

GigaDevice Semiconductor Inc.

Device limitations of GD32M531

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. UART	8
2.1.1. When UART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again	8
2.1.2. IDLEF is set when waking up UART from mute mode via an idle frame.....	8
2.2. I2C	8
2.2.1. When I2C operates as master in 10-bit address mode, failure to send a STOP signal after transmitting a frame of data causes subsequent data frame transmission errors.....	8
2.2.2. Abnormal transmission timing when I2C operates as master in 10-bit address receive mode with HEAD10R = 1	9
2.2.3. 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	9
2.2.4. When I2C works in 7/10 address slave mode, receiving an abnormal timing will cause the SDA line to be stuck.....	10
2.2.5. 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	10
2.3. SPI	11
2.3.1. 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, which will cause the slave CRC to continue working	11
2.3.2. When SPI operates as master or slave with CRC function enabled, the CRC register does not automatically clear after CRC transmission or verification is completed.....	11
2.4. CAN	12
2.4.1. When TFO is set to 1, after sequentially enabling 3 mailboxes for transmission, aborting the lowest priority transmit mailbox will also cause the second priority transmit mailbox to be aborted.....	12
2.4.2. When TFO is cleared and the identifiers of transmission mailbox 1 or 2 are configured as 0x1FFFFFFF, the data in transmission mailbox 1 or 2 cannot be sent	12
2.5. Core	12

2.5.1.	Access permission faults are prioritized over unaligned Device memory faults.....	13
3.	Revision history	14

List of Figures

Figure 1-1. Device revision code of GD32M531 6

List of Tables

Table 1-1. Applicable products	6
Table 1-2. Device limitations	6
Table 3-1. Revision history	14

1. Introduction

This document applies to GD32M531 product series, as shown in [Table 1-1. Applicable products](#). It offers technical guidance for using GD32 MCU and provides workaround to current device limitations.

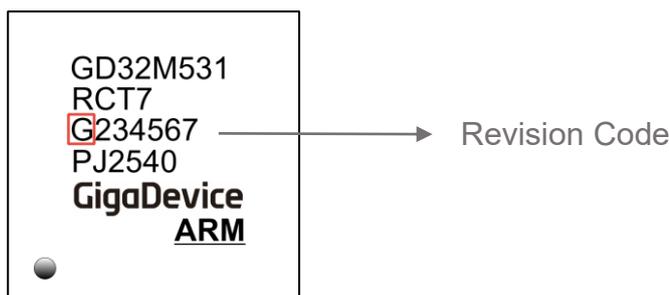
Table 1-1. Applicable products

Type	Part Numbers
MCU	GD32M531xx 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 GD32M531](#).

Figure 1-1. Device revision code of GD32M531



1.2. Summary of device limitations

The device limitations of GD32M531 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 G
UART	<i>When UART is woken up from mute mode by an idle frame, it will not be woken up when it enters mute mode again</i>	N
	<i>IDLEF is set when waking up UART from mute mode via an idle frame</i>	Y
I2C	<i>When I2C operates as master in 10-bit address mode, failure to send a STOP signal after transmitting a frame of data causes subsequent data frame transmission errors</i>	Y
	<i>Abnormal transmission timing when I2C operates as master</i>	Y

Module	Limitations	Workaround
		Rev. Code G
	<i>in 10-bit address receive mode with HEAD10R = 1</i>	
	<i>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</i>	Y
	<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>	Y
SPI	<i>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</i>	Y
	<i>When SPI operates as master or slave with CRC function enabled, the CRC register does not automatically clear after CRC transmission or verification is completed</i>	Y
CAN	<i>When TFO is set to 1, after sequentially enabling 3 mailboxes for transmission, aborting the lowest priority transmit mailbox will also cause the second priority transmit mailbox to be aborted</i>	Y
	<i>When TFO is cleared and the identifiers of transmission mailbox 1 or 2 are configured as 0x1FFFFFFF, the data in transmission mailbox 1 or 2 cannot be sent</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. UART

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

Workarounds

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

2.1.2. IDLEF is set when waking up UART from mute mode via an idle frame

Description & impact

When waking up UART 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. I2C

2.2.1. When I2C operates as master in 10-bit address mode, failure to send a STOP signal after transmitting a frame of data causes subsequent data frame transmission errors

Description & impact

When I2C operates as master and fails to send a STOP signal after transmitting a frame of data, and the software then modifies the configuration to master receive mode, the waveform of the master receive section will always be RESTART + 10-bit address head + Master

Receive, regardless of whether HEAD10R is configured as 0 or 1, rendering the HEAD10R configuration ineffective.

If I2C operates as master and fails to send a STOP signal after transmitting a frame of data without modifying the master transmission direction (remaining in transmit mode), when HEAD10R = 0, the subsequent RESTART will directly enter the master receive waveform (RESTART + 10-bit address head + Master Receive); when HEAD10R = 1, the subsequent RESTART will repeatedly send the first part of the master address sequence (RESTART + 10-bit address head).

Workarounds

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

2.2.2. Abnormal transmission timing when I2C operates as master in 10-bit address receive mode with HEAD10R = 1

Description & impact

When I2C operates as master in 10-bit address receive mode with HEAD10R = 1, the I2C master timing is START + 10-bit address head + Master Receive, which causes the slave to not ACK and makes it impossible to address the slave device. In 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.

2.2.3. 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.

2.2.4. 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.

2.2.5. 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.

2.3. SPI

2.3.1. 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, which will cause the slave CRC to 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).

2.3.2. When SPI operates as master or slave with CRC function enabled, the CRC register does not automatically clear after CRC transmission or verification is completed

Description & impact

When SPI operates as master or slave with CRC verification enabled, after CRC transmission or verification is completed (by setting CRCNT to send or receive CRC data and perform verification), the CRC registers (SPI_RCRC and SPI_TCRC) do not automatically clear. This causes the CRC calculation for the next frame of data communication to continue using the CRC value calculated from the previous frame, resulting in the CRCERR flag being set.

Workarounds

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

2.4. CAN

2.4.1. When TFO is set to 1, after sequentially enabling 3 mailboxes for transmission, aborting the lowest priority transmit mailbox will also cause the second priority transmit mailbox to be aborted

Description & impact

When all pending transmit mailboxes are sent in first-in-first-out order (TFO=1), such as the transmission sequence is 0->1->2. When aborting mailbox 2 transmission, if mailbox 1 is still in pending status, mailbox 1 will also be aborted, meaning the data in transmit mailbox 1 will not be sent out.

Workarounds

Use one of the following solutions:

- 1) Before aborting the lowest priority transmit mailbox, ensure that the second priority transmit mailbox is not in pending status.
- 2) After aborting the lowest priority transmit mailbox, reconfigure and transmit the second priority and lowest priority transmit mailbox.

2.4.2. When TFO is cleared and the identifiers of transmission mailbox 1 or 2 are configured as 0x1FFFFFFF, the data in transmission mailbox 1 or 2 cannot be sent

Description & impact

When TFO is cleared and the identifiers of transmission mailbox 1 or 2 are configured as 0x1FFFFFFF, the data in transmission mailbox 1 or 2 cannot be sent. For example, when TFO is set to 0 and the ID of transmission mailbox 1 is configured as 0x1FFFFFFF, the data in transmission mailbox 1 cannot be sent.

Workarounds

Use one of the following solutions:

- 1) When the identifier is 0x1FFFFFFF, use transmission mailbox 0 for data frame transmission.
- 2) Use first-in-first-out (FIFO) transmission order, i.e., set TFO = 1.

2.5. 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.

2.5.1. 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.3 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.