BMduino Keil Driver

Quick Start Guide

Revision: V1.10 Date: December 04, 2024

www.bestmodulescorp.com

Table of Contents

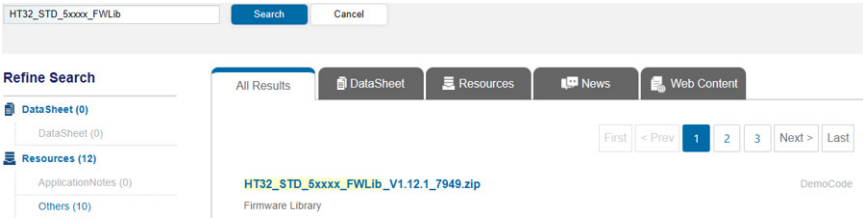
Introduction

The BMduino Keil Driver is a driver library with examples on the Keil platform when using BMduino modules. By downloading and using the BMduino Keil Driver, users can quickly create the development board projects of BM53A367A and ESK32-30501, respectively corresponding to their master MCUs, HT32F52367 and HT32F52352. This is convenient for users to quickly develop BMduino modules on the Keil platform. The following section will take the OLED module (BMD31M090) as an example to explain how to use a Keil Driver quickly and how to use multiple Keil Drivers simultaneously.

Quick Start

- Step 1: Download the HT32 FW Library

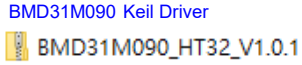
  Search the HT32 STD 5xxxx FWLib and download the HT32 FW Library V1.12.1 or later on the Holtek website.



- Step 2: Get a Keil driver

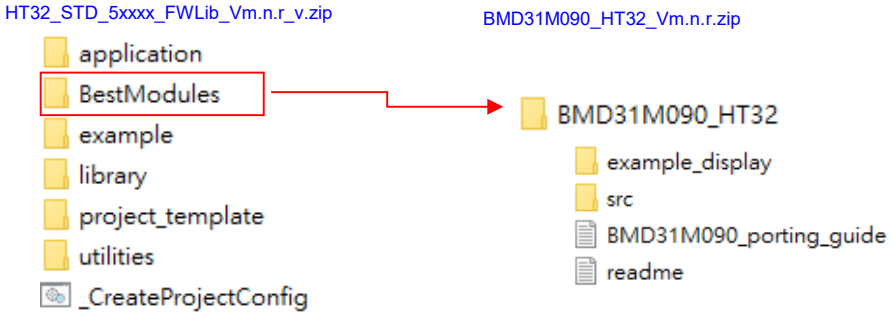
  Method 1 – Get it online: search the corresponding module on the Best Modules website https://www.bestmodulescorp.com/ and download it from the “DOCUMENTS” directory.

  Method 2 – Get it Offline: consult the BEST MODULES to obtain.



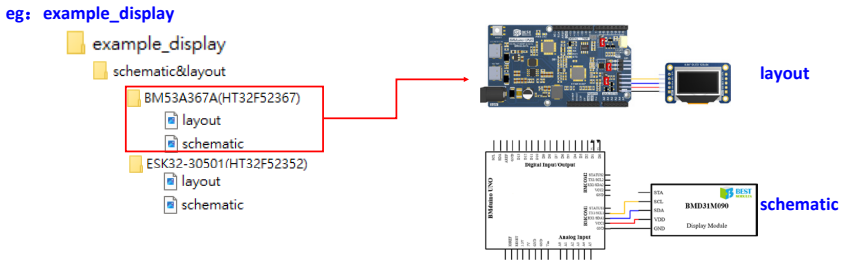
- Step 3: Configure a path

  Place the BMXXXXXXXXX_HT32 directory in the Keil Driver into the BestModules folder of the HT32 F/W Library to complete the path configuration.



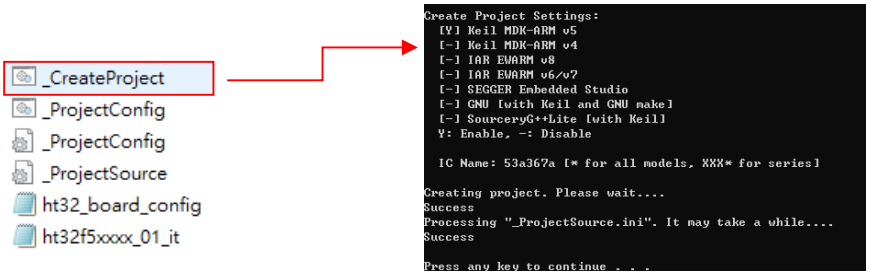
- Step 4: Wire connection

  Connect the development board to the module according to the schematic & layout diagrams in the example.



- Step 5: Create a project

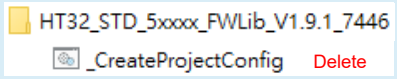
  Click "_CreateProject.bat" in the example → Enter "1" to select Keil MDK-ARM v5, then enter "N" to enter the next step → Enter "53a367a" to select the development board and press "Enter" → Waiting for the project to be created and press any key to finish.



Note: 1. Create a project: the projects can be used directly when the 53A367A and the 52352 projects have been created. For details, see the Guide for Porting Programs section.

2. Delete a project: click "_ClearProject.bat" in the example → Enter "yes" → Wait for the deletion to complete and press any key to finish.

3. Change the MCU or development board project: HT32 F/W Library will automatically save the steps of creating the project for the next project creation. When users need to change the MCU or development board project, delete the HT32_STD_5xxxx_FWLib_Vm.n.r_v.zip/_CreateProjectConfig.bat file to reselect a project.



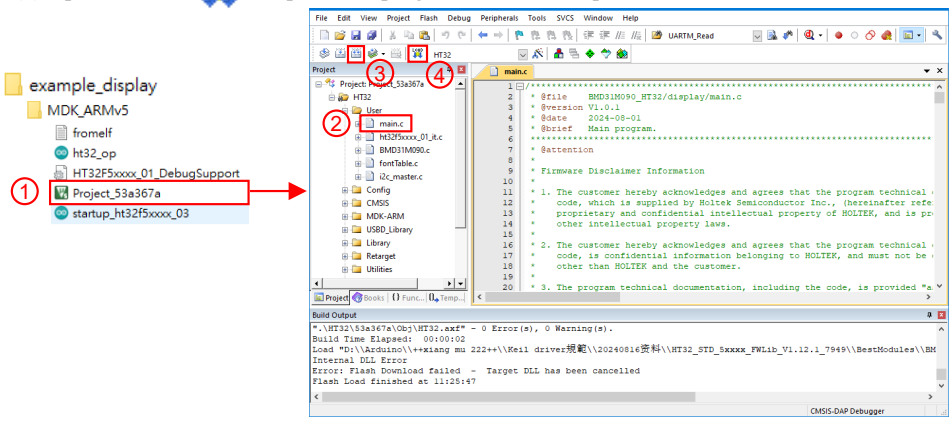
- Step 6: Example applications

  (1) Open File: double-click to open the example_XXX\MDK_ARMv5\Project_53a367a.uvprojx

  (2) Select the main.c file

  (3) Compile: Click “ ” to compile

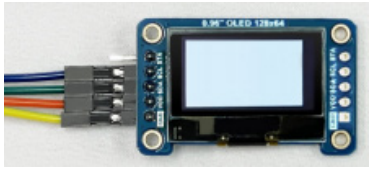
  (4) Upload: Click “LOAD” to upload the program to the development board



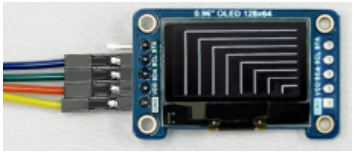
- Step 7: View an operation result

  The OLED module circularly demonstrates the following functions of string display, drawing pixels, drawing lines, drawing bitmaps, picture scrolling and invert display.

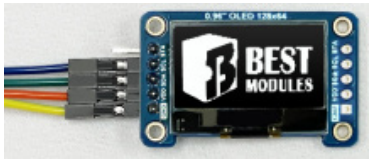
String Display



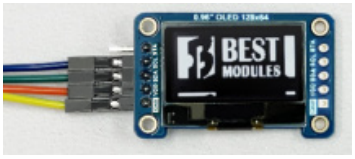
Drawing Pixels



Drawing Lines



Drawing Bitmaps



Scrolling

Invert Display

Change the Communication Interface

Communication interface: defaults to the BMXXXXX_XXX macro definition value in ht32_board_config.h file

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

(1) UART: BMXXXXX_SERIAL (0: Serial 1~4: Serial1~Serial4)

(2) SPI: BMXXXXX_SPI (0: SPI 1~2: SPI1~SPI2)

(3) I²C: BMXXXXX_WIRE (0: Wire 1~2: Wire1~Wire2)

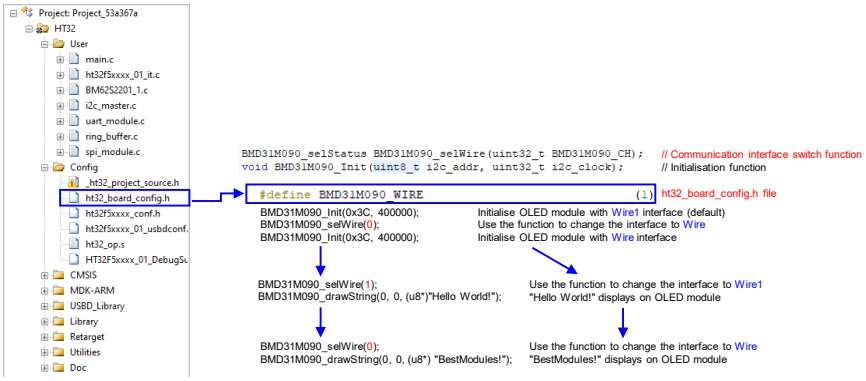
Change the communication interface using functions

(1) UART: BMXXXXX_selSerial (uint32_t serial_number)

(0: Serial 1~4: Serial1~Serial4)

(2) SPI: BMXXXXX_selSPI(uint32_t spi_number) (0: SPI 1~2: SPI1~SPI2)

(3) I²C: BMXXXXX_selWire(uint32_t wire_number) (0: Wire 1~2: Wire1~Wire2)



Principle: define a global variable, gBMXXXXX_XXX

(such as gBMXXXXX_WIRE/gBMXXXXX_SERIAL/gBMXXXXX_SPI)

(1) Default communication interface

gBMXXXXX_XXX: defaults to the BMXXXXX_XXX macro definition value in the ht32_board_config.h file

(2) Change the communication interface

Use a function to change the value of the global variable gBMXXXXX_XXX

Guide for Porting Programs

Port programs to other development boards: refer to the BMXXXXXXX_HT32\XXXXXXX_porting_guide.txt file.

eg: BMD31M090_porting_guide.txt

1. Port programs to other HT32 development boards

(1) Modify the example_xxxxx_ProjectConfig.ini file to add the MCU or development board model to be ported under DEVICE_NAME

```
REM DEVICE_NAME
REM ═══════════════════
53a367a
52352
```

eg: To port the program on the HT32F52352 development board

(2) Modify the example_xxxxx \ ht32_board_config.h file to add module communication interface pin settings

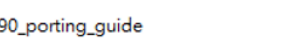
```
/*-------------------------------------------------------------------------------------------------------*/
/* USE_HT32F52352_SK                                                                                     */
/*-------------------------------------------------------------------------------------------------------*/
#if defined(USE_HT32F52352_SK)
/* communication interface-----------------------------------------------------------------------------*/
  #define BMD31M090_WIRE                        (0)
/* UART Module Setting---------------------------------------------------------------------------------*/
  #define HTCFG_UARTM_CH0                       USART0
  #define HTCFG_UARTM0_TX_GPIO_PORT             A
  #define HTCFG_UARTM0_TX_GPIO_PIN              2
  #define HTCFG_UARTM0_RX_GPIO_PORT             A
  #define HTCFG_UARTM0_RX_GPIO_PIN              3
  #define HTCFG_UARTM0_TX_BUFFER_SIZE           128
  #define HTCFG_UARTM0_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_M0_SCL_GPIO_PORT                A
  #define HTCFG_M0_SCL_GPIO_PIN                 0
  #define HTCFG_M0_SDA_GPIO_PORT                A
  #define HTCFG_M0_SDA_GPIO_PIN                 1
#endif
```

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

2. Port programs to other manufacturers' MCUs

(1) Decompress the Keil driver "XXXXXX_HT32_V1.0.1.zip"

eg：BMD31M090 Keil Driver



(2) Copy the ".c" and ".h" files from the src directory to the driver folders in other MCU projects

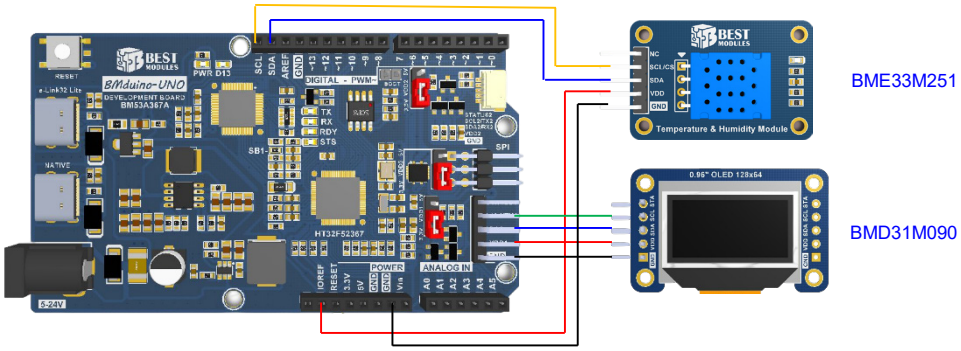
(3) Add the ".c" and ".h" files into the desired project to complete the path configuration

(4) Modify the underlying function that requires to be modified

Note: The port file only describes the .h and .c driver files, not the main.c file. Therefore, the main.c has some functions (serial monitor, delay function, etc.) that should be rewritten when porting if users use them.

Multiple Keil Drivers Used Together

Most of the scenarios require multiple modules, which means multiple Keil Drivers are used at the same time. The following is an example of how to use multiple Keil Drivers at the same time by reading the temperature and humidity values from the temperature and humidity detection module and displaying them on the OLED module. The wires are connected as shown in the figure.



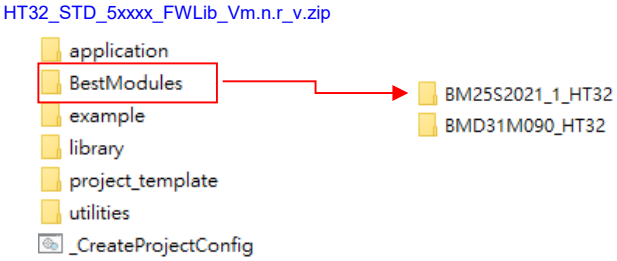
- Step 1: Get the Keil Driver

  Get the Keil Drivers for both modules.

Module	Keil Driver
Temperature & Humidity Module – BME33M251	BM25S2021_1_HT32
OLED Display Module – BMD31M090	BMD31M090_HT32

- Step 2: Configure the path

  Place the BMXXXXXXXXX_HT32 directory from the Keil Driver into the BestModules folder in the HT32 F/W Library to complete the path configuration.

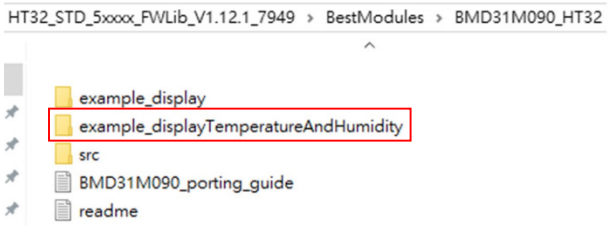


- Step 3: Create an example

  Users can create an example in other projects and also can copy an example from Keil Driver to modify. Here is an example of copying the BMD31M090_HT32/example_display example to modify.

  (1) Copy an example, the BMD31M090_HT32/example_display

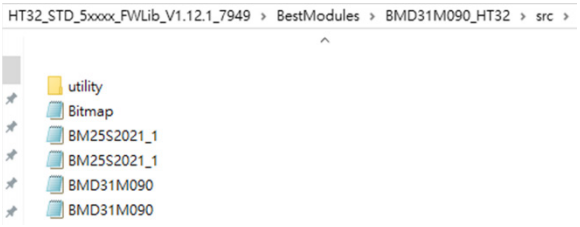
  (2) Change the example name to a desired name such as example_displayTemperatureAndHumidity



  (3) Create a project named 53a367a and open it. Refer to the “Quick Start” section for details.

- Step 4: Add the “BMXXXXXXXX.c” and “BMXXXXXX.h” files of Keil Driver to the project.

  (1) Copy the BM25S2021_1.c/.h files from the BM25S2021_1_HT32/src directory to the BMD31M090_HT32/src directory.



> Note: Since this demonstration uses the BMD31M090_HT32 example, the BMD31M090.c/.h files do not need to be re-added, only the BM25S2021_1.c/.h files need to be added. If the two Keil drivers are used in other projects, the .c/.h files of both Keil drivers must be added.

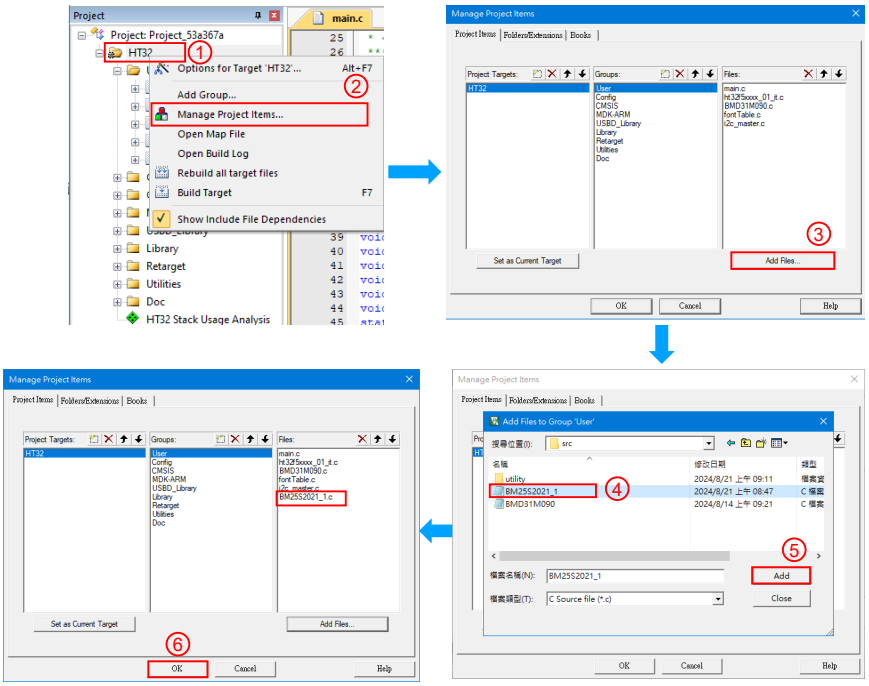
- Step 5: Add the .c file

  (1) In Keil IDE, right-click “HT32” to select “Manage Project Items...”.

  (2) Click “Add Files...” in the pop-up window.

  (3) In the pop-up window, select the BM25S2021_1.c file into the BMD31M090_HT32/src directory. Click “Add” and close the pop-up window.

  (4) After confirming that the C file in the list has been added, click “OK” to complete the addition of the .c file.



> Note: Since this demonstration uses the BMD31M090_HT32 example, the BMD31M090.c file does not need to be re-added, only the BM25S2021_1.c file needs to be added. If the two Keil drivers are used in other projects, the .c file of both Keil drivers must be added.

- Step 6: Add the .h path

  This is not necessary for this demo because the path has already been added.

  Note: 1. This demonstration uses the BMD31M090_HT32 example. The location of the BM25S2021_1.h file is the same as that of the BMD31M090.h file, which is in the BMD31M090_HT32/src. Therefore there is no need to re-add the .h. path.

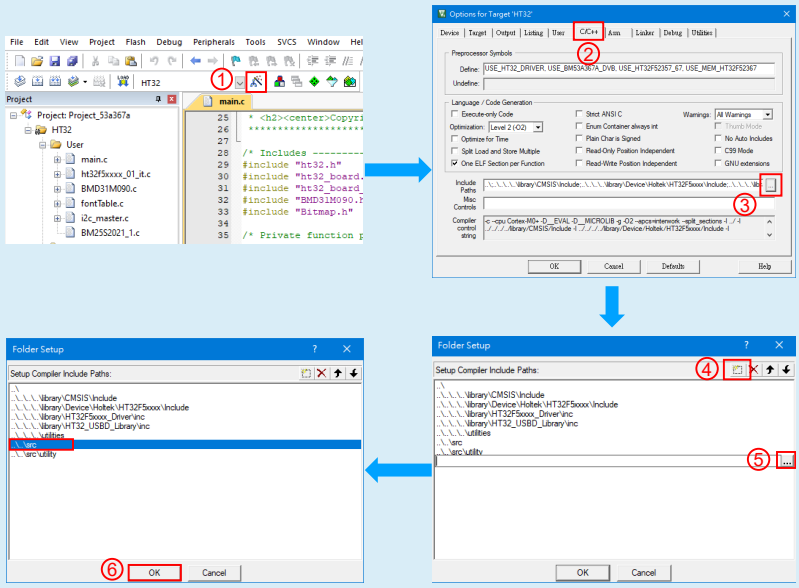
  2. If the two Keil drivers are used in other projects, the.h paths of both Keil drivers must be added. The following describes the process of adding paths to other projects:

  (1) Click on the “Options for Target...” icon.

  (2) Click “CC++” in the pop-up window and click the “...” button on right of “Include Paths”

  (3) Select “New(Insert)” to add a new path box and click the “...” button on the right side of the path box. Then select the directory where the BMXXXXXX.h file is located.

  (4) After confirming that the .h path has been successfully added, click “OK” to complete the addition of the .c file.



- Step 7: Write the ht32_board_config.h file

  This file is used to configure the module communication interface pins. For details, refer to the corresponding Keil driver example for each module. The module communication interface settings in this demonstration are as follows:

Module	Communication Method	Communication Interface
BME33M251	I²C	Wire
BMD31M090	I²C	Wire1

The communication interface is configured in the ht32_board_config.h file as follows:

```
/*-------------------------------------------------------------------------------------------------------*/
/* 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)
```

- Step 8: Integrate the ht32f5xxxx_01_it file

  This file includes the interrupt service functions. The interrupt service functions used in the ht32f5xxxx_01_it file of the Keil Driver for both modules must be integrated.

```
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;
    }
  }
}
```

- Step 9: Write the main.c file according to applications

  (1) Include the used.h file of Keil driver

  (2) Programming: when the module is initialised, call the corresponding Keil driver function to implement a certain function as required

```
#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) View the operation result