀 0x02 (SCROLL_128FRAMES)：滾動 128 幀 0x03 (SCROLL_256FRAMES)：滾動 256 幀 scrollVDirection：對角滾動方向 0x00 (SCROLLV_NONE)：不滾動 0x01 (SCROLLV_TOP)：向上滾動 0x3F (SCROLLV_BOTTOM)：向下滾動
	返回值	void
	備註	—
14	void startScrollLeft(uint8_t startRow, uint8_t endRow, uint8_t scrollSpeed, uint8_t scrollVDirection=SCROLLV_NONE)	
	描述	開始向左滾動
	參數	startRow：起始列位址・範圍 0~7 endRow：結束列位址・範圍 0~7 scrollSpeed：滾動速度 0x07 (SCROLL_2FRAMES)：滾動 2 幀 0x04 (SCROLL_3FRAMES)：滾動 3 幀 0x05 (SCROLL_4FRAMES)：滾動 4 幀 0x00 (SCROLL_5FRAMES)：滾動 5 幀 0x06 (SCROLL_25FRAMES)：滾動 25 幀 0x01 (SCROLL_64FRAMES)：滾動 64 幀 0x02 (SCROLL_128FRAMES)：滾動 128 幀 0x03 (SCROLL_256FRAMES)：滾動 256 幀 scrollVDirection：對角滾動方向 0x00 (SCROLLV_NONE)：不滾動 0x01 (SCROLLV_TOP)：向上滾動 0x3F (SCROLLV_BOTTOM)：向下滾動
	返回值	void
	備註	—
15	void stopScroll(void)	
	描述	停止滾動
	參數	void
	返回值	void
	備註	—

參數設定		
16	void setFont(const unsigned char* font)	
	描述	設定字體格式
	參數	font：字體格式 FontTable_6X8：字體格式為 6×8 FontTable_8X16：字體格式為 8×16 FontTable_16X32：字體格式為 16×32 FontTable_32X64：字體格式為 32×64
	返回值	void
	備註	—
17	void setPixelRow(uint8_t x, uint8_t row)	
	描述	設定顯示在哪一列
	參數	x：x 座標，範圍 0~127 row：列，範圍 0~7
	返回值	void
	備註	—
18	void dim(bool dim)	
	描述	設定畫面亮度
	參數	dim：亮度選擇 true：暗 false：正常亮度
	返回值	void
	備註	—
19	void invertDisplay(bool i)	
	描述	反白顯示
	參數	i：是否反白 true：白底黑字 (black-on-white) false：黑底白字 (white-on-black)
	返回值	void
	備註	—

註：OLED 屏顯示，豎方向根據函式，採用不同的方式

- (1) 函式 #5、#6、#7、#8、#12，豎方向採用 "Y 座標"，範圍為 0~63
- (2) 函式 #9、#10、#11、#13、#14、#17，豎方向採用 "列"，範圍為列 0~列 7

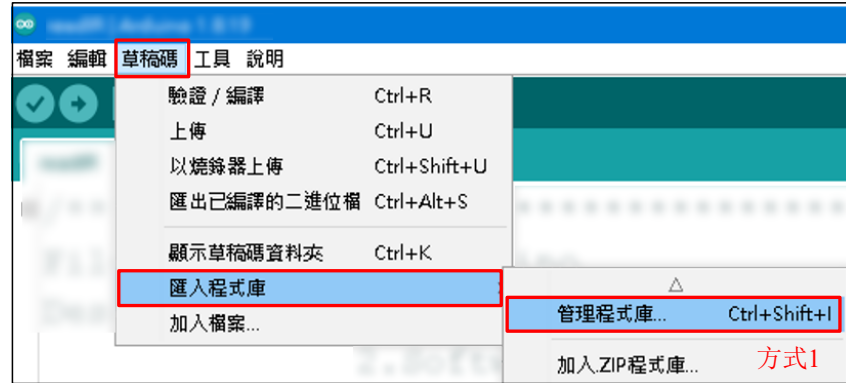


Arduino Lib 下載及安裝

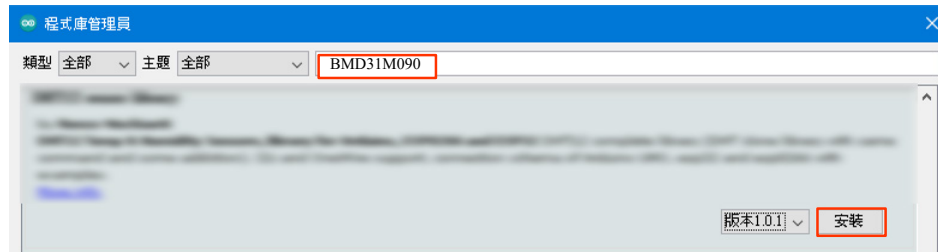
BMD31M090 Library：可參考下面兩種方法安裝 BMD31M090 的 Arduino Library

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 BMD31M090 → 安裝



搜索安裝流程 1

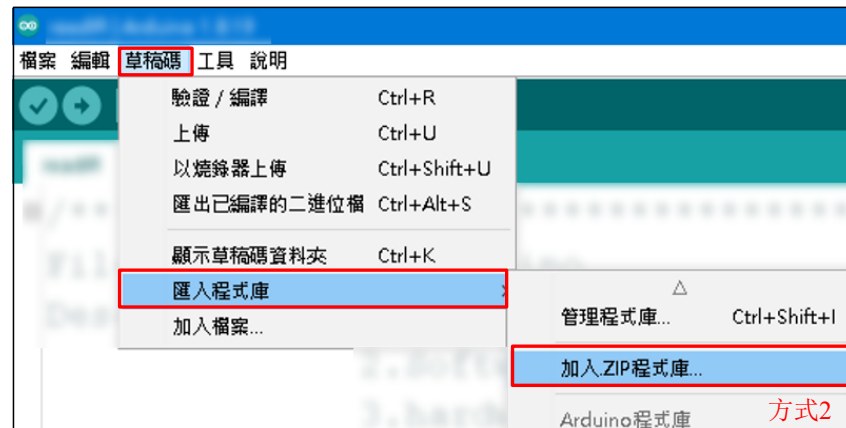


搜索安裝流程 2

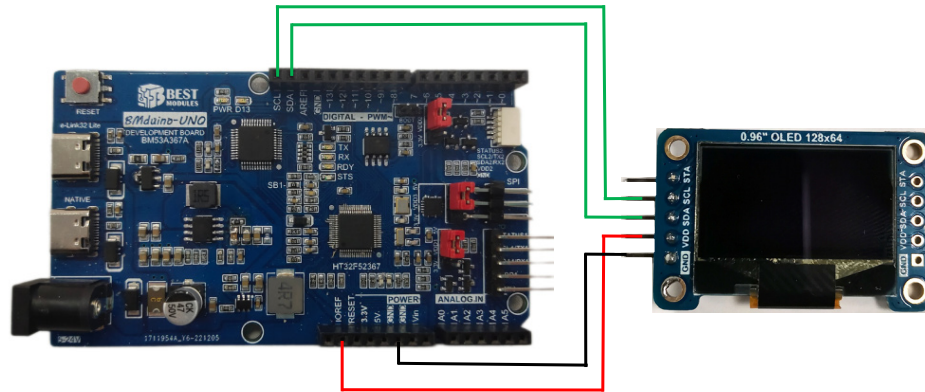
方式 2：添加 .ZIP 程式庫，需提前下載 .ZIP 程式庫

下載方法：打開倍創官方網站 (<https://www.bestmodulescorp.com/bmd31m090.html>) 檔目錄下的 Arduino 範例程式 (BMD31M090 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：display

範例實現功能：將 BMD31M090 內所提供函式功能進行使用，於 OLED 模組上進行顯示。

1. 範例打開：檔案 → 範例 → Lib 選擇 (BMD31M090) → 選擇範例 (display)
2. 範例說明：
 - a. 構建 & 初始化對象

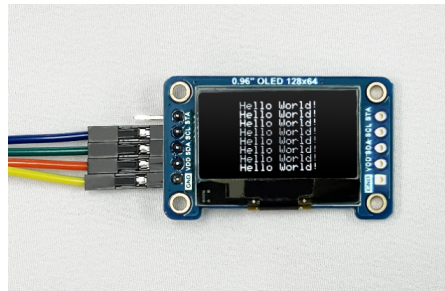
```
#include "BMD31M090.h"
#include "Bitmap.h"
#define BMD31M090_WIDTH 128           // 設定模組顯示寬度，以像素為單位
#define BMD31M090_HEIGHT 64          // 設定模組顯示高度，以像素為單位
#define BMD31M090_ADDRESS 0x3C       // 設定模組 I2C 位址
uint8_t t = ' ';
BMD31M090 BMD31(BMD31M090_WIDTH, BMD31M090_HEIGHT, &Wire); // 建立對象
void setup()
{
    Serial.begin(115200);              // 設定序列埠監視視窗
    Serial.println("BMD31M090 0.96\" OLED Module Sketch");
    BMD31.begin(BMD31M090_ADDRESS);    // 初始化模組
    delay(100);                        // 初始化模組後延遲 100ms
    test_drawString_6x8();             // 測試功能：繪製字串繪製字體 (6×8) 你好世界！
    test_drawString_8x16();            // 測試功能：繪製字串繪製字體 (8×16) 你好世界！
    test_drawString_drawChar_drawNum(); // 測試功能：drawString、drawChar
                                        // 和 drawNum
    test_drawPixel();                  // 測試功能：drawPixel
    test_drawFastHLine_drawFastVLine(); // 測試功能：drawFastHLine 和
                                        // drawFastVLine
    test_drawBitmap();                 // 測試功能：繪製點陣圖顯示 BestModule_LOGO 和
                                        // NameBestModule_LOGO
    test_variousScroll();               // 測試功能：滾動功能可滾動各個方向
    test_invertDisplay();               // 測試功能：反轉顯示和恢復顯示
    test_dim();                         // 測試功能：變暗顯示
}
```

b. loop

```
void loop()
{
}
```

c. test_drawString_6x8()

```
void test_drawString_6x8(void)
{
    BMD31.clearDisplay();
    BMD31.display();
    uint8_t col, row;
    BMD31.setFont(FontTable_6X8); // 設定字體類型 - 6x8 (預設字體類型: 8x16)
    col = (128 - (6 * sizeof( "Hello World!" ))) / 2;
    for (row=0; row<8; row++)
    {
        BMD31.drawString(col, row, (u8*) "Hello World!" );
    }
    delay(500);
}
```

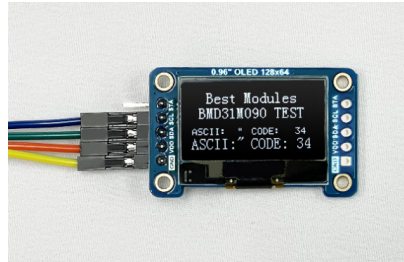
**d. test_drawString_8x16()**

```
void test_drawString_8x16(void)
{
    BMD31.clearDisplay();
    BMD31.display();
    uint8_t col, row;
    BMD31.setFont(FontTable_8X16); // 設定字體類型 - 8x16
    col = (128 - (8 * sizeof( "Hello World!" ))) / 2;
    for (row=0; row<8; row+=2)
    {
        BMD31.drawString(col, row, (u8*) "Hello World!" );
    }
    delay(500);
}
```



e. test_drawString_drawChar_drawNum()

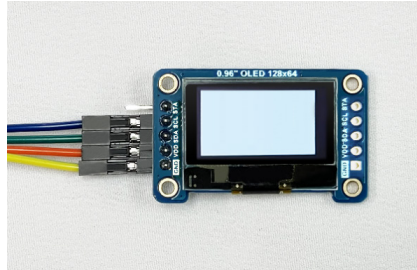
```
void test_drawString_drawChar_drawNum(void)
{
    BMD31.clearDisplay();
    BMD31.display();
    BMD31.drawString(16, displayROW0, (u8*) "Best Modules" );
    BMD31.drawString(8, displayROW2, (u8*) "BMD31M090 TEST" );
    // 按順序顯示字體 6×8 和 8×16 的 ASCII " " (0x20/32) 到 "~" (0x7E/126) 值
    do
    {
        BMD31.setFont(FontTable_6X8);
        BMD31.drawString(0, displayROW5, (u8*) "ASCII:" );
        BMD31.drawString(63, displayROW5, (u8*) "CODE:" );
        BMD31.drawChar(48, displayROW5, t);
        BMD31.drawNum(103, displayROW5, t, 3);
        BMD31.setFont(FontTable_8X16);
        BMD31.drawString(0, displayROW6, (u8*) "ASCII:" );
        BMD31.drawString(63, displayROW6, (u8*) "CODE:" );
        BMD31.drawChar(48, displayROW6, t);
        BMD31.drawNum(103, displayROW6, t, 3);
        delay(10);
    } while(++t < '~' );
    t=' ';
}
}
```



f. test_drawPixel()

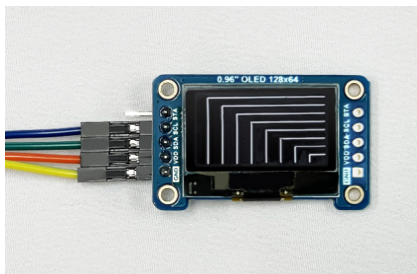
```
void test_drawPixel(void)
{
    BMD31.clearDisplay();
    for (uint8_t col=0; col<128; col++)
    {
        for(uint8_t row=0; row<64; row++)
        {
            BMD31.drawPixel(col, row, pixelColor_WHITE);
        }
    }
    BMD31.display();
    delay(500);
    for (uint8_t col=0; col<128; col++)
    {
        for(uint8_t row=0; row<32; row++)
        {
            BMD31.drawPixel(col, row, pixelColor_BLACK);
        }
    }
    BMD31.display();
}
```

```
delay(500);
for (uint8_t col=0; col<128; col++)
{
    for(uint8_t row=0; row<64; row++)
    {
        BMD31.drawPixel(col, row, pixelColor_INVERSE);
    }
}
BMD31.display();
delay(500);
}
```



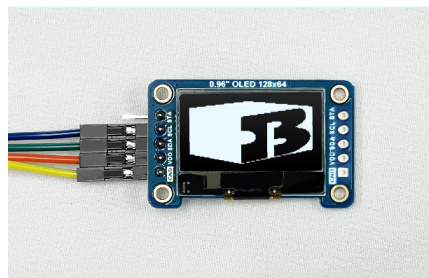
g. test_drawFastHLine_drawFastVLine()

```
void test_drawFastHLine_drawFastVLine(void)
{
    int8_t col, row;
    BMD31.clearDisplay();
    col = 112;
    for (row=0; row<64; row+=8)
    {
        BMD31.drawFastHLine(0, row, col, pixelColor_WHITE);
        BMD31.drawFastVLine(col, row, 64 - row, pixelColor_WHITE);
        col -= 14;
    }
    BMD31.display();
    delay(500);
    BMD31.clearDisplay();
    col = 112;
    for (row=56; row>=0; row-=8)
    {
        BMD31.drawFastHLine(col, row, 128 - col, pixelColor_WHITE);
        BMD31.drawFastVLine(col, row, 64 - row, pixelColor_WHITE);
        col -= 14;
    }
    BMD31.display();
    delay(500);
}
```

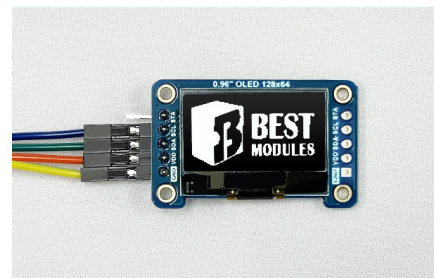


h. test_drawBitmap()

```
void test_drawBitmap(void)
{
    BMD31.clearDisplay();
    BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64, pixelColor_WHITE);
    BMD31.display();
    delay(300);
    BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64, pixelColor_BLACK);
    BMD31.display();
    delay(300);
    BMD31.drawBitmap(0, 0, BestModule_LOGO, 128, 64,
        pixelColor_INVERSE);
    BMD31.display();
    delay(300);
    BMD31.clearDisplay();
    BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
        pixelColor_WHITE);
    BMD31.display();
    delay(300);
    BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
        pixelColor_BLACK);
    BMD31.display();
    delay(300);
    BMD31.drawBitmap(0, 0, BestModule_LOGOandName, 128, 64,
        pixelColor_INVERSE);
    BMD31.display();
    delay(300);
}
```



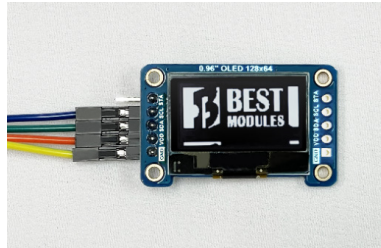
BestModule_LOGO



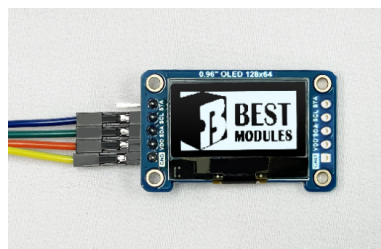
BestModule_LOGOandName

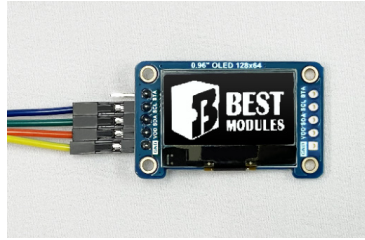
i. test_variousScroll()

```
void test_variousScroll(void)
{
  // 滾動功能測試
  uint8_t startRow = displayROW0;
  uint8_t endRow = displayROW7;
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES);
  delay(500);
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_TOP);
  delay(500);
  BMD31.startScrollRight(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_BOTTOM);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_TOP);
  delay(500);
  BMD31.startScrollLeft(startRow, endRow, SCROLL_2FRAMES,
                        SCROLLV_BOTTOM);
  delay(500);
  BMD31.stopScroll();
}
```

**j. test_invertDisplay()**

```
void test_invertDisplay(void)
{
  BMD31.invertDisplay(TRUE); // 反轉顯示模式：白底黑字
  delay(500);
  BMD31.invertDisplay(FALSE); // 正常顯示模式：黑底白字
  delay(500);
  BMD31.invertDisplay(TRUE); // 反轉顯示模式：白底黑字
  delay(500);
  BMD31.invertDisplay(FALSE); // 正常顯示模式：黑底白字
  delay(500);
}
```

**BMD31.invertDisplay(TRUE)-black-on-white**

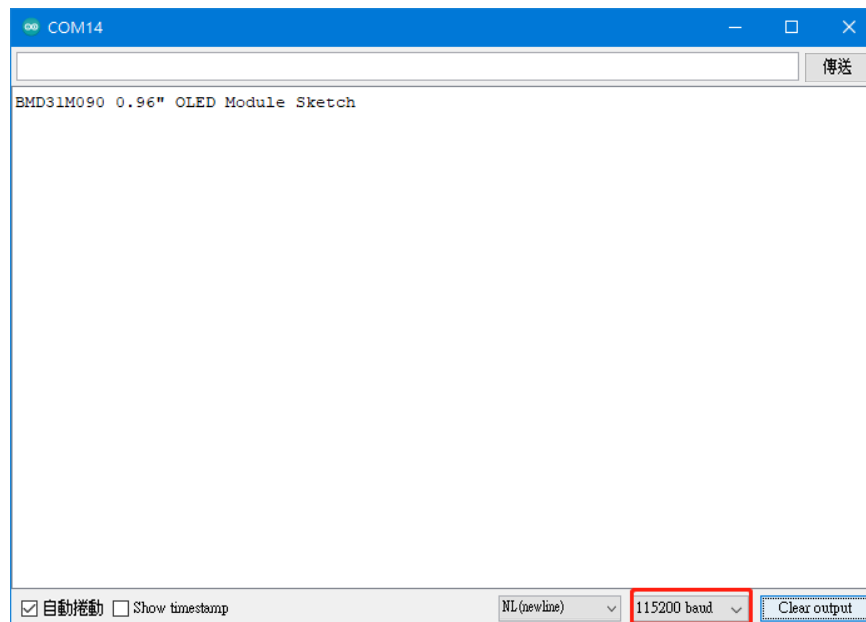


BMD31.invertDisplay(FALSE)-white-on-black

k. test_dim()

```
void test_dim(void)
{
  BMD31.dim(TRUE); // 昏暗模式：對比度 - 0x00
  delay(500);
  BMD31.dim(FALSE); // 正常亮度模式：對比度 - 0xCF
  delay(500);
  BMD31.dim(TRUE); // 昏暗模式：對比度 - 0x00
  BMD31.dim(FALSE); // 正常亮度模式：對比度 - 0xCF
  delay(500);
}
```

3. 打開序列埠監視視窗，串列傳輸速率選擇 115200；序列埠監視視窗顯示如下：



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、範例、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。