

GigaDevice Semiconductor Inc.

Device limitations of GD32G5x3

Errata Sheet

Revision 1.0

(Mar. 2026)

Table of Contents

Table of Contents	2
List of Figures	4
List of Tables	5
1. Introduction	6
1.1. Revision identification	6
1.2. Summary of device limitations	6
2. Descriptions of device limitations	8
2.1. FMC	8
2.1.1. Modifying the FMC_SWP bit when the data buffer is enabled causes an error when the CPU reads the data in main flash.....	8
2.2. USART	8
2.2.1. The high baud rate of USART will cause data loss when using hardware flow control mode	8
2.2.2. In smartcard mode during data reception, EBF flag may fail to be set during retransmission	8
2.2.3. In smartcard mode during data reception, a parity error detected during TX transmission is considered a retransmission, and FERR and RBNE cannot be set	9
2.2.4. When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again	9
2.2.5. IDLEF is set when waking up USART from mute mode via an idle frame	9
2.2.6. When DENR = 1, DDRE = 0, and HCM = 1, RTS remains asserted high.....	10
2.2.7. In Smartcard mode, RTF is set prematurely when the receive timeout threshold (RT) is configured as 0	10
2.2.8. In smartcard mode, the PERR flag is set abnormally	10
2.2.9. In synchronous mode, the PERR flag is set abnormally	10
2.2.10. In deep-sleep mode, the parity error caused by wakeup frames will set PERR bit but not EPERR.....	11
2.2.11. When the TX FIFO function is disabled, the TFE flag is cleared when writing data to TDATA	11
2.2.12. When the TX FIFO function is enabled, the TFNF flag is not cleared as expected	11
2.3. I2C	12
2.3.1. When I2C works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to be stuck.....	12
2.3.2. When I2C is operating as a master transmitter, if the slave responds with NACK to the last byte, a START condition cannot be correctly issued in the transfer complete interrupt	12
2.4. HRTIMER	13
2.4.1. The PER event output is abnormal.....	13
2.4.2. Part of comparator 1 (exclude Slave timer 0 and Slave timer 3) function fails in double source	

trigger mode	13
2.4.3. The STxCHyEN bit fails to accurately reflect the actual channel output state	13
2.4.4. Glitches occur during channel fault auto-recovery	14
2.5. CAN	15
2.5.1. When the CAN operating clock frequency is less than CK_APB2 and an error occurs in DLC segment, which will lead to receiving an incorrect frame ID	15
2.5.2. The mailbox data is not read in time may result in incorrect data reading for the current and next frames	15
2.6. Core	16
2.6.1. Access permission faults are prioritized over unaligned Device memory faults	16
3. Revision history	18

List of Figures

Figure 1-1. Device revision code of GD32G5x3 6

List of Tables

Table 1-1. Applicable products	6
Table 1-2. Device limitations	6
Table 3-1. Revision history	18

1. Introduction

This document applies to GD32G5x3 product series, as shown in [Table 1-1. Applicable products](#). It provides the technical details that need to be paid attention to in the process of using GD32 MCU, as well as solutions to related problems.

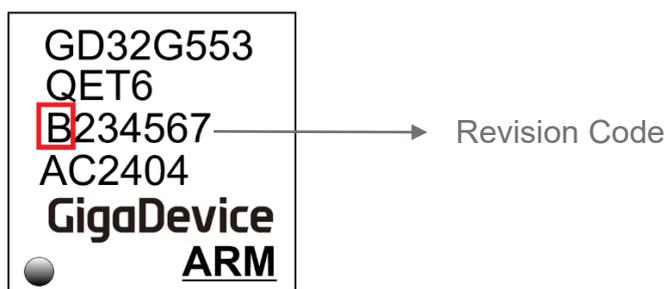
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32G553xx series

1.1. Revision identification

The device revision can be determined by the mark on the top of the package. The 1st code on the line 3 of the mark represents product revision code. As the picture shown in [Figure 1-1. Device revision code of GD32G5x3](#).

Figure 1-1. Device revision code of GD32G5x3



1.2. Summary of device limitations

The device limitations of GD32G5x3 are shown in [Table 1-2. Device limitations](#), please refer to section 2 for more details.

Table 1-2. Device limitations

Module	Limitations	Workaround
		Rev. Code D
FMC	<i>Modifying the FMC_SWP bit when the data buffer is enabled causes an error when the CPU reads the data in main flash</i>	Y
USART	<i>The high baud rate of USART will cause data loss when using hardware flow control mode</i>	Y
	<i>In smartcard mode during data reception, EBF flag may fail to be set during retransmission</i>	N
	<i>In smartcard mode during data reception, a parity error detected during TX transmission is considered a</i>	N

Module	Limitations	Workaround
		Rev. Code D
	<i>retransmission, and FERR and RBNE cannot be set</i>	
	<i>When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again</i>	Y
	<i>IDLEF is set when waking up USART from mute mode via an idle frame</i>	Y
	<i>When DENR = 1, DDRE = 0, and HCM = 1, RTS remains asserted high</i>	Y
	<i>In Smartcard mode, RTF is set prematurely when the receive timeout threshold (RT) is configured as 0</i>	Y
	<i>In smartcard mode, the PERR flag is set abnormally</i>	N
	<i>In synchronous mode, the PERR flag is set abnormally</i>	N
	<i>In deep-sleep mode, the parity error caused by wakeup frames will set PERR bit but not EPERR</i>	Y
	<i>When the TX FIFO function is disabled, the TFE flag is cleared when writing data to TDATA</i>	Y
	<i>When the TX FIFO function is enabled, the TFNF flag is not cleared as expected</i>	Y
I2C	<i>When I2C works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to be stuck</i>	Y
	<i>When I2C is operating as a master transmitter, if the slave responds with NACK to the last byte, a START condition cannot be correctly issued in the transfer complete interrupt</i>	N
HRTIMER	<i>The PER event output is abnormal</i>	N
	<i>Part of comparator 1 (exclude Slave timer 0 and Slave timer 3) function fails in double source trigger mode</i>	Y
	<i>The STxCHyEN bit fails to accurately reflect the actual channel output state</i>	Y
	<i>Glitches occur during channel fault auto-recovery</i>	Y
CAN	<i>When the CAN operating clock frequency is less than CK_APB2 and an error occurs in DLC segment, which will lead to receiving an incorrect frame ID</i>	Y
	<i>The mailbox data is not read in time may result in incorrect data reading for the current and next frames</i>	Y
Core	<i>Access permission faults are prioritized over unaligned Device memory faults</i>	N

Note:

Y = Limitation present, workaround available

N = Limitation present, no workaround available

'-' = Limitation fixed

2. Descriptions of device limitations

2.1. FMC

2.1.1. Modifying the FMC_SWP bit when the data buffer is enabled causes an error when the CPU reads the data in main flash

Description & impact

When the DCEN bit in the FMC_WS register is set, after modifying the FMC_SWP bit in the SYSCFG_CFG0 register, the cache data may be misread, which results in an error in the first main flash data read by the CPU.

Workarounds

Disable DCEN before modifying FMC_SWP and enable DCEN after modifying FMC_SWP.

2.2. USART

2.2.1. The high baud rate of USART will cause data loss when using hardware flow control mode

Description & impact

When using hardware flow control, during high baud rate communication of the USART, data loss may occur due to CTS not being pulled low in time (flow control delay).

Workarounds

Avoid using high baud rates, or use 2 stop bits at high baud rates. When hardware flow control is enabled and 1-bit stop bits are used, the baud rate should be limited to within 3.17 MHz (APBx = 216 MHz).

2.2.2. In smartcard mode during data reception, EBF flag may fail to be set during retransmission

Description & impact

In smartcard mode reception, if the retransmitted data frame satisfies the block count value = BL + 4, and the current retransmitted data frame remains with a parity error and is not moved into the data register, the block end flag (EBF) cannot be set. Additionally, the subsequent BL counter continues counting, causing counter overflow, and EBF remains reset.

Workarounds

Not available.

2.2.3. In smartcard mode during data reception, a parity error detected during TX transmission is considered a retransmission, and FERR and RBNE cannot be set

Description & impact

In smartcard mode during data reception, a parity error detected during TX transmission is considered a retransmission, but the TX pin does not detect a NACK signal. The read data buffer not empty flag (RBNE) and the framing error flag (FERR) cannot be set.

Workarounds

Not available.

2.2.4. When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again

Description & impact

When USART works in multiprocessor communication mode and the USART is woken from mute mode by an idle frame, it will cause the USART will not be woken up when the bus is in idle mode and the USART enters mute mode.

Workarounds

When an idle frame is used to wake the USART mute mode, it is not allowed to enter mute mode while the bus is idle.

2.2.5. IDLEF is set when waking up USART from mute mode via an idle frame

Description & impact

When waking up USART from mute mode via an idle frame, IDLEF is set. If the IDLE interrupt is enabled at this time, the system will enter the IDLE interrupt handler after waking up via the idle frame.

Workarounds

Disable the IDLE interrupt before entering mute mode, and enable the IDLE interrupt when needed.

2.2.6. When DENR = 1, DDRE = 0, and HCM = 1, RTS remains asserted high

Description & impact

When DENR = 1, DDRE = 0, and HCM = 1, the RTS signal remains asserted high, causing hardware flow control to fail.

Workarounds

Ensure that the above three conditions are not all true at the same time during operation.

2.2.7. In Smartcard mode, RTF is set prematurely when the receive timeout threshold (RT) is configured as 0

Description & impact

In Smartcard mode, when the receive timeout threshold (RT) is set to 0, the receive timeout flag (RTF) is set at the START bit of the first data frame.

Workarounds

Configure the RT register with the desired receive timeout threshold value plus 11.

2.2.8. In smartcard mode, the PERR flag is set abnormally

Description & impact

In smartcard mode, when NACK is disabled (NKEN = 0) and SCRTNUM is configured to a non-zero value, the PERR bit fails to be set after the USART receives a frame with a parity error.

Workarounds

Not available.

2.2.9. In synchronous mode, the PERR flag is set abnormally

Description & impact

In synchronous mode, when the data bit inversion function is enabled (DINV = 1), the PERR bit will still be set even if the USART receives a frame with no parity error.

Workarounds

Not available. Do not enable the data bit inversion function in synchronous mode.

2.2.10. In deep-sleep mode, the parity error caused by wakeup frames will set PERR bit but not EPERR

Description & impact

In deep sleep mode, parity errors caused by wake-up frames will set the PERR bit but not EPERR bit. For example, when using USART address-match to wake up from deep sleep mode, if a frame with a parity error and a non-matching address is received first, followed by a frame with no parity error and a matching address, the PERR bit will be set after wakeup while the EPERR bit remains unset..

Workarounds

The software ignores the parity error flag generated in this case.

2.2.11. When the TX FIFO function is disabled, the TFE flag is cleared when writing data to TDATA

Description & impact

When the TX FIFO function is disabled, the TFE (TX FIFO Empty) flag is cleared when writing data to TDATA.

Workarounds

When the TX FIFO function is disabled, ignore all TX FIFO-related status flags.

2.2.12. When the TX FIFO function is enabled, the TFNF flag is not cleared as expected

Description & impact

When the TX FIFO function is enabled, continuously writing data until the FIFO is full results in TFNF = 0. If one data word is then transmitted, TFNF becomes 1. Subsequently, writing one data word into the FIFO fills it again, but TFNF remains 1 instead of being cleared to 0 as expected.

Workarounds

When using the TX FIFO function, use the TFF or TFT flag instead of the TFNF flag as the criterion when writing data.

2.3. I2C

2.3.1. When I2C works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to be stuck

Description & impact

When the I2C is operating as a slave device in 7-bit address mode and the I2C master simulates I2C communication via IO. If the master sends the following sequence, the I2C slave will enter an error state, causing it to malfunction and the SDA line to remain low:

Start + 10-bit Match Head Address + Start + 7-bit Address Read + Wait ACK + Start

When the I2C is operating as a slave device in 10-bit address mode and the I2C master simulates I2C communication via IO. If the master sends the following sequence, the I2C slave will enter an error state, causing it to malfunction and the SDA line to remain low:

Start + 10-bit Mismatch Head Address + Start

or

Start + 10-bit Match Head Address + Wait ACK + 10-bit Mismatch 8-bit Address + Start

Workarounds

Software periodically checks the status of the SDA line. If SDA is detected to be stuck low, reinitialize the I2C module.

2.3.2. When I2C is operating as a master transmitter, if the slave responds with NACK to the last byte, a START condition cannot be correctly issued in the transfer complete interrupt

Description & impact

When I2C is operating as a master and has finished transmitting the last byte of data, if the slave responds with a NACK signal, the master cannot correctly issue a START condition within the transfer complete (TC) interrupt, meaning the next transfer cannot be initiated.

Workarounds

Send a STOP condition in the NACK interrupt handler first, then initiate the next transfer.

2.4. HRTIMER

2.4.1. The PER event output is abnormal

Description & impact

When the slave timer is configured to the center-aligned counting mode and the corresponding OUTROVM bit in HRTIMER_STxCTL1 register is configured to 2'b01, the PER event output results are not as expected.

Workarounds

Not available.

2.4.2. Part of comparator 1 (exclude Slave timer 0 and Slave timer 3) function fails in double source trigger mode

Description & impact

When the TRIGEN bit is set and TRIG0 bit is reset in HRTIMER_STxCTL1 register and the shadow register function is disabled, the valid value periodic automatic accumulation function of comparator 1 fails. When the TRIGEN bit is set and TRIG0 bit is reset in HRTIMER_STxCTL1 register and the shadow register function is enabled, the comparison value of comparator 1 is not updated automatically after the comparison value of comparator 1 is matched. The above two situations are unable to continuously generate the DAC step trigger signals within a PWM cycle, and therefore cannot generate a ramp compensation signal that is synchronized with the PWM.

Workarounds

Add an additional Slave_TIMERx and set the DACTRGS[1:0] bits in corresponding HRTIMER_STxCTL0 to 0b01, 0b10, or 0b11, and then use the timer update event of the added Slave_TIMERx as the trigger source for the DAC step signal.

At the same time, reset the newly added Slave_TIMERx when the originally Slave_TIMER count which resets the DAC signal reaches the period value.

2.4.3. The STxCHyEN bit fails to accurately reflect the actual channel output state

Description & impact

The STxCHyEN bit in the HRTIMER_CHOUTEN register does not correctly indicate the actual output state of the corresponding channel. The specific issues are as follows:

- 1) After a hardware fault auto-recovery, the PWM channel resumes output and the PWM

waveform is restored, but the STxCHyEN bit remains incorrectly set to 0 instead of recovering to 1.

- 2) Under fault conditions, even when the STxCHyEN bit is manually set via software, the PWM output remains disabled, while the STxCHyEN bit incorrectly displays a value of 1.

Due to these issues, when using the fault input function, the STxCHyEN bit cannot reliably indicate the channel state. Therefore, it is recommended to avoid relying on reading this bit for status verification.

Workarounds

Depending on the application scenario, the following alternative solutions can be adopted:

- 1) In certain situations, detect the voltage level of the PWM output pin. Use a rising edge trigger to set the EXTI flag as an alternative means for assessing the channel output status instead of relying on the STxCHyEN bit.
- 2) In certain situations, the channel output state can be determined by detecting fault flags or CMP output status as a substitute for the STxCHyEN bit.

For detailed implementation guidance, refer to the "AN203 GD32G5x3 High-Resolution Timer User Guide".

2.4.4. Glitches occur during channel fault auto-recovery

Description & impact

When using the hardware fault auto-recovery feature and PWM hardware complementary mode (dead-time insertion enabled), glitches lasting tens of nanoseconds may occur prior to the actual restoration of the PWM waveform. The glitch width varies between different Slave Timers, which could result in unintended switch-on of the switching devices and lead to adverse consequences.

Workarounds

Depending on the application scenario, the following alternative solutions can be adopted:

- 1) In certain scenarios, deploy appropriate filtering circuits on the PWM output or driver side to absorb undesired glitches.
- 2) Replace hardware complementary functionality with software-based complementary control.
- 3) In some cases, use software-controlled cycle-by-cycle fault recovery instead of relying on hardware auto-recovery.

For detailed implementation guidance, refer to the "AN203 GD32G5x3 High-Resolution Timer User Guide".

2.5. CAN

2.5.1. When the CAN operating clock frequency is less than CK_APB2 and an error occurs in DLC segment, which will lead to receiving an incorrect frame ID

Description & impact

When the CAN operating clock frequency is less than CK_APB2, if external interference on the CAN bus causes an error frame to be sent due to an error detected in the DLC segment while receiving a frame, this will result in the subsequent frame receiving the error frame ID (even if this error frame ID is filtered by the CAN filter).

Workarounds

Use one of the following solutions:

- 1) The software configures CAN clock source as CK_APB2. Taking CAN1 as an example, the reference code is as follow:

```
{  
    /* configure the CAN1 clock source as CK_APB2 */  
    rcu_can_clock_config(IDX_CAN1, RCU_CANSRC_APB2);  
}
```

- 2) Software checks the frame ID (applicable when the CAN operating clock frequency is less than CK_APB2). When the CAN mailbox receives a non-target frame ID and an error occurs, the software needs to reconfigure the CAN reception mailbox parameters. The specific configuration steps are as follows:
 - a) Enter inactive mode
 - b) Reconfigure the reception mailbox parameters
 - c) Enter normal mode

2.5.2. The mailbox data is not read in time may result in incorrect data reading for the current and next frames

Description & impact

When handling mailbox receive, if a new CAN frame (the next frame) is moved into the receive mailbox while reading this receive mailbox data (the current frame), it may lead to incorrect reading of the current frame and the next frame data.

Note: Only the current frame and the next frame are affected.

Workarounds

Use one of the following solutions:

- 1) Use mailbox reception interrupt and configure it to highest priority. When a receive mailbox interrupt occurs, promptly read and process the mailbox data (before the end of the next frame).
- 2) Use CAN FIFO reception instead of mailbox reception.
- 3) Enable the mailbox queue by setting the RPFQEN bit and set the mailbox reception interrupt to the highest priority.

2.6. Core

About Cortex-M33 limitations, please refer to “Cortex-M33 AT623 and Cortex-M33 with FPU AT624 Software Developer Errata Notice”. This document can be downloaded on ARM official website.

2.6.1. Access permission faults are prioritized over unaligned Device memory faults

This limitation refers to Arm ID number 1080541 in “Cortex-M33 AT623 and Cortex-M33 with FPU AT624 Software Developer Errata Notice”.

Description & impact

A load or store which causes an unaligned access to Device memory will result in an UNALIGNED UsageFault exception. However, if the region is not accessible because of the MPU access permissions (as specified in MPU_RBAR.AP), then the resulting MemManage fault will be prioritized over the UsageFault.

This erratum affects all configurations of the Cortex-M33 processor with the MPU enabled.

The failure occurring conditions are as follows:

The MPU is enabled and:

- A load/store access occurs to an address which is not aligned to the data type specified in the instruction.
- The memory access hits one region only.
- The region attributes (specified in the MAIR register) mark the location as Device memory.
- The region access permissions prevent the access (that is, unprivileged or write not allowed).

The implications of this limitation is that the MemManage fault caused by the access permission violation will be prioritized over the UNALIGNED UsageFault exception because of the memory attributes.

Workarounds

Not available. However, it is expected that no existing software is relying on this behavior since it was permitted in Armv7-M.(The CM33 is Armv8-M).

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Mar.2 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.