GigaDevice Semiconductor Inc.

GD32F103xx
ARM® Cortex™-M3 32-bit MCU

Application Note
AN001

Table of Contents

List of Figures

1   Introduction

The objective of this application note is to explain how to migrate the software from different MCU platform to GD32F103xx. The purpose of this document is not to provide detailed information on the devices, but to highlight the differences between them.

2   Software migration description

After system reset, the HSI oscillator is selected as system clock. If the HSE oscillator is used as the system clock, the switch from HSI to HSE occurs automatically. When the HSE is powered up, it will not be released for use until the HSE crystal oscillator is stable. This specific delay period is known as the oscillator "Start-up time". As the HSE becomes stable, an interrupt will be generated if the related bit in the register is set. At this point the HSE clock can be used directly as the system clock source or the PLL input clock. If HSE is not ready or a failure is detected on the HSE clock, the switch of system clock will not occur and the system clock will switch to HSI oscillator.

Since the HSE oscillator startup time is different between the MCU platforms, the HSE startup timeout value should be changed in the device specific header file (device.h) when use GD32F103xx.

There is also difference of MCU Flash programming algorithm between the MCU platforms, so you should use the relevant IDE project settings based on GD32F103xx for application.

3   Software migration examples

3.1   Keil IDE example

In order to make Keil RealView MDK project work properly with GD32F103xx chip, two modifications need to be made:

■   First, the HSE_STARTUP_TIMEOUT, defined in device peripheral access layer header file *"__32f10x.h"*, need to be changed to the value 0xFFFF. That is

```
#define HSE_STARTUP_TIMEOUT   ((uint16_t)0x0500)
Changed to
#define HSE_STARTUP_TIMEOUT   ((uint16_t)0xFFFF)
```

Figure 1. HSE_STARTUP_TIMEOUT change in Keil IDE

■ Secondly, extract the "*GD32F103xx KEIL IDE Config.rar*" file to the "Keil/arm/flash" folder, and add the new flash programming algorithm to the project, as illustrated in the below screenshots.

Figure 2. Flash programming algorithm change in Keil IDE
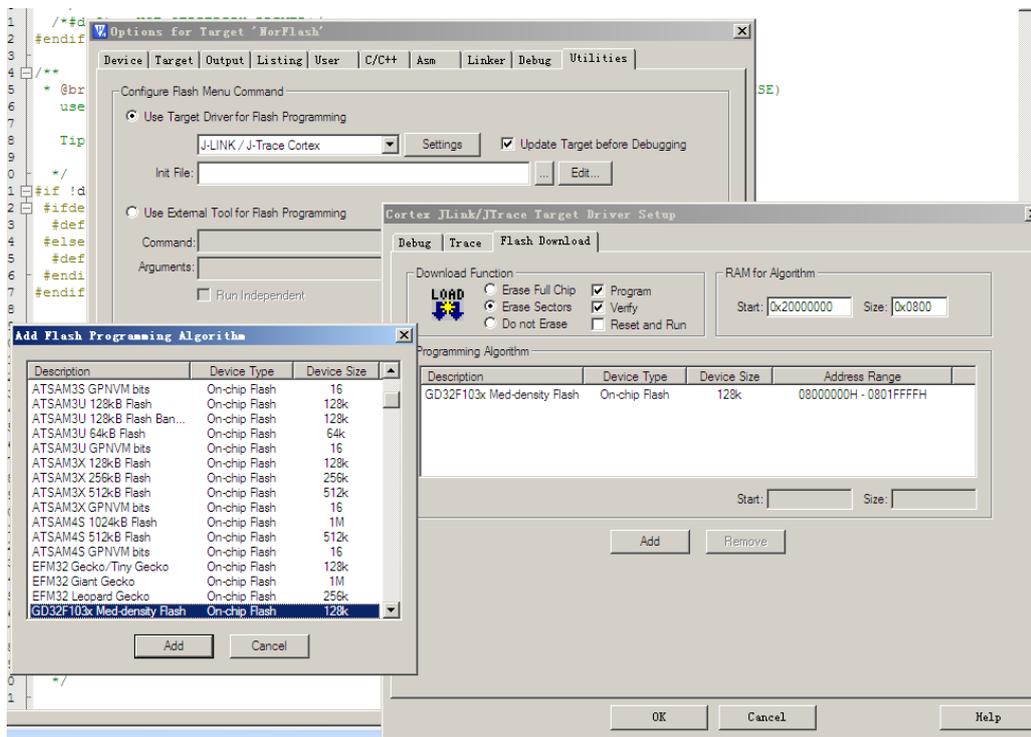
3.2   IAR IDE example

In order to make IAR EWARM project work properly with GD32F103xx chip, two modifications need to be made:

■ First, the HSE_STARTUP_TIMEOUT, defined in device peripheral access layer header file "*__32f10x.h*", need to be changed to the value 0xFFFF. That is

```
#define HSE_STARTUP_TIMEOUT   ((uint16_t)0x0500)
```
Changed to
```
#define HSE_STARTUP_TIMEOUT   ((uint16_t)0xFFFF)
```

Figure 3. HSE_STARTUP_TIMEOUT change in IAR IDE

```
//#define HSE_STARTUP_TIMEOUT    ((uint16_t)0x0500) /*!< Time out for HSE start up */
#define HSE_STARTUP_TIMEOUT    ((uint16_t)0xFFFF) /*!< Time out for HSE start up */
```

■   Secondly, extract the "*GD32F103xx IAR IDE Config.rar*" file to the "IAR/arm/config" folder, and select relevant device, as illustrated in the below screenshots.

Figure 4. Device selection in IAR IDE
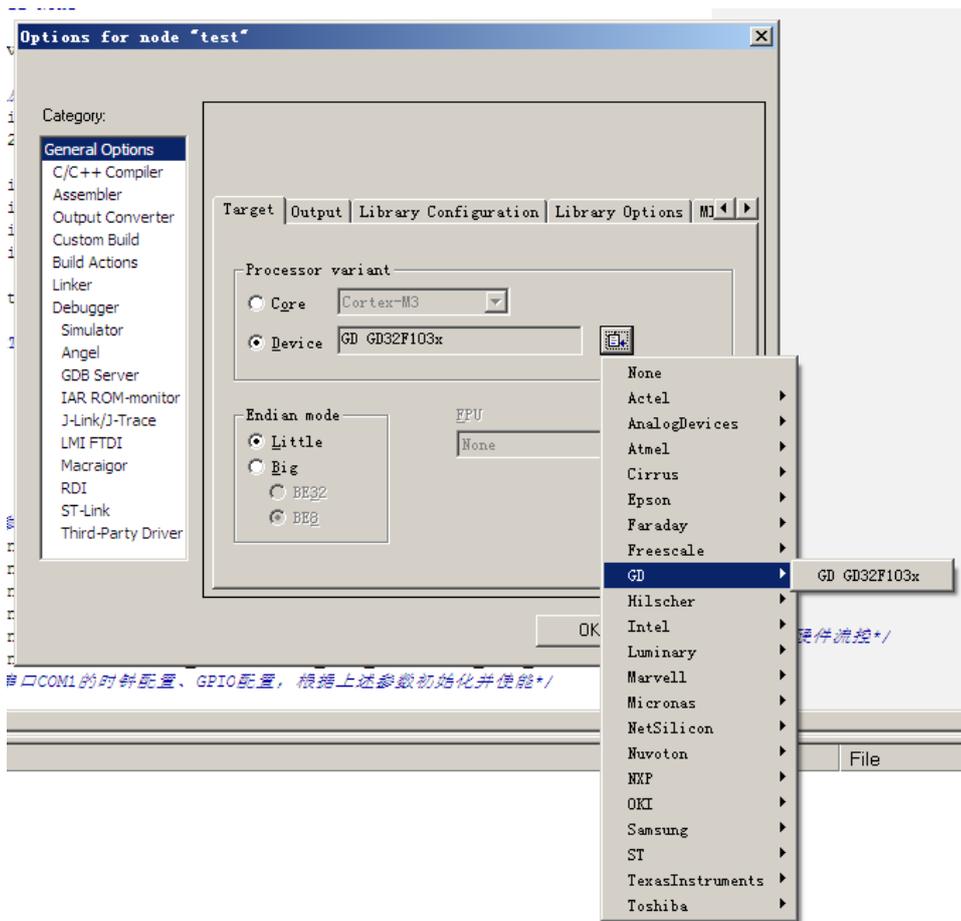
It is recommend to use the link driver in the latest version of IAR software install pack.

4   Revision history

Table 1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jan.14, 2013
1.1	Add IAR IDE support	Apr. 7, 2013