GigaDevice Semiconductor Inc.

GD32F103xx ARM[®] Cortex[™]-M3 32-bit MCU

Application Note AN001



Table of Contents

List c	List of Figures2				
1	Introduction	. 3			
2	Software migration description	. 3			
3	Software migration examples	. 3			
3.1	Keil IDE example	3			
3.2	IAR IDE example	4			
4	Revision history	.6			



List of Figures

Figure 1. HSE STARTUP TIMEOUT change in Keil IDE	4
Figure 2. Flash programming algorithm change in Keil IDE	4
Figure 3. HSE_STARTUP_TIMEOUT change in IAR IDE	5
Figure 4. Device selection in IAR IDE	5



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.



1 /*#d 2 #endif 3 - 4 = /** 5 * @br 6 use 7 8 Tip 9 0 - */ 1 = #if !d 2 = #ifde	V Options for Device Target Configure Flat © Use Targ Int File	Turget 'Norf' Uutput Listin sh Menu Command - et Driver for Flash Pr J-LINK/J-Trace (: mal Taal for Flash Pr	Lash' ng User C, ogramming Cortex	ic++ ▼	Asm Linker Debug Utilities Settings ✓ Update Target before Debugging Edt	
3 #def	Use Exte		ogramming	Lort	rtex JLink/Jirace larget Briver Setup	<u>~</u>
4 #else	Command	d:		De	Debug Trace Flash Download	
5 #def	Arguments	r [1
6 - #endi	reguinerite	1		- I r	Download Function RAM for Algorithm	
7 #endif		Run Independe	ent		LOAD Class Full Unip M Program	
					C Do not Erase Reset and Run	
Add Flash P	rogramming A	lgorithm		×	X	
				_	Programming Algorithm	
Description		Device Type	Device Size		Description Device Type Device Size Address Range	
ATSAM3S G	PNVM bits 29kB Flach	On-chip Flash	16 128k		GD32F103x Med-density Flash On-chip Flash 128k 08000000H - 0801FFFFH	
ATSAM3U 1	28kB Flash Ban	On-chip Flash	128k	_		
ATSAM3U 6	4kB Flash	On-chip Flash	64k			
ATSAM3U G	PNVM bits	On-chip Flash	16			
ATSAM3X 1	28kB Flash	On-chip Flash	128k			
ATSAM3X 2	56kB Flash	On-chip Flash	256k			
ATSAM3X 5	IZKB Hash	On-chip Hash	512k		Start: Size:	
ATSAM3X G	024kB Flash	On-chip Flash	16 1M		,	
ATSAM4S 5	12kB Flash	On-chip Flash	512k		Add Remove	
ATSAM4S G	PNVM bits	On-chip Flash	16		Auu neilluve	
EFM32 Geck	ko/Tiny Gecko	On-chip Flash	128k			
EFM32 Gian	t Gecko	On-chip Flash	1M			
EFM32 Leop	ard Gecko	Un-chip Flash	256k			
i juguazi iluax	Med-density Flash	On-chip Hash	126K	Ľ		
8	Add	Cancel				
5 */						
1 -						
				-	OK Cancel Help	

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 0xFFF. 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



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

Options for node "test"						
i Category:						
2 General Ontions						
C/C++ Compiler						
i Assembler						
1 Output Converter Target Output Library Configuration Libra	ry Options MI					
i Custom Build						
¹ Build Actions Processor verient						
Linker						
C Cortex-M3						
Simulator GD GD32F103x						
Angel						
GDB Server	None					
IAR ROM-monitor	Actel					
	AnalogDevices •					
Macraigor O Bi a	Atmel					
RDI Curren	Cirrus 🕨					
ST-Link	Epson					
Third-Party Driver	Faraday 🕨					
	Freescale 🕨					
r	GD GD GD32F103x					
	Hilscher 🕨					
, OK	Intel F 网体油动*/					
·	Luminary					
。 自口COM1的时好配置、GPIO配置,根据上述参数初始化并使能*/	Marvell 🕨					
	Micronas 🕨					
	NetSilicon 🕨					
	Nuvoton + File					
	NXP >					
	OKI 🕨					
	Samsung 🕨					
	ST +					
	TexasInstruments					
	Toshiha					
	TOBIL DE .					

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