GigaDevice Semiconductor Inc.

Device limitations of GD32C231

Errata Sheet

Revision 1.4

(May. 2026)

Table of Contents

List of Figures

List of Tables

1. Introduction

This document applies to GD32C231 product series, as shown in ***Table 1-1. Applicable products***. It offers technical guidance for using GD32MCU and provides workaround to current device limitations.

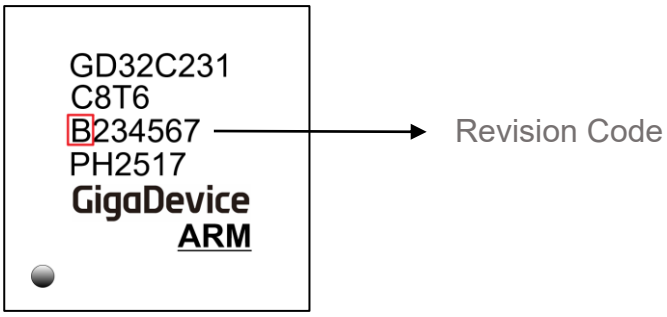
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32C231xx series

1.1. Revision identification

The device revision can be identified according to the mark on the top of the package. The 1st code on Line 3 of the mark is the product revision code, as shown in ***Figure 1-1. Device revision code of GD32C231***.

Figure 1-1. Device revision code of GD32C231



1.2. Summary of device limitations

The device limitations of GD32C231 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 A	Rev. Code B	Rev. Code D
FMC	***When the first Flash page is configured as an execute-only region, booting the program from SRAM causes RPERR to be set***	Y	Y	Y
PMU	***When the chip is powered on, there is a probability that the PORRSTF flag cannot be set***	Y	Y	--
	Frequent wake-up signals before and after MCU enters the standby mode results in wake-up failure in the standby mode	N	N	--

Module	Limitations	Workaround		
		Rev. Code A	Rev. Code B	Rev. Code D
	When MCU enters Deep-sleep / Deep-sleep1 mode, if a system reset occurs, a very small number of chips may experience program runaway	Y	Y	--
	When the MCU operating voltage VDD is between VPOR and VDD_Min, there is a low probability of EFLASH read errors	Y	Y	--
RCU	*The clock monitor of the HXTAL is abnormal*	Y	Y	--
	When OBRLD is set, both EPRSTF and OBLRSTF flags are set	Y	Y	--
GPIO	*PA12 port LOCK function is abnormal*	Y	Y	--
ADC	*ADC data acquisition error occurs when the ADC clock is equal to or less than 1/4 of its APB bus clock*	Y	Y	Y
	When the ADC external trigger function is enabled and the trigger source switches from low level to high level, an ADC conversion will be triggered	Y	Y	Y
	ADC_WDx_OUT signal cannot be cleared after the analog watchdog function is disabled	Y	Y	Y
Tsensor	*Enabling or disabling Tsensor can cause VCORE fluctuations*	N	N	--
USART	*When USART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again*	Y	Y	--
	When waking up the USART from mute mode via an idle frame, IDLEF is set	Y	Y	--
	The high baud rate of USART will cause data loss when using hardware flow control mode	Y	Y	Y
	When automatic baud rate detection mode 1 and the data bit inversion function are enabled simultaneously, the reception of the data for auto baud rate detection fails	Y	Y	Y
	In smartcard mode during data reception, EBF flag may fail to be set during retransmission	N	N	N
	In smartcard mode during data reception, a parity error detected during TX transmission is considered a retransmission, and FERR and RBNE cannot be set	N	N	N
	When DENR = 1, DDRE = 0, and HCM = 1, RTS remains asserted high	N	N	N
	In deep-sleep mode, the parity error caused by wakeup frames will set PERR and EPERR bit	N	N	N

Module	Limitations	Workaround		
		Rev. Code A	Rev. Code B	Rev. Code D
	In smartcard mode, the PERR flag is set abnormally	N	N	N
	In synchronous mode, the PERR flag is set abnormally	N	N	N
	In deep-sleep mode, the parity error caused by wakeup frames will set PERR bit but not EPERR	N	N	N
	When an auto baud rate detection error occurs, the USART_BAUD register is updated with an incorrect baud rate value	Y	Y	Y
SPI	*When the SPI slave works in non-TI mode and the data valid sampling edge is the first clock transition edge and CRC function is enabled, if the slave is not selected by the chip and there is still a clock on the SCK line, the slave CRC will continue working*	Y	Y	Y
	When SPI works as master or slave and CRC function is enabled, the CRC register is not automatically cleared after CRC transmission or verification is completed	Y	Y	Y
I2S	*The I2S MCK clock output function is abnormal*	Y	Y	--
I2C	*When I2C works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to be stuck*	Y	Y	--
	When the I2C slave is configured in 10-bit address mode, if the external master does not send a STOP signal after transmitting a frame of data, the I2C slave will be unable to match the slave address in subsequent operations	Y	Y	--
	SMBUS master timeout caused by the slave pulling down the SCL line may result in the SMBUS master failing to issue a STOP signal	N	N	--
	When the I2C is configured as a master in 10-bit address mode and fails to send a STOP signal after transmitting a data frame, subsequent data frame transmissions will encounter anomalies	Y	Y	Y
	Transmission timing abnormality when the I2C master is configured in 10-bit address reception mode with HEAD10R=1	Y	Y	Y
	When the I2C clock is configured to an IRC48M division factor other than 1, an address mismatch in low-power wake-up scenarios will cause subsequent wake-up failure	Y	Y	--

Module	Limitations	Workaround		
		Rev. Code A	Rev. Code B	Rev. Code D
	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	Y	Y	Y

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. When the first Flash page is configured as an execute-only region, booting the program from SRAM causes RPERR to be set

Description & impact

If the first page of Flash is configured as an execute-only region, booting the program from SRAM will cause RPERR to be set unexpectedly.

Workarounds

When this issue occurs, ignore the first RPERR event. The flag can be cleared by writing 1 to the bit.

2.2. PMU

2.2.1. When the chip is powered on, there is a probability that the PORRSTF flag cannot be set

Description & impact

When the chip is powered on, there is a probability that the PORRSTF flag cannot be set, making it impossible to use the PORRSTF flag to detect whether a power-on reset has occurred.

Workarounds

During the application initialization phase, the code should first check whether a specified backup domain register (such as RTC_BKP0) contains a marked value (such as 0xA5). If it does not contain the marked value, it is considered that a power-on reset has occurred, and the software should then write the marked value to the backup domain register. Otherwise, it is considered that a power-on reset has not occurred. This method requires occupying one backup domain register.

2.2.2. Frequent wake-up signals before and after MCU enters the standby mode results in wake-up failure in the standby mode

Description & impact

When there are frequent wake-up signals on the wake-up pin (WKUPx), if the MCU exits quickly (within 20us) after entering the standby mode, the internal signal CORE_POR_H cannot be set to 1 after reset, which results in the CPU cannot run, and finally the MCU cannot be woken up.

Note: When the above problem occurs, the external NRST reset also fails to make the CPU run again, and the chip needs to be repowered.

Workarounds

Not available. For the above application scenarios, it is not recommended to use standby mode, and it is recommended to use Deep-sleep / Deep-sleep1 mode instead.

2.2.3. When MCU enters Deep-sleep / Deep-sleep1 mode, if a system reset occurs, a very small number of chips may experience program runaway

Description & impact

When the MCU enters Deep-sleep / Deep-sleep1 mode (at this time, the EFLASH enters low-power mode or power-down mode), if a reset occurs, the EFLASH power-on timing parameter settings approach the limit (this restriction does not apply to other modes). This may cause abnormal loading of the first few bytes of EFLASH data in a very small number of chips, resulting in MCU program runaway.

Workarounds

Use one of the following solutions:

1) Avoid resetting the MCU in deep sleep mode/deep sleep mode 1 in the application (such as, NRST or FWDGT reset). Normal wake-up does not have this issue.
2) Enable the hardware watchdog through option bytes. If the issue occurs, the hardware watchdog can reset and recover the system.

Note: If you have any questions regarding this erratum, it is recommended to contact the original manufacturer's technical support for more detailed information.

2.2.4. When the MCU operating voltage V_{DD} is between V_{POR} and V_{DD_Min}, there is a low probability of EFLASH read errors

Description & impact

When the MCU operating voltage V_{DD} is within the range of V_{POR} (typical value 1.633V) and V_{DD_Min} (typical value 2.3V), there is a low probability of EFLASH read errors, causing the MCU to enter an abnormal state.

Workarounds

For the above application scenario, the BOR rising threshold can be set to 2.1V, the BOR falling threshold to 2.0V, and the BOR reset function can be enabled by modifying the option bytes.

2.3. RCU

2.3.1. The clock monitor of the HXTAL is abnormal

Description & impact

After the HXTAL clock monitor is enabled, when the HXTAL clock is lost, the system clock will automatically switch to CK_IRC48MDIV_SYS (clock source provided by IRC48M). However, there is a probability that the NMI interrupt will not be generated. That is, the HXTAL clock monitor cannot reliably detect the loss of the HXTAL clock.

Workarounds

After enabling the HXTAL clock monitor function, the software polls the current clock source of the system to check whether a clock switch has occurred.

2.3.2. When OBRLD is set, both EPRSTF and OBLRSTF flags are set

Description & impact

A system reset caused by OBRLD being set will result in both EPRSTF and OBLRSTF flags being set simultaneously.

Workarounds

In the system reset source processing code snippet, when it is detected that both EPRSTF and OBLRSTF are set simultaneously, the software determines it as a system reset caused by option byte loading, and clears all reset flags after processing all reset source decisions.

2.4. GPIO

2.4.1. PA12 port LOCK function is abnormal

Description & impact

After enabling the LOCK function of PA12 port immediately after reset, the CTL12 bit field in the GPIOA_CTL register can still be configured, but the configuration result does not match expectations.

Workarounds

To use the LOCK function on PA12 port, software needs to first read the entire GPIOA_CTL

register, write the read value back, and then use the LOCK function.

2.5. ADC

2.5.1. ADC data acquisition error occurs when the ADC clock is equal to or less than 1/4 of its APB bus clock

Description & impact

When the ADC clock is equal to or less than 1/4 of its APB bus clock, the ADC_RDATA register is read immediately after the EOC is set and a data acquisition error occurs.

Workarounds

When the delay between reading EOC flag and reading ADC_RDATA is no more than two ADC clocks, after the EOC flag is set, software needs to delay two ADC clocks before reading the ADC_RDATA register.

2.5.2. When the ADC external trigger function is enabled and the trigger source switches from low level to high level, an ADC conversion will be triggered

Description & impact

When the ADC external trigger function is enabled and the trigger source switches from low level to high level, an ADC sampling will be triggered. For example, when the external trigger source function of the routine sequence is enabled (ETERC = 1), and the external trigger source switches from software trigger (0b111) to TIMER0_CH1 (0b001), an ADC conversion will be triggered. This issue imposes limitations on applications that disable the external trigger function by switching the external trigger source.

Workarounds

When disabling the ADC external trigger, directly disable the ADC external trigger function instead of switching the external trigger source.

2.5.3. ADC_WDx_OUT signal cannot be cleared after the analog watchdog function is disabled

Description & impact

After the analog watchdog function is disabled, the ADC_WDx_OUT signal cannot be cleared. In this case, if ADC_WDx_OUT remains high and is used as an external trigger input for a

timer, the timer will be triggered continuously.

Workarounds

The ADC must be reinitialized by software.

2.6. Tsensor

2.6.1. Enabling or disabling Tsensor can cause V_{CORE} fluctuations

Description & impact

When the Tsensor operating state switches from disabled to enabled or from enabled to disabled, it may cause V_{CORE} fluctuations, potentially leading to a HardFault.

Workarounds

Not available. Not recommended for customer use.

2.7. USART

2.7.1. 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 to 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.

2.7.2. When waking up the USART from mute mode via an idle frame, IDLEF is set

Description & impact

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

Workarounds

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

2.7.3. 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. In flow control mode with 1 stop bit, limit the baud rate to below 0.7 MHz.

2.7.4. When automatic baud rate detection mode 1 and the data bit inversion function are enabled simultaneously, the reception of the data for auto baud rate detection fails

Description & impact

When automatic baud rate detection mode 1 (ABDM = 01) and the data bit inversion function (DINV = 1) are enabled, after sending an automatic baud rate detection request (ABDCMD = 1), the data received in the next frame differs from the data on the RX line as retrieved from USART_RDATA, resulting in a reception error.

Note: The issue only affects the data of the auto baud rate detection frame.

Workarounds

If the user is concerned about the data of the automatic baud rate detection frame, one of the following solutions can be used:

1) If both the automatic baud rate detection function and data bit inversion function are required, automatic baud rate detection mode 0 (ABDM = 00) can be used instead.
2) Manually invert the most significant bit of the data in software. If parity check is enabled, the parity bit must be recalculated in software.

2.7.5. 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.

2.7.6. 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.

2.7.7. 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

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

2.7.8. In deep-sleep mode, the parity error caused by wakeup frames will set PERR and EPERR bit

Description & impact

In deep sleep mode, parity errors caused by wake-up frames will set the PERR 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 and EPERR bit will be set after wakeup.

Workarounds

Not available. The software ignores the parity error flag generated in this case.

2.7.9. 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.

2.7.10. 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.

2.7.11. 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 parity error and a matching address, the PERR bit will be set after wakeup while the EPERR bit remains reset.

Workarounds

Not available. The software ignores the parity error flag generated in this case.

2.7.12. When an auto baud rate detection error occurs, the USART_BAUD register is updated with an incorrect baud rate value

Description & impact

When an auto baud rate detection error occurs, the USART_BAUD register is updated with an incorrect baud rate value.

Workarounds

When this issue occurs (ABDE = 1), reconfigure the baud rate to the default value.

2.8. SPI

2.8.1. When the SPI slave works in non-TI mode and the data valid sampling edge is the first clock transition edge and CRC function is enabled, if the slave is not selected by the chip and there is still a clock on the SCK line, the slave CRC will continue working

Description & impact

When SPI works in the slave non-TI mode (TMOD = 0) and the data effective sampling edge is the first clock transition edge (CKPH = 0), and the CRC function is enabled, if the slave is not selected by the chip at this time, but there is still a clock on the SCK line, the slave CRC will continue to work, and then CRCERR will be set. This issue imposes limitations on multi-slave (one-master, multiple-slave) applications.

Workarounds

Use one of the following solutions:

1) Use software chip selection. When the slave detects that it is not selected, it actively disables the CRC functionality.
2) The master and slave agree that the effective data sampling edge is the second clock transition edge (CKPH =1).

2.8.2. When SPI works as master or slave and CRC function is enabled, the CRC register is not automatically cleared after CRC transmission or verification is completed

Description & impact

When SPI works as master or slave and CRC function is enabled, after CRC transmission or verification is completed (by setting CRCNT to send or receive CRC data and verify), the CRC registers (SPI_RCRC and SPI_TCRC) will not be automatically cleared. This will cause the CRC calculation to continue using the CRC value calculated in the previous frame during the next frame of data communication, resulting in CRCERR being set.

Workarounds

Before each frame of data communication, software clears the CRC value (by clearing CRCEN and then setting it).

2.9. I2S

2.9.1. The I2S MCK clock output function is abnormal

Description & impact

When the I2S master clock output function (MCKOEN = 1) is enabled, the MCK clock cannot be output properly, which makes it impossible to provide clock input to devices that require an additional master clock.

Workarounds

Emulating MCK clock outputs by using timers to generate PWM signals. For specific implementation details, refer to the <11_I2S_Audio_Player> example in “GD32C2x1_Demo_Suites\GD32C231C_EVAL_Demo_Suites”.

2.10. I2C

2.10.1. 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.

2.10.2. When the I2C slave is configured in 10-bit address mode, if the external master does not send a STOP signal after transmitting a frame of data, the I2C slave will be unable to match the slave address in subsequent operations

Description & impact

When the I2C slave is configured in 10-bit address mode, if the external master does not send a STOP signal after transmitting a frame of data and instead sends a START signal to initiate the transmission of a second frame, the I2C slave will misinterpret the second byte of the slave address (the lower 8 bits of the 10-bit address) as data, and the address match flag (ADDSEND) will not be set. For example, if the slave is in address polling mode, it will continuously wait for an address match and remain stuck in a loop. Similarly, if the slave is in interrupt or DMA mode, it will fail to process subsequent data due to the inability to match the slave address.

Workarounds

When the I2C slave is operating in 10-bit address mode, the external I2C master must send the corresponding STOP signal at the end of each frame transmission.

2.10.3. SMBUS master timeout caused by the slave pulling down the SCL line may result in the SMBUS master failing to issue a STOP signal

Description & impact

When I2C acts as an SMBUS master, the timeout caused by the slave pulling down the SCL line may result in the SMBUS master failing to issue a STOP signal, which does not comply with SMBUS protocol requirements.

Workarounds

Not available.

2.10.4. When the I2C is configured as a master in 10-bit address mode and fails to send a STOP signal after transmitting a data frame, subsequent data frame transmissions will encounter anomalies

Description & impact

When the I2C master fails to send a STOP signal after transmitting a data frame and the software subsequently configures it to master receive mode, regardless of whether HEAD10R is set to 0 or 1, the waveform of the master receive part will always be RESTART + 10-bit

address head + Master Receive. This means the HEAD10R configuration becomes ineffective.

If the I2C master fails to send a STOP signal after transmitting a data frame, when HEAD10R = 0, the subsequent RESTART will directly enter the waveform of master receive mode. when HEAD10R = 1, the subsequent RESTART will repeatedly send the first part of the address sequence in a loop (RESTART+10bit address head).

Workarounds

When the I2C master is configured in 10-bit address mode, a corresponding STOP signal must be sent at the end of each frame transmission.

2.10.5. Transmission timing abnormality when the I2C master is configured in 10-bit address reception mode with HEAD10R=1

Description & impact

When the I2C master is configured in 10-bit address reception mode with HEAD10R=1, the I2C master timing sequence is START + 10-bit address head + Master Receive, which causes the slave device to not ACK and fails to address the slave device. Under this configuration, a normal master transmission sequence should be START + 10-bit address head (write) + second address byte + RESTART + 10-bit address head (read).

Workarounds

When the master needs to send the sequence START + 10-bit address head (write) + second address byte + RESTART + 10-bit address head (read), configure HEAD10R to 0.

2.10.6. When the I2C clock is configured to an IRC48M division factor other than 1, an address mismatch in low-power wake-up scenarios will cause subsequent wake-up failure

Description & impact

When the I2C clock is configured to an IRC48M division factor other than 1, an address mismatch in low-power wake-up scenarios may cause the I2C clock to be lost, resulting in subsequent wake-up failure.

Workarounds

For low-power wake-up scenarios, configure IRC48MDIV_PER to divide by 1.

2.10.7. 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.

2.11. Core

About Cortex-M23 (r1p0) limitations, please refer to “Cortex-M23 Software Developer Errata Notice”. This document can be downloaded on ***ARM official website***.

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jun.7 2025
1.1	Add limitations of PMU, refer to ***Frequent wake-up signal before and after MCU enters the standby mode results in wake-up failure in the standby mode***	Sep.22 2025
1.2	1. Add limitations of PMU, refer to ***When MCU enters Deep-sleep/Deep-sleep1 mode, if a system reset occurs, a very small number of chips may experience program runaway*** 2. Update the workarounds description of PMU limitation, refer to ***Frequent wake-up signal before and after MCU enters the standby mode results in wake-up failure in the standby mode*** 3. Add limitations of RCU, refer to ***When OBRLD is set, both EPRSTF and OBLRSTF flags are set*** 4. Update the description of GPIO limitation, refer to ***PA12 port LOCK function is abnormal*** 5. Add limitations of ADC, refer to ***When the ADC operates in routine sequence mode, setting the ADCON bit again after enabling the ADC will start an ADC conversion*** 6. Add limitations of ADC, refer to ***When the ADC external trigger function is enabled and the trigger source switches from low level to high level, an ADC conversion will be triggered*** 7. Add limitations of SPI, refer to ***When the SPI slave works in non-TI mode and the data valid sampling edge is the first clock transition edge and CRC function is enabled, if the slave is not selected by the chip and there is still a clock on the SCK line, which will cause the slave CRC to continue working*** 8. Add limitations of I2C, refer to ***When the I2C***	Nov.3 2025

	slave is configured in 10-bit address mode, if the external master does not send a STOP signal after transmitting a frame of data, the I2C slave will be unable to match the slave address in subsequent operations 9. Add limitations of I2C, refer to ***SMBUS master timeout caused by the slave pulling down the SCL line may result in the SMBUS master failing to issue a STOP signal***	
1.3	1. Add limitations of Rev. Code A 2. Update the description of RCU limitation, refer to ***The clock monitor of the HXTAL is abnormal*** 3. Delete ADC limitation, refer to ***When the ADC operates in routine sequence mode, setting the ADCON bit again after enabling the ADC will start an ADC conversion*** 4. Add limitations of PMU, refer to ***When the MCU operating voltage VDD is between VPOR and VDD_Min, there is a low probability of EFLASH read errors*** 5. Add limitations of Tsensor, refer to ***Enabling or disabling Tsensor can cause VCORE fluctuations*** 6. Add limitations of ***USART*** 7. Add limitations of SPI, refer to ***When SPI works as master or slave and CRC function is enabled, the CRC register is not automatically cleared after CRC transmission or verification is completed*** 8. Add limitations of I2C, refer to ***When the I2C is configured as a master in 10-bit address mode and fails to send a STOP signal after transmitting a data frame, subsequent data frame transmissions may encounter anomalies*** and ***Transmission timing abnormality when the I2C master is configured in 10-bit address reception mode with HEAD10R=1***	Jan.4 2026
1.4	1. Add limitations of Rev. Code D 2. Add limitations of FMC, refer to ***When the first Flash page is configured as an***	May.24 2026

	execute-only region, booting the program from SRAM causes RPERR to be set 3. Add limitations of ADC, refer to ***ADC_WDx_OUT signal cannot be cleared after the analog watchdog function is disabled*** 4. Add limitations of USART, refer to ***When DENR = 1, DDRE = 0, and HCM = 1, RTS remains asserted high*** and ***In deep-sleep mode, the parity error caused by wakeup frames will set PERR and EPERR bit*** and ***In smartcard mode, the PERR flag is set abnormally*** and ***In synchronous mode, the PERR flag is set abnormally*** and ***In deep-sleep mode, the parity error caused by wakeup frames will set PERR bit but not EPERR*** and ***When an auto baud rate detection error occurs, the USART_BAUD register is updated with an incorrect baud rate value*** 5. Add limitations of I2C, refer to ***When the I2C clock is configured to an IRC48M division factor other than 1, an address mismatch in low-power wake-up scenarios will cause subsequent wake-up failure*** and ***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***	

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.