GigaDevice Semiconductor Inc.

Device Limitations of GD32A503

Errata Sheet

Table of Contents

List of Figures

List of Tables

1. Introduction

This document applies to GD32A503 product series, as shown in ***Table 1-1. Applicable products***. It offers technical guidance for using GD32MCU and provides workaround to current device limitations.

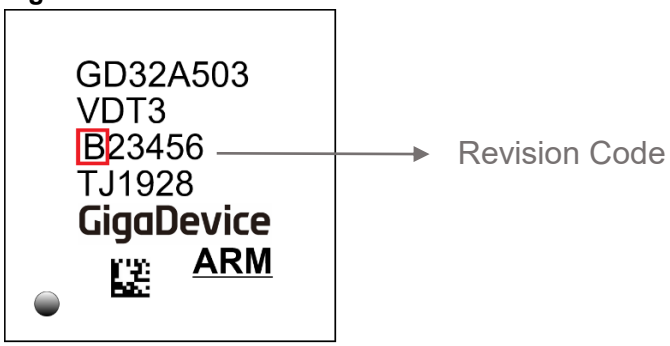
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32A503xx 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 GD32A503***.

Figure 1-1. Device revision code of GD32A503



1.2. Summary of device limitations

The device limitations of GD32A503 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 B	Rev. Code E
System	***The on-chip SRAM boot mode is not supported***	--	N
	When the system clock switches between high and low frequencies frequently, the system may enter an abnormal state	Y	Y
FMC	***When the START command is sent after MER and FSTPG are set, PGSERR error flag is not set***	Y	Y
PMU	***Frequent wake-up signal before and after MCU enters the standby mode results in wake-up failure in the***	N	--

Module	Limitations	Workaround	
		Rev. Code B	Rev. Code E
	standby mode		
	MCU resets twice during the power-on process	N	--
	When the MCU enters deep-sleep mode, the VDD of some 64-Pin package chips will increase the current consumption of about 200uA	Y	Y
BKP	*Reading of BKP_DATA register after BKPRST is set results in MCU crash*	Y	Y
RCU	*MCU in deep sleep mode cannot be woken up after DSLP_HOLD bit in DBG register is set*	Y	Y
GPIO	*The square wave or negative voltage on PD4 will affect the stability of core voltage*	N	--
MFCOM	*When TMOUT is set to 0b'10 or 0b'11 under the condition that MFCOM is used as a UART receiver, the reset function of the timer encounters error*	Y	Y
	When the baud rate is low under the condition that MFCOM is used as a UART receiver, the receiving of the data after the second frame encounters error	Y	Y
	When MFCOM is configured in USART mode, if there is a deviation in the baud rate, synchronizing the baud rate can lead to errors in data transmission and reception	N	N
DBG	*MCU cannot enter the debug mode from the standby mode after STB_HOLD bit in DBG register is set*	Y	Y
	MCU in the standby mode cannot be woken up after STB_HOLD bit in DBG register is set	Y	Y
ADC	*Over 5 V input voltage of PB13 pin results in incorrect voltage sampling of PD14 pin*	Y	--
USART	*Negative narrow pulse interference results in wrong data received by the serial port*	N	N
TIMER	*CHxCAPFLT and CHxCAPPSC can be set only if the bit is set twice after converting from the output mode to the input mode in PROT mode 2*	Y	Y
	A constantly high or low level in a cycle of PWM is output when the duty ratio is over 50% in the composite PWM mode	Y	Y
CAN	*Reset of USART0/USART1 peripheral results in communication error of CAN0/CAN1*	Y	--
	CAN mailbox 0 converted from receiving mailbox to transmitting mailbox fails to send the data frame	Y	--
	As a transmitting node, CAN executes unexpected self-calibration function	Y	--

Module	Limitations	Workaround	
		Rev. Code B	Rev. Code E
	CAN manual bus off recovery function faults	Y	Y
	CAN RAM area may be tampered in receiving mailbox processing	Y	Y
	After Bus off recovery, transmit error count is not cleared automatically	Y	Y
	The mailbox data is not read in time may result in incorrect data reading for the current and next frames	Y	Y
	When the CAN operating clock frequency is less than CK_PCLK2 and an error occurs in DLC segment, which will lead to receiving an incorrect frame ID	Y	Y

Note:

Y = Limitation present, workaround available

N = Limitation present, no workaround available

'--' = Limitation fixed

2. Descriptions of device limitations

2.1. System

2.1.1. The on-chip SRAM boot mode is not supported

Description & impact

The on-chip SRAM boot mode is not supported on the MCU with Rev. Code E, but supported on the MCU with Rev. Code B. User can configure Boot0 and Boot1 pins to be high level for booting from on-chip SRAM on the MCU with Rev. Code B.

Workarounds

Please refer to “GD32A50x_User_Manual_Rev1.5” or later for the detail boot mode configuration.

2.1.2. When the system clock switches between high and low frequencies frequently, the system may enter an abnormal state

Description & impact

When the system clock frequency is switched directly from high to low or from low to high, the system may enter an abnormal state, such as a HardFault or system crash, due to the core voltage fluctuations. The preceding application scenarios include jumping between boot code and app code or entering and exiting low-power applications, etc.

Workarounds

For application scenarios that require jumping between boot code and app code, users can update and use the "GD32A50x_Firmware_Library_V1.4.0" or later, which includes related frequency switching patches.

For application scenarios that require entering and exiting low power modes, users can refer to the frequency switching code patches in the latest firmware library to avoid issues. Before entering low power mode, add the down-frequency patch to reduce the system clock to the internal IRC8M. After exiting low power mode, add the up-frequency patch to increase the system clock to the specified operating frequency.

For other frequency switching demands, please contact GD FAE for support.

2.2. FMC

2.2.1. When the START command is sent after MER and FSTPG are set, PGSERR error flag is not set

Description & impact

PGSERR error does not occur when START is set after MER and FSTPG in FMC_CTL0 register are set during operation in bank0 area.

Workarounds

Make sure that MER and FSTPG are not set at the same time in the software during operation in bank0 area.

2.3. PMU

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

Description & impact

When the internal signal STBY_CTL is reset to allow MCU to enter the standby mode, if Tglitch is less than 100 ns, MCU cannot be woken up because of incorrect output of Vcore due to narrow glitch.

Note: Tglitch is the time from low level of STBY_CTL to wake-up signal (PA0 is a high level).

Workarounds

Avoid frequent periodic wake up signals on PA0.

2.3.2. MCU resets twice during the power-on process

Description & impact

POR and BOR act on NRST successively, which will cause NRST to have a 60us high level to low level then high level process and cause MCU to reset twice.

Workarounds

Customers need to evaluate based on specific application scenarios.

2.3.3. When the MCU enters deep-sleep mode, the VDD of some 64-Pin package chips will increase the current consumption of about 200uA

Description & impact

When the MCU enters deep-sleep mode and the flash memory enters the power-down mode, some 64-Pin package chips will increase the current consumption of about 200uA on the VDD.

Workarounds

Before entering the deep-sleep mode, software sets the SLEEP_SLP bit in the FMC_WS register. That is, when the MCU enters the deep sleep-mode, the flash memory enters the sleep-mode, and the VDD will have an additional current consumption of about 11uA. The reference code can refer to the firmware library driver interface pmu_to_deepsleepmode, as follows:

```
void pmu_to_deepsleepmode(uint32_t ldo, uint32_t lowdrive, uint8_t deepsleepmodecmd)
{
    /* flash goto sleep mode when MCU enters deepsleep mode */
    REG32(0x40022000) |= (uint32_t)(1<<14);
    /* clear stbmod and ldolp bits */
    PMU_CTL &= ~((uint32_t)(PMU_CTL_STBMOD | PMU_CTL_LDOLP | PMU_CTL_LDEN));
    /* set ldolp bit according to pmu_ldo */
    PMU_CTL |= (ldo | lowdrive);
    /* set sleepdeep bit of Cortex-M33 system control register */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;
    /* select WFI or WFE command to enter deepsleep mode */
    if(WFI_CMD == deepsleepmodecmd) {
        __WFI();
    } else {
        __SEV();
        __WFE();
        __WFE();
    }
    /* reset sleepdeep bit of Cortex-M33 system control register */
    SCB->SCR &= ~((uint32_t)SCB_SCR_SLEEPDEEP_Msk);
}
```

2.4. BKP

2.4.1. Reading of BKP_DATA register after BKPRST is set results in MCU crash

Description & impact

Reading of BKP_DATA register after BKPRST bit in RCU_BDCTL register is set results in

MCU crash.

Workarounds

Make sure that BKPRST is reset when BKP_DATA register is being read.

2.5. RCU

2.5.1. MCU in deep sleep mode cannot be woken up after DSLP_HOLD bit in DBG register is set

Description & impact

MCU in deep sleep mode cannot be woken up by EXTI after DSLP_HOLD bit in DBG register is set.

Workarounds

Switch the system clock to internal IRC8M before MCU enters the deep sleep mode.

2.6. GPIO

2.6.1. The square wave or negative voltage on PD4 will affect the stability of core voltage

Description & impact

The square wave or negative voltage on PD4 will affect the stability of core voltage (1.1V domain).

Workarounds

Avoid input square wave signal or negative voltage signal on PD4 pin.

2.7. MFCOM

2.7.1. When TMOUT is set to 0b’10 or 0b’11 under the condition that MFCOM is used as a UART receiver, the reset function of the timer encounters error

Description & impact

When TMOUT is set to 0b’10 or 0b’11, the reset function of the timer encounters error. When

TMOUT output is opposite to the configured TMOUT, the reset signal set for TMRST will flip the TMOUT output. The lower 8 bits of the counter will be reloaded, but the operation of decrementing the higher 8 bits by one is not performed, causing an additional clock output by TMOUT.

Workarounds

1) When MFCOM is used as a serial port receiver, properly decrease TMCVALUE in TMCMPx register, and increase the baud rate.
2) When MFCOM is used as a serial port receiver, set TMRST[2:0] to 0'b000 (never reset the timer) and TMSTOP[1:0] to 0b'00 (disable the stop bit).

2.7.2. When the baud rate is low under the condition that MFCOM is used as a UART receiver, the receiving of the data after the second frame encounters error

Description & impact

When the baud rate of the serial port is low, the valid start bit of the next data cannot be received due to TMEN signal loss during counting of the stop bit, which causes the error in receiving subsequent data.

Workarounds

1) When MFCOM is used as a serial port receiver, properly decrease TMCVALUE in TMCMPx register, and increase the baud rate.
2) When MFCOM is used as a serial port receiver, set TMRST[2:0] to 0'b000 (never reset the timer) and TMSTOP[1:0] to 0b'00 (disable the stop bit).

2.7.3. When MFCOM is configured in USART mode, if there is a deviation in the baud rate, synchronizing the baud rate can lead to errors in data transmission and reception

Description & impact

When MFCOM is used for USART functionality and there is a deviation in the baud rate, setting the TMRST to synchronize the USART baud rate may cause an increase in the data bit width, which lead to errors in data transmission and reception.

Workarounds

Avoid synchronizing the USART baud rate by setting the TMRST.

2.8. DBG

2.8.1. MCU cannot enter the debug mode from the standby mode after STB_HOLD bit in DBG register is set

Description & impact

MCU cannot enter the debug mode from the standby mode after STB_HOLD bit in DBG register is set.

Workarounds

Switch the system clock to internal IRC8M before MCU enters the standby mode.

2.8.2. MCU in the standby mode cannot be woken up after STB_HOLD bit in DBG register is set

Description & impact

MCU in the standby mode cannot be woken up after STB_HOLD bit in DBG register is set.

Workarounds

Switch the system clock to internal IRC8M before MCU enters the standby mode.

2.9. ADC

2.9.1. Over 5 V input voltage of PB13 pin results in incorrect voltage sampling of PD14 pin

Description & impact

As PB13 pin and PD14 pin belong to the same group of ADC multiplex channel (ADC1_IN15), over 5 V input voltage of PB13 pin results in electric leakage of PMOS on PB13 pin, then voltage increase of ADC1_IN15 to be close to 5 V, and finally PMOS breakover on PD14 pin, so that the voltage of PD14 pin is equal to that of ADC1_IN15.

The above case applies to any group of IO port with the same multiplex function among ADC.

Workarounds

Taking the IO port of PB13/PD14 pin as an example, increase the operating voltage of MCU so that the voltage of VDD/VDDA/VREF+ is equal to the maximum voltage on PB13 pin.

Note: As the reference voltage is changed, ADC conversion processing should be changed

accordingly in the software. In this case, users should evaluate the system impact of such change.

2.10. USART

2.10.1. Negative narrow pulse interference results in wrong data received by the serial port

Description & impact

Negative narrow pulse interference on the receiving data cable (Rx) of the serial port results in detection of wrong start bit by USART and data receiving error.

Workarounds

Avoid narrow pulse interference on the Rx line.

2.11. TIMER

2.11.1. CHxCAPFLT and CHxCAPPSC can be set only if the bit is set twice after converting from the output mode to the input mode in PROT mode 2

Description & impact

When complementary registers are set to PROT mode 2 for protection control and TIMER is converted from the output mode to the input mode, CHxCAPFLT and CHxCAPPSC bit fields can be set only if CMxMS bit in CHCTLx register is set twice.

Workarounds

Operation is not affected.

2.11.2. A constantly high or low level in a cycle of PWM is output when the duty ratio is over 50% in the composite PWM mode

Description & impact

A constantly high level in a cycle of PWM is output when the duty ratio is updated from over 50% to below 50% in composite PWM mode 0; a constantly low level in a cycle of PWM is output when the duty ratio is updated from below 50% to over 50% in composite PWM mode 1.

Workarounds

Disable the shadow register in software, and update the comparative threshold through DMA request.

2.12. CAN

2.12.1. Reset of USART0/USART1 peripheral results in communication error of CAN0/CAN1

Description & impact

Setting USART0RST bit in RCU_APB2RST register or USART1RST bit in RCU_APB1RST register results in operation failure of CAN0/CAN1 during CAN communication and data receiving error.

Workarounds

In the software, first initialize USART0/USART1 and then CAN0/CAN1, and do not reset initialized USART0/USART1.

2.12.2. CAN mailbox 0 converted from receiving mailbox to transmitting mailbox fails to send the data frame

Description & impact

CAN mailbox 0 converted from receiving mailbox to transmitting mailbox fails to send the data again after sending a frame of data.

Workarounds

In the software, convert the receiving mailbox to the transmitting mailbox after reading the data in the receiving mailbox.

2.12.3. As a transmitting node, CAN executes unexpected self-calibration function

Description & impact

When the delay (Tx-Rx readback time + 2*CK_CAN) is over a Tq, as a transmitting node, CAN executes self-calibration, which results in broadening of sent dominant level and communication error.

Workarounds

Use the GD32 MCU CAN transmission software solution, referring to “AN222 GD32A5x3 software evasion of CAN bit time problem”.

2.12.4. CAN manual bus off recovery function faults

Description & impact

When the CAN is in bus off state due to bus short or other bus exception, which cause bus to remain in a recessive state, after enabling the CAN automatic bus off recovery (ABORDIS) function, the bus off recovery flag (BORF) will be set. If CAN is still at bus off state at this time, the bus off flag (BOF) will be set again. If bus off interrupt (BOIE) and bus off recovery interrupt (BORIE) are enabled, the corresponding interrupt process will be entered ceaselessly.

Workarounds

Stop mailbox transmission before enabling and disabling automatic bus off recovery function. Taking CAN1 as an example, the following reference code can be called in manual bus off recovery scenario.

```
{
    /* stop mailbox transmission */
    transmit_message.code = CAN_MB_TX_STATUS_ABORT;
    can_mailbox_config(CAN1, 1, &transmit_message);
    /* enable then disable bus off recovery function */
    can_auto_busoff_recovery_enable(CAN1);
    can_auto_busoff_recovery_disable(CAN1);
}
```

2.12.5. CAN RAM area may be tampered in receiving mailbox processing

Description & impact

If the global mailbox unlocking operation is not performed in the receiving mailbox processing routine (due to incorrect operation in the software), there is a certain probability that the CAN RAM area will be tampered, which will cause data transmission and reception exceptions.

Workarounds

Wait for the sending completion flag in CAN_STAT register to be set instead of judging by the CODE segment value of the sent mailbox before every data transmission. The reference code is as follow:

```
/* Indicate whether the CAN node has transmitted frames or not after module reset */
FlagStatus can_tx_state = RESET;
{
    if((RESET == can_tx_state) || (SET == can_flag_get(CAN1, CAN_FLAG_MB1))){
        can_tx_state = SET;
        can_flag_clear(CAN1, CAN_FLAG_MB1);
        /* transmit message */
        can_mailbox_config(CAN1, 1, &transmit_message);
        /* user code */
```

```
    }
}
```

2.12.6. After Bus off recovery, transmit error count is not cleared automatically

Description & impact

After Bus off recovery, transmit error count (TECNT) is not cleared automatically by hardware.

Workarounds

After Bus off recovery, using software method to clear the transmit error count (TECNT). Taking CAN1 as an example, the reference code is as follow.

```
{
    /* bus off recovery flag is set */
    if(RESET != can_flag_get(CAN1, CAN_FLAG_BUSOFF_RECOVERY))
    {
        /* enter inactive mode */
        can_operation_mode_enter(CAN1, CAN_INACTIVE_MODE);
        /* clear transmit error count */
        CAN_ERR0(CAN1) &= ~(CAN_ERR0_TECNT);
        /* enter normal mode */
        can_operation_mode_enter(CAN1, CAN_NORMAL_MODE);
        can_flag_clear(CAN1, CAN_FLAG_BUSOFF_RECOVERY);
    }
}
```

2.12.7. 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.

2.12.8. When the CAN operating clock frequency is less than CK_PCLK2 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_PCLK2, 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 should configure CAN clock source as CK_PCLK2. Taking CAN1 as an example, the reference code is as follow:

```
{
    /* configure the CAN1 clock source as CK_PCLK2 */
    rcu_can_clock_config(CAN1, RCU_CANSRC_PCLK2);
}
```

2) Software checks the frame ID (applicable when the CAN operating clock frequency is less than CK_PCLK2). 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

3. Revision history

Table 3-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.20, 2023
1.1	Add GPIO limitation, referring to chapter 2.5.1	Jun.27, 2024
1.2	1. Add limitations of CAN, refer to ***CAN RAM area may be tampered in receiving mailbox processing***. 2. Update the workaround of FMC, refer to ***Power failure/reset results in MCU crash when write operation is performed for EEPROM***. 3. Add limitations of PMU, refer to ***When the MCU enters deep-sleep mode, the VDD of some 64-Pin package chip will increase the current consumption of about 200uA***. 4. Add limitations of MFCOM, refer to ***When MFCOM is configured in USART mode, if there is a deviation in the baud rate, synchronizing the baud rate can lead to errors in data transmission and reception***. 5. Update workarounds of CAN, refer to ***As a transmitting node, CAN executes unexpected self-calibration function***.	Jul.27, 2024
1.3	Add limitations of Rev. Code E	Oct.11, 2024
1.4	1. Delete limitations ***Power failure/reset results in MCU crash when write operation is performed for EEPROM*** due to not support hardware EEPROM function 2. Add limitations of System, refer to ***The on-chip SRAM boot mode is not supported*** and ***When the system clock switches between high and low frequencies frequently, the system may enter an abnormal state*** 3. Add limitations of CAN, refer to ***After Bus off recovery, transmit error count is not cleared automatically*** and ***The mailbox data is not read in time may result in incorrect data reading for the current and next frames*** and ***When the CAN operating clock frequency is less than CK_PCLK2 and an error occurs in DLC segment, which will lead to receiving***	Dec.25, 2024

	an incorrect frame ID	

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 under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. 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.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do 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 Products. Customers shall indemnify and hold the Company 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 Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.