



BMduino Keil Driver

快速入门指南

版本：V1.10 日期：2024-12-04

www.bestmodulescorp.com

目录

简介	3
快速入门	3
更换通信接口	6
移植程序指南	7
同时使用多个 Keil Driver	8

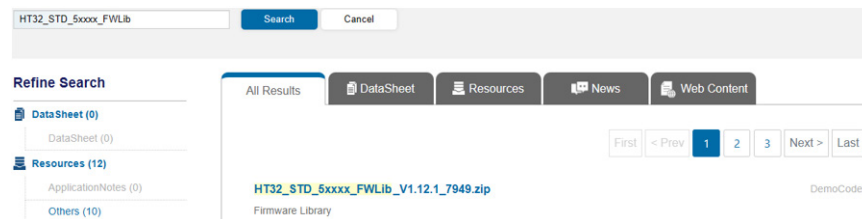
简介

BMduino Keil Driver 是在 Keil 平台使用 BMduino 模块时的驱动库，并带有范例。通过下载和使用 BMduino Keil Driver，可以快速创建 BM53A367A 开发板 (MCU: HT32F52367) 和 ESK32-30501 开发板 (MCU: HT32F52352) 的工程，方便用户在 Keil 平台快速开发 BMduino 模块。下文将以 OLED 模块 (BMD31M090) 为例介绍如何快速使用 Keil Driver，以及介绍如何同时使用多个 Keil Driver。

快速入门

- 步骤 1: 下载 HT32 F/W Library

在 Holtek 官网，搜索“HT32_STD_5xxxx_FWLib”并下载 HT32 F/W Library (注意版本需在 V1.12.1 及以上)



- 步骤 2: 获取 Keil Driver

方式 1 (在线获取): 自行通过倍创官网 <https://www.bestmodulescorp.com/>，搜索对应模块，在“文件”目录下进行下载

方式 2 (线下获取): 咨询倍创获取

eg: BMD31M090 Keil Driver

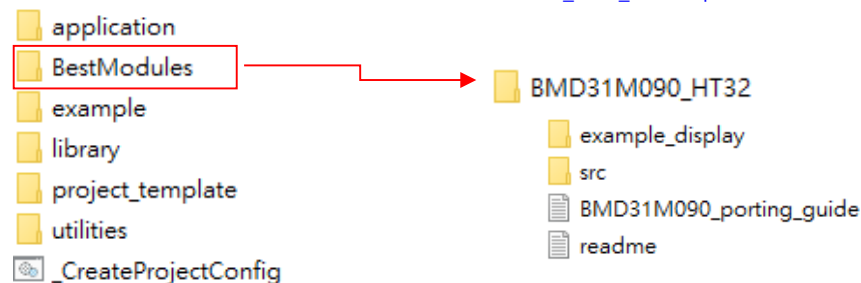
 BMD31M090_HT32_V1.0.1

- 步骤 3: 路径配置

将 Keil Driver 内的 **BMXXXXXXXX_HT32** 目录放入 HT32 F/W Library 的 **BestModules** 文件夹，完成路径配置

[HT32_STD_5xxxx_FWLib_Vm.n.r.v.zip](#)

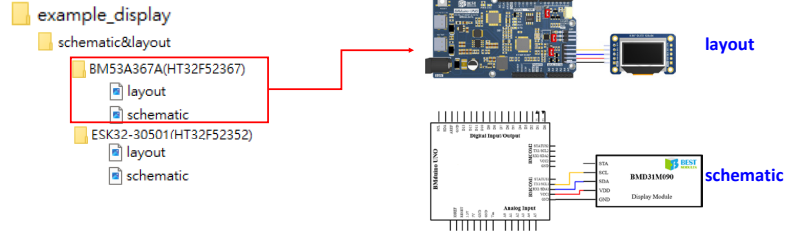
[BMD31M090_HT32_Vm.n.r.zip](#)



● 步骤 4：线路连接

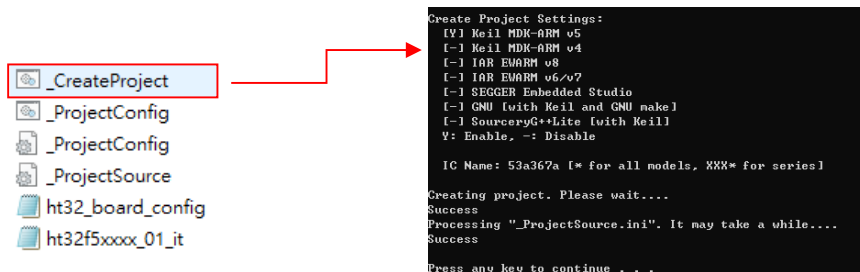
按照范例下的 **schematic & layout** 文件夹里的图片进行开发板和模块的连接

eg: example_display



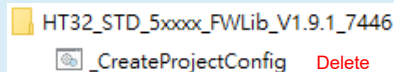
● 步骤 5：创建工程

点击范例中的 “_CreateProject.bat” → 输入 “1” 选择 Keil MDK-ARM v5，再输入 “N” 进入下一步 → 输入 “53a367a” 选择开发板后按 Enter 键 → 等待创建完成，按任意键完成



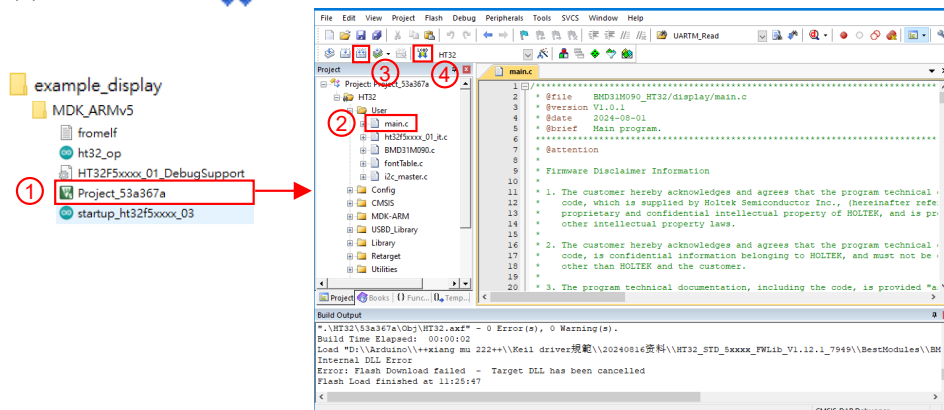
备注：

1. 创建工程：目前 53a367a 和 52352 工程创建后可直接使用。其它请参考“移植程序指南”章节操作。
2. 删除工程：点击范例中的 “_ClearProject.bat” → 输入 “yes” → 等待删除完成，按任意键完成。
3. 更换 MCU / 开发板工程：HT32 F/W Library 会自动保存创建工程的步骤以便下次创建，若需更换 MCU / 开发板工程，可删除 HT32_STD_5xxxx_FWLib_Vm.n.r_v.zip/_CreateProjectConfig.bat 文件，就可以重新选择工程。



● 步骤 6：范例应用

- (1) 打开工程：双击打开 example_XXX\MDK_ARMv5\Project_53a367a.uvprojx
- (2) 选择 main.c 文档
- (3) 编译：点击 “ ” 编译
- (4) 上传：点击 “ ” 上传程序到开发板



● 步骤 7：运行结果查看

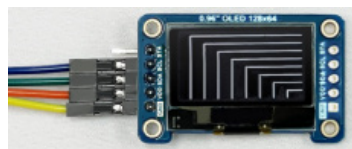
OLED 模块循环演示以下功能，包括显示字符串、画点、画线、画图、滚动、反白显示等。



显示字符串



画点



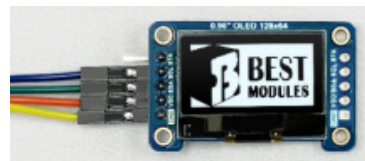
画线



画图



滚动



反白

更换通信接口

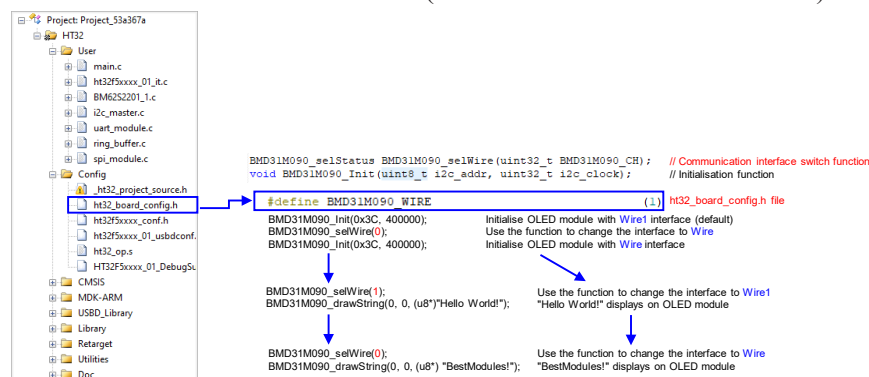
通信接口：默认为 `ht32_board_config.h` 文档下的 `BMXXXXXX_XXX` 宏定义数值

```
#define BMD31M090_WIRE (1)
```

- (1) UART: `BMXXXXXX_SERIAL` (0: Serial 1~4: Serial1~Serial4)
- (2) SPI: `BMXXXXXX_SPI` (0: SPI 1~2: SPI1~SPI2)
- (3) I²C: `BMXXXXXX_WIRE` (0: Wire 1~2: Wire1~Wire2)

更改通信接口：使用函数切换

- (1) UART: `BMXXXXXX_selSerial` (uint32_t serial_number)
(0: Serial 1~4: Serial1~Serial4)
- (2) SPI: `BMXXXXXX_selSPI` (uint32_t spi_number)
(0: SPI 1~2: SPI1~SPI2)
- (3) I²C: `BMXXXXXX_selWire` (uint32_t wire_number)
(0: Wire 1~2: Wire1~Wire2)



原理：定义全局变量 `gBMXXXXXX_XXX`

(如: `gBMXXXXXX_WIRE/gBMXXXXXX_SERIAL/gBMXXXXXX_SPI`)

(1) 默认通信接口

`gBMXXXXXX_XXX` 默认等于 `ht32_board_config.h` 文档下的 `BMXXXXXX_XXX` 宏定义数值

(2) 切换通信接口

使用函数用来改变全局变量 `gBMXXXXXX_XXX` 的数值

移植程序指南

移植程序到其它开发板使用：请阅读 BMXXXXXXX_HT32\XXXXXXX_porting_guide.txt 文档进行操作。

eg: BMD31M090_porting_guide.txt

 BMD31M090_HT32 →  BMD31M090_porting_guide

1. 移植程序到其它 HT32 开发板

- (1) 修改 example_xxxx\ProjectConfig.ini 文档，在 DEVICE_NAME 下添加要移植的 MCU / 开发板型号

```
REM DEVICE_NAME
REM =====
53a367a
52352 → eg: To port the program on the
HT32F52352 development board
```

- (2) 修改 example_xxxx\ht32_board_config.h 文档，添加模块通信接口引脚设定

```
/* USE HT32F52352_SK */
/*
*/
/* communication interface */
#define BMD31M090_WIRE (0)
/* UART Module Setting */
#define HTCFG_UARTM_CH0 USART0
#define HTCFG_UARTM_TX_GPIO_PORT A
#define HTCFG_UARTM_TX_GPIO_PIN 2
#define HTCFG_UARTM_RX_GPIO_PORT A
#define HTCFG_UARTM_RX_GPIO_PIN 3
#define HTCFG_UARTM_TX_BUFFER_SIZE 128
#define HTCFG_UARTM_RX_BUFFER_SIZE 128
/* SPI Module Setting */
#define HTCFG_SPI_CH0 SPI0
#define HTCFG_CH0_SCK_GPIO_PORT B
#define HTCFG_CH0_SCK_GPIO_PIN 3
#define HTCFG_CH0_SEL_GPIO_PORT B
#define HTCFG_CH0_SEL_GPIO_PIN 2
#define HTCFG_CH0_MOSI_GPIO_PORT B
#define HTCFG_CH0_MOSI_GPIO_PIN 4
#define HTCFG_CH0_MISO_GPIO_PORT B
#define HTCFG_CH0_MISO_GPIO_PIN 5
/* I2C Master Setting */
#define HTCFG_I2CM_CH0 I2C1
#define HTCFG_MO_SCL_GPIO_PORT A
#define HTCFG_MO_SCL_GPIO_PIN 0
#define HTCFG_MO_SDA_GPIO_PORT A
#define HTCFG_MO_SDA_GPIO_PIN 1
#endif
```

eg: Add the HT32F52352 development board used communication interface pin settings.

2. 移植程序到其它厂商的 MCU

- (1) 解压 Keil Driver “XXXXXXX_HT32_V1.0.1.zip”

eg: BMD31M090 Keil Driver

 BMD31M090_HT32_V1.0.1

- (2) 将 src 目录下的 “.c” 和 “.h” 文档复制到其它 MCU 工程的驱动库文件夹下
- (3) 将 “.c” 和 “.h” 文档添加到所需工程，完成路径配置
- (4) 修改需要修改的底层函数

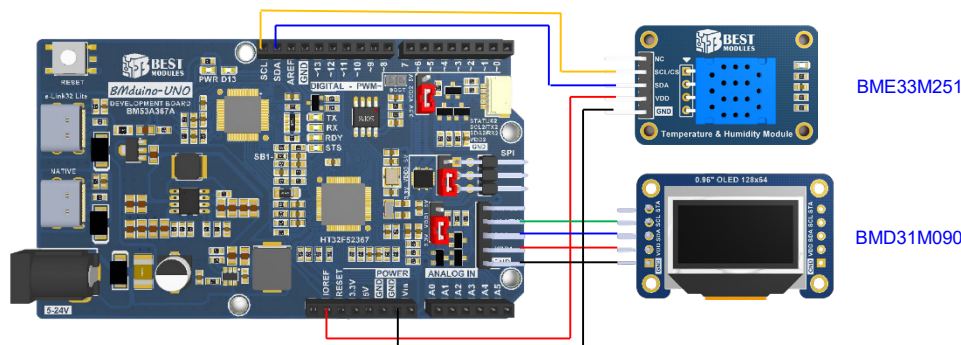
 BMD31M090_porting_guide

```
Update I2C initial function
Update I2C master write function
modification :
static void setCommand(uint8_t command)
static void setData(uint8_t command)
```

备注：移植文档内只描述驱动文档 (.h /.c) 文档，未描述 main.c 文档，因此 main.c 有应用到的一些功能 (串口监视器，延时函数等)，如果用户需要使用，移植的时候也需要重写。

同时使用多个 Keil Driver

绝大部分使用场景是需要使用到多个模块，也就是需要同时使用到多个 Keil Driver。以下将以读取温湿度检测模块的温湿度数值，显示在 OLED 模块为例，讲解如何同时使用多个 Keil Driver。线路连接如图：



● 步骤 1：获取 Keil Driver

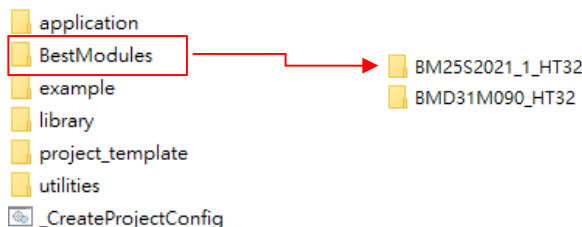
获取两个模块的 Keil Driver：

模块	Keil Driver
温湿度检测模块 BME33M251	BM25S2021_1_HT32
OLED 模块 BMD31M090	BMD31M090_HT32

● 步骤 2：路径配置

将 Keil Driver 内的 **BMXXXXXXX_HT32** 目录放入 HT32 F/W Library 的 **BestModules** 文件夹，完成路径配置

[HT32_STD_5xxx_FWLib_Vm.n.r_v.zip](#)



● 步骤 3：范例创建

范例创建，可以在其它工程，也可以选择复制 Keil Driver 的一个范例出来修改，这里以复制 BMD31M090_HT32/example_display 范例出来修改为例。

(1) 复制 BMD31M090_HT32/example_display 范例

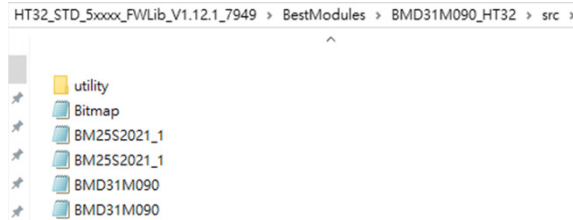
(2) 将范例名称改为想要的名称，如 example_displayTemperatureAndHumidity

HT32_STD_5xxx_FWLib_V1.12.1_7949 > BestModules > BMD31M090_HT32



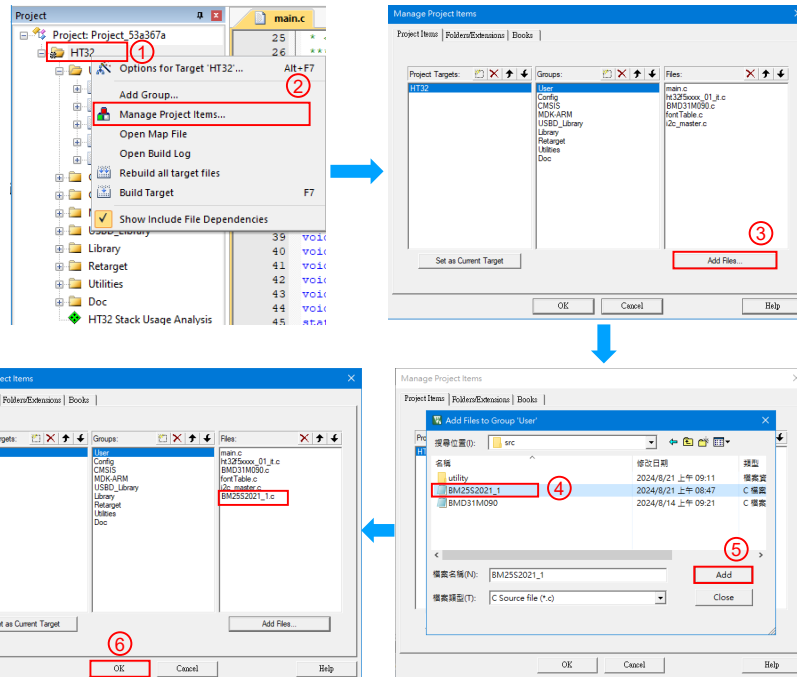
(3) 创建 53a367a 工程，并打开（具体方式参考“快速入门”章节）

- 步骤 4: 添加 Keil Driver 的 “BMXXXXXX.c” , “BMXXXXXX.h” 到工程。
 - (1) 将 BM25S2021_1_HT32/src 目录下的 BM25S2021_1.c/h 档复制到 BMD31M090_HT32/src 目录下



备注: 因为本次演示是使用 BMD31M090_HT32 范例, 所以 BMD31M090.c/h 档不用重新添加, 只需要添加 BM25S2021_1.c/h 档。若是在其它工程使用这两个 Keil Driver, 则两个 Keil Driver 的 .c/.h 档都要添加。

- 步骤 5: 添加 .c 档
 - (1) 在 Keil IDE, 右击 “HT32”, 选择 “Manage Project Items...”
 - (2) 在弹窗点击 “Add Files...”
 - (3) 在弹出的窗口选择 BMD31M090_HT32/src 目录的 BM25S2021_1.c 档, 点击 “Add” 后, 关掉弹窗
 - (4) 确认清单已经添加完成后, 点击 “OK”, 即可完成 .c 档添加

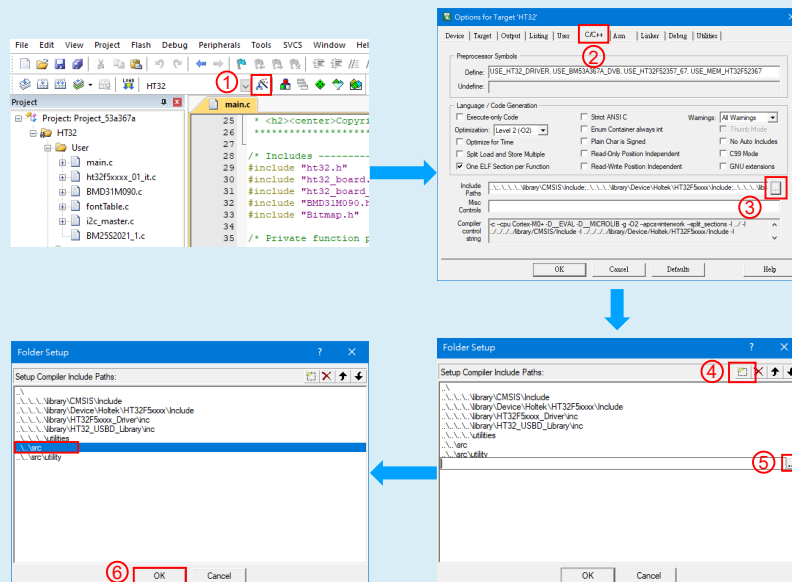


备注: 因为本次演示是使用 BMD31M090_HT32 范例, 所以 BMD31M090.c 档不用重新添加, 只需要添加 BM25S2021_1.c 档。若是在其它工程使用这两个 Keil Driver, 则两个 Keil Driver 的 .c 档都要添加。

- 步骤 6: 添加 .h 路径
路径已添加，本次演示无需进行此操作。

备注:

1. 因为本次演示是使用 BMD31M090_HT32 范例，且 BM25S2021_1.h 档的位置同 BMD31M090.h 一致，都在 BMD31M090_HT3/src，所以不用重新添加 .h 路径。
2. 若是在其它工程使用这两个 Keil Driver，则两个 Keil Driver 的 .h 路径都要添加，以下为在其它工程的操作过程：
 - (1) 点击 “Options for Target...” 图标
 - (2) 点击弹窗上面菜单的 “C/C++”，点击 “Include Paths” 右侧的 “...” 按键
 - (3) 选择 “New(Insert)” 新增一个路径框后，选择路径框右侧的 “...” 按键，选择使用到的 BMXXXXXX.h 档所在的目录
 - (4) 确认已成功添加 .h 路径后，点击 “OK”，完成 .h 路径添加



- 步骤 7: 撰写 ht32_board_config.h 档
此文档用于配置模块的通信引脚，具体可学习各个模块 Keil Driver 对应的范例。本演示中模块的通信接口如下：

模块	通信方式	通信接口
BME33M251	I ² C 通信	Wire 接口
BMD31M090	I ² C 通信	Wire1 接口

因此 ht32_board_config.h 档配置通信接口如下：

```

/*-----*/
/* USE_BM53A367A_DVB */
/*-----*/
#define BM25S2021_1_WIRE (0)
#define BMD31M090_WIRE (1)

/* BM25S2021_1 Data Pin(One-wire communication) -----*/
#define BM25S2021_1_DATAPIN_GPIO STRCAT2(P, C)
#define BM25S2021_1_DATAPIN_GPIO_ID STRCAT2(GPIO_P, C)
#define BM25S2021_1_DATAPIN_AFIO_PIN STRCAT2(AFIO_PIN_, 13)
#define BM25S2021_1_DATAPIN_GPIO_PORT STRCAT2(HT_GPIO, C)
#define BM25S2021_1_DATAPIN_GPIO_PIN STRCAT2(GPIO_PIN_, 13)

```

- 步骤 8：整合 ht32f5xxxx_01_it 档

此文档是中断服务函数的内容，因此需要将两个模块的 Keil Driver 的 ht32f5xxxx_01_it 档内容使用到的中断服务函数整合到一起。

```
extern vu32 BM25S2021_1_SYSTICK_DelayTime;
extern vu8 BM25S2021_1_timeFlag;
extern vu32 BM25S2021_1_timeCount;
void SysTick_Handler(void)
{
    if(BM25S2021_1_SYSTICK_DelayTime != 0)
    {
        BM25S2021_1_SYSTICK_DelayTime--;
    }
    if(BM25S2021_1_timeFlag != 0)
    {
        BM25S2021_1_timeCount ++;
        if(BM25S2021_1_timeCount == 0xffffffff)
        {
            BM25S2021_1_timeFlag = 0;
            BM25S2021_1_timeCount = 0;
        }
    }
}
```

- 步骤 9：根据应用撰写 main.c 档

(1) 引用使用到的 Keil Driver 的 .h 档

(2) 撰写实现程序：初始化模块后，根据需求调用对应 Keil Driver 函数实现功能。

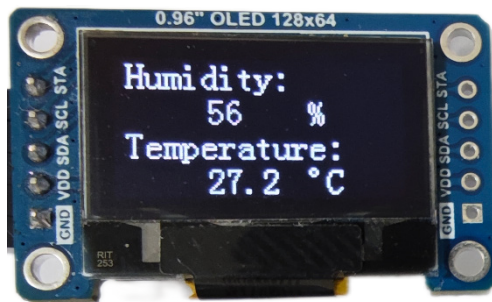
```
#include "BM25S2021_1.h"
#include "BMD31M090.h"
#include "Bitmap.h"
static void __Delay_ms(vu32 count);
void displayFloatNum(uint8_t x, uint8_t row, float floatNum,
    int decimalPlaces);
float humidity;
float temperature;
int main(void)
{
    BM25S2021_1_I2C_Init(); // initialization BME33M251 Module
    BMD31M090_Init(BMD31M090_DEVICEADDR0,BMD31M090_CLOCK_SPEED);
    // initialization BMD31M090 Module
    /* Display the initial interface */
    BMD31M090_clearDisplay();
    BMD31M090_drawPixel(86,50,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(87,50,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(85,51,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(88,51,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(85,52,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(88,52,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(86,53,BMD31M090_pixelColor_WHITE);
    BMD31M090_drawPixel(87,53,BMD31M090_pixelColor_WHITE);
    BMD31M090_display();
    BMD31M090_drawString(85, 2, (u8*)"");
    BMD31M090_drawString(93, 6, (u8*)"C");
    BMD31M090_setFont(FontTable_8X16); // Set Font Type:8X16
    BMD31M090_drawString(5, 0, (u8*)"Humidity:");
    BMD31M090_drawString(5, 4, (u8*)"Temperature:");
}
```

```
while(1)
{
    humidity = BM25S2021_1_readHumidity();
    BMD31M090_drawNum(40, 2, humidity, 2); // display humidity on OLED
    temperature = BM25S2021_1_readTemperature(false);
    displayFloatNum(40, 6, temperature, 1); // display temperature on OLED
    __Delay_ms(2000);
}
}

void displayFloatNum(uint8_t x, uint8_t row, float floatNum,
int decimalPlaces)
{
    int decimalNum = 0;
    int baseTemp = 1;
    int decimalPlacesTemp = decimalPlaces;
    while (decimalPlacesTemp != 0)
    {
        baseTemp *= 10;
        decimalPlacesTemp--;
    }
    BMD31M090_drawNum(x, row, (uint32_t)floatNum, 2);
    BMD31M090_drawString(x+16, row, (u8*)"");
    decimalNum = (int)(floatNum * baseTemp) % baseTemp;
    BMD31M090_drawNum(x+24, row, (uint32_t)decimalNum, decimalPlaces);
}

static void __Delay_ms(vu32 count)
{
    count = SystemCoreClock / 8000 * count;
    while(count--);
}
```

(3) 运行结果查看



Copyright® 2024 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。倍创不担保任何明示、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。