GigaDevice Semiconductor Inc.

GD32F50x Software Development Guide

Application Note
AN270

Revision 1.2

(May. 2026)

Table of Contents

List of Figures

List of Tables

1. Overview

This document is specifically provided for the GD32F50x series MCUs, introducing how to set up projects based on GD32F50x MCUs and perform debugging, as well as how to utilize various modules. The purpose of this application note is to provide illustrative functional introductions to the peripheral resources on GD32F50x series MCUs, enabling users to understand how to perform rapid software development using GD32F50x series MCUs.

Table 1-1. Applicable Products

Type	Model
MCU	GD32F50x Series

2. Software Function Development

2.1. Boot mode selection and configuration

The GD32F50x devices provide four kinds of boot sources. The boot mode is influenced by Security Protection, NBTSB and BTFOSEL bits in OTP3, and Boot pins. The details are shown in the following table ***Table 2-1. Boot Modes***.

The value on the BOOT0 and BOOT1 pins is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT0 and BOOT1 pins after a power-on reset or a system reset to select the required boot source. Once the two pins have been sampled, they are free and can be used for other purposes.

Table 2-1. Boot Modes

Security protection	OTP3		Boot pins		BOOT_MODE[2:0]	Boot source selection
	NBTSB	BTFOSEL	BOOT0	BOOT1		
No protection/Low protection level	0	x	1	1	011	SRAM
No protection/Low protection level	0	x	1	0	001	Bootloader
No protection/Low protection level	0	0	0	x	000	Main Flash memory
No protection/Low protection level	0	1	0	x	101	OTP1
x	1	0	x	x	000	Main Flash memory
x	1	1	x	x	101	OTP1
High protection level	x	0	x	x	000	Main Flash memory
High protection level	x	1	x	x	101	OTP1

After power-up sequence or system reset, the Arm® Cortex®-M33 processor first fetches the stack top value from address 0x0000 0000, then retrieves the base address of the boot code from address 0x0000 0004, and subsequently begins program execution from the base address of the boot code. The selected boot source's memory space will be mapped to the boot memory space, starting from address 0x0000 0000. If the on-chip SRAM (memory space starting at 0x2000 0000) is selected as the boot source, the user must reset the vector table to SRAM in the application initialization code by modifying the NVIC exception vector table and offset address.

When the main FLASH memory is selected as the boot source, the memory space starting from 0x0800 0000 will be mapped to the boot memory space. When OTP1 is selected as the boot source, the memory space starting from address 0x1FF0 0000 will be mapped to the boot memory space.

2.2. SRAM ECC configuration description

2.2.1. SRAM ECC enable and disable

SRAM ECC enable and disable are controlled by bit 4 (ECC_EN bit) in the USER section of the option bytes. When bit 4 is set to 0, SRAM ECC is disabled; when bit 4 is set to 1, SRAM ECC is enabled. By default, SRAM ECC is enabled in the MCU as delivered.

2.2.2. SRAM ECC configuration

GD32F50x supports 7-bit ECC functionality when accessing the first 32KB of SRAM. It can correct 1-bit errors and detect multi-bit (2-bit) errors.

The first 32KB of SRAM with ECC support must be written before being read; otherwise, ECC errors are likely to occur. Unaligned read operations are executed as 32-bit reads. Unaligned write operations involve a read-modify-write process. For example, a 16-bit write first reads 16 bits and then writes the combined 16-bit data. Therefore, when initializing SRAM, only 32-bit writes are allowed.

The EEIC (ECC Error Interrupt Control) module is used to manage SRAM ECC error status and interrupt configuration. When a single-bit correctable error is detected, the status bit (SRAMECCSEIF) in the SYSCFG_SRAMECCSTAT register is set, and the error address is recorded in the SYSCFG_SRAMECCCS register. The status can be cleared by software by writing 1 to it. If a multi-bit (2-bit) uncorrectable error is detected, the status bit (SRAMECCMEIF) in the SYSCFG_SRAMECCSTAT register is set, and the error address is recorded in the SYSCFG_SRAMECCCS register. By enabling the corresponding interrupt enable bits (SRAMECCSEIE or SRAMECCMEIE), these events can trigger NMI and SRAM ECC interrupts when they occur.

2.3. FLASH Configuration Description

2.3.1. Frequency Control

The FMC module uses the CK_FMC clock to access SIP FLASH, and a specific frequency relationship must be maintained with the clock CK_AHB. CK_FMC must not be slower than CK_AHB but must not exceed seven times the frequency of CK_AHB.

$$CK\_FMC \geq CK\_AHB \geq 1/7\ CK\_FMC$$

The recommended frequency switching configuration method is as follows:

If increasing the frequency:

1. Keep the CK_FMC clock source selected as clock CK_AHB, and increase the frequency of CK_FMC and CK_AHB.

2. Then, individually increase the CK_FMC frequency as needed.

If reducing frequency:

1. First, set CK_FMC clock source to system clock CK_AHB.

2. Lower the frequency of CK_FMC and CK_AHB synchronously.

2.3.2. PWDN bit description.

The PWDN bit in the FMC_CTL0 register can control the Flash to enter deep power-down mode when not being operated, reducing power consumption. The PWDN bit is set or cleared by software, does not reset after a system reset, and resets upon a power-on reset. Note the following points:

1. In power-saving mode, the flash memory will only enter deep power-down mode when the PWDN bit is set to 1.

2. When CPU Cbus timeout is enabled (CPUCBUSTO=1) and PWDN=1, accessing the non-zero wait area (data area) of the flash memory after it enters deep power-down mode will cause CBUS timeout, triggering Hardfault error.

3. When CPU Cbus timeout is not enabled (CPUCBUSTO=0) and PWDN=1, accessing the non-zero wait area (data area) of the flash memory after it enters deep power-down mode will wake up the flash memory, requiring flash wake-up time. Accessing the zero wait area (code area) of the flash memory will not wake up the flash memory and does not require waiting.

2.4. RCU clock configuration limitation description

2.4.1. FMC interface clock limitation

The FMC interface clock can be selected as one of CK_AHB, CK_SYS, CK_PLL0, or CK_PLL1 through the FMCSEL bit field in the RCU_ADDCTL register, and it is provided after division by FMCDIV, as shown in ***Figure 2-1. FMC interface clock*** and ***Table 2-2. FMC interface clock configuration***.

Note: Ensure that the FMC interface clock is greater than or equal to the CK_AHB clock but does not exceed 7 times the CK_AHB clock throughout the chip's lifecycle; otherwise, unpredictable issues may occur**.**

An example of FMC interface clock configuration is as follows:

1. When the FMC interface clock source input is selected as CK_AHB, FMCDIV can only be configured as 1 division.

2. When the FMC interface clock source input is selected as CK_SYS, AHBPSC can only be configured as 1, 2, or 4 division. If AHBPSC is configured as 1 division, FMCDIV can only be configured as 1 division; if AHBPSC is configured as 2 division, FMCDIV can only be configured as 1 or 2 division; if AHBPSC is configured as 4 division, FMCDIV can only be configured as 1, 2, 3, or 4 division.
3. When the FMC interface clock source input is selected as CK_PLL0, it must be ensured that CK_AHB≤CK_PLL0≤7*CK_AHB.
4. When the FMC interface clock source input is selected as CK_PLL1, it must be ensured that CK_AHB≤CK_PLL1≤7*CK_AHB.

Additionally, during system frequency scaling up or down, the above restrictions must still be followed. Take GD32F503xx as an example, as follows:

1. When the FMC interface clock source input is selected as CK_PLL1=220MHz, and the system clock switches from 220MHz to internal IRC8M, it must be ensured that the FMC interface clock source input is first switched to CK_AHB before downscaling, then the FMC interface clock is configured to [8MHz, 56MHz] as required.
2. When the FMC interface clock source input is selected as CK_PLL1=8MHz, and the system clock switches from internal IRC8M to 220MHz, it must be ensured that the FMC interface clock source input is first switched to CK_AHB before upscaling, then the FMC interface clock is configured to [220MHz, 252MHz] as required.

Figure 2-1. FMC interface clock

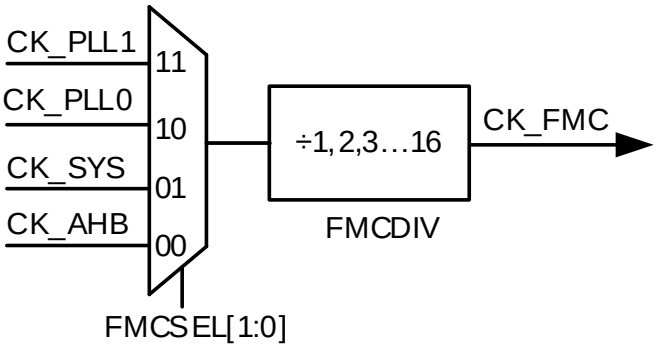

Table 2-2. FMC interface clock configuration

Fields	Descriptions
FMCSEL[1:0]	FMC clock source selection Set and reset by software to control the FMC clock source. 00: CK_AHB selected as FMC source clock (default value) 01: CK_SYS selected as FMC source clock 10: CK_PLL0 selected as FMC source clock 11: CK_PLL1 selected as FMC source clock **Note:** During the lifecycle of the chip, ensure that the FMC interface clock is greater than or equal to the CK_AHB clock, but no more than 7 times the CK_AHB clock , otherwise unpredictable problems may occur.
FMCDIV[3:0]	FMC division factor The division factor is FMCDIV + 1. **Note:** During the lifecycle of the chip, ensure that the FMC interface clock is greater than or equal to the CK_AHB clock, but no more than 7 times the CK_AHB clock , otherwise unpredictable problems may occur.

2.4.2. Increase FMC interface clock

Within the limitation range, increasing the FMC interface clock can improve Data Flash (zero-wait state region) access efficiency. Taking system running clock CK_SYS = CK_AHB = 220MHz with external 8MHz clock source and CK_FMC configured as 240MHz as an example, the code snippet in ***Table 2-3. FMC clock interface 240MHz frequency configuration*** can be added to the end of the SystemInit function in the system_gd32f50x.c file.

Table 2-3. FMC clock interface 240MHz frequency configuration

```
void fmc_clock_240m_by_pll1(void)
{
    /* CK_PLL1P = (CK_IRC8M)/2 * 60 = 240 MHz */
    RCU_CFG1 &= ~(RCU_CFG1_PLL1SEL | RCU_CFG1_PREDIV1);
    RCU_CFG1 |= (RCU_PLL1SRC_HXTAL | RCU_PREDIV1_DIV2);
    RCU_CFG1 &= ~(RCU_CFG1_PLL1MF);
    RCU_CFG1 |= RCU_PLL1_MUL60;

    /* enable PLL1 */
    RCU_CTL |= RCU_CTL_PLL1EN;

    /* wait until PLL1 is stable */
    while(0U == (RCU_CTL & RCU_CTL_PLL1STB)) {
    }
}
```

2.5. TIMER Usage Guide

2.5.1. TIMER Break Polarity Configuration

TIMERx (x=0, 7, 15, 16) has a break function that can be used to handle faults from system sources, on-chip peripherals, and external input signal sources. The polarity of BREAK0 can be configured through BRK0P, or the polarity inversion of the break input signals BRKIN0 and CMP0_OUT can be configured through BRK0INP/BRK0CMP0P.

For advanced timers TIMERx (x=0, 7), channel y (y=0...2) pair has an individual break input CHyBRKIN. The polarity of all channel breaks can be configured by setting the CHBRKP bit in the TIMERx_CCHP0 register, or the polarity inversion of individual channel breaks can be configured by setting CHyBRKINP (y=0...2) in the TIMERx_CHBRKCTL register.

To avoid false triggering, configure TIMERx's BRK0P and CHBRKP as active high whenever possible.

If low-level triggering of break is required, you can choose to configure BRK0INP/BRK0CMP0P and CHyBRKINP to set polarity inversion.

Table 2-4. TIMER break polarity configuration

```
/* BREAK configuration */
timer_break_struct_para_init(&timer_breakpara);
timer_breakpara.runoffstate              = TIMER_ROS_STATE_ENABLE;
timer_breakpara.ideloffstate             = TIMER_IOS_STATE_ENABLE;
timer_breakpara.deadtime                 = 255U;
timer_breakpara.outputautostate          = TIMER_OUTAUTO_ENABLE;
timer_breakpara.protectmode              = TIMER_CCHP0_PROT_OFF;
timer_breakpara.break0state              = TIMER_BREAK0_DISABLE;
timer_breakpara.break0filter             = 0U;
timer_breakpara.break0polarity           = TIMER_BREAK0_POLARITY_HIGH;
timer_break_config(TIMER0, &timer_breakpara);
/* select TIMER0_BRKIN0 with TIMER0_BRKIN0 pin */
timer_break_external_source_config(TIMER0, TIMER_BRKIN0, ENABLE);
/* select TIMER0_BRKIN0 will be inverted */
timer_break_external_polarity_config(TIMER0, TIMER_BRKIN0, TIMER_BRKIN_POLARITY_HIGH);
```

Table 2-5. TIMER channel break polarity configuration

```
/* Channel BREAK configuration */
timer_channel_break_external_status_config(TIMER0, TIMER_CH_0, ENABLE);
timer_channel_break_external_polarity_config(TIMER0,TIMER_CH_0,TIMER_CHx_BREAK_IN_INV);
timer_channel_break_external_status_config(TIMER0, TIMER_CH_1, ENABLE);
timer_channel_break_external_polarity_config(TIMER0,TIMER_CH_1,TIMER_CHx_BREAK_IN_NOT_INV);
timer_channel_break_external_status_config(TIMER0, TIMER_CH_2, ENABLE);
timer_channel_break_external_polarity_config(TIMER0,TIMER_CH_2,TIMER_CHx_BREAK_IN_NOT_INV);
timer_channel_break_config(TIMER0,TIMER_CH_BREAK_ENABLE,TIMER_CH_BREAK_POLARITY_HIGH,
1);
```

2.5.2. TIMER channel break interrupt configuration.

The break mode interrupt for TIMERx (x=0,7) channel y (y=0...2) requires enabling both BRKIE in the TIMERx_DMAINTEN register and CHyBRKIE in the TIMERx_CHBRKCTL register. When the CHyBRKIF bit is set to 1, a channel break interrupt will be triggered.

2.5.3. TIMER break external signal source selection.

TIMERx (x=0,7,15,16) external signal break input sources TIMERx_BRKIN and TIMERx_CHyBRKIN (y=0...2) have default Break input signals.
The break input signal can be selected via the TRIGSEL configuration in the TRIGSEL_TIMERxBRKIN and TRIGSEL_TIMERxCHBRKIN registers. The registers have default values, indicating default break input signals.

2.5.4. Advanced TIMER primary output configuration.

For TIMERx (x=0,7) channel y (y=0...2) outputs CHy_O / CHy_ON, in addition to enabling the corresponding channel CHyEN/CHyNEN, you also need to enable POEN in the TIMERx_CCHP0 register and CHPOENy in the TIMERx_CHBRKCTL register.

Table 2-6. Advanced TIMER primary output configuration

```
/* Channel primary output configuration */
timer_channel_primary_output_config(TIMER0, TIMER_CH_0, ENABLE);
timer_channel_primary_output_config(TIMER0, TIMER_CH_1, ENABLE);
timer_channel_primary_output_config(TIMER0, TIMER_CH_2, ENABLE);
timer_primary_output_config(TIMER0, ENABLE);
```

2.6. CMP usage instructions.

When enabling CMP on GD32F50x series chips, set the CMP0SEN and CMP0EN bits in the CMP0_CS register. When disabling CMP, reset the CMP0SEN and CMP0EN bits in the CMP0_CS register.

2.7. PMU usage instructions.

For 1.2V power domain power detection, before enable VOVD, LVD must be enabled first. After a delay of 50μs, VOVD can then be enabled. Otherwise, VOVD may generate false trigger signals. Subsequently, LVD can either remain enabled or be disabled.

3. Revision History

Table 3-1. Revision History

Revision No.	Description	Date
1.0	Initial Release	Aug. 10, 2025
1.1	1. Modify the description of ***chapter 2.4.1***.	Oct. 31, 2025
1.2	1. Change CK_SYS to CK_AHB in ***chapter 2.3.1***.	May. 11, 2026

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.