



MAX32665–MAX32668 ERRATA SHEET

Revision A3 Errata

The errata listed below describe situations where components of this revision perform differently than expected or differently than described in the data sheet. Maxim Integrated Products, Inc. may, at its own discretion, take future steps to correct these errata when the opportunity to redesign the product presents itself. Prior to that, Maxim has determined the following potential workarounds that customers may want to consider when addressing one of the situations described below.

This errata sheet only applies to components of this revision. These components are branded on the topside of the package with a six-digit code in the form yywwRR, where yy and ww are two-digit numbers representing the year and work week of manufacture, respectively, and RR is the revision of the component. To obtain an errata sheet on other die revisions, visit the Maxim website at www.maximintegrated.com/errata.

1) SPIXF BUS IDLE FEATURE CANNOT BE DISABLED

Description:

The SPIXF bus idle feature cannot be disabled. The SPIXF will unexpectedly force the start of a new transaction, requiring resending of the SPI transaction header, if two consecutive reads of the cache both generate a cache miss. (14358)

Workaround:

Set SPIXF_BUS_IDLE.busidle to 0x1 to approximate the operation of the SPIXF when the bus idle feature is disabled. This will have minimal effect on sequential code execution.

2) EXTERNAL CAPACITORS REQUIRED FOR RTC OPERATION

Description:

External capacitors are required to meet the loading requirements for RTC accuracy. (14416)

Workaround:

Specific values are based on PCB and internal oscillator capacitances. Contact factory for details on the characterization procedure.

3) SPI MODE 1 AND MODE 3 OPERATION MAY BE AFFECTED WHEN QSPIn_CLK_CFG.SCALE = 0

Description:

The following settings are invalid when operation the SPI is operating in mode 1 or mode 3: (13390)

SCALE=0, CLOCKHI=0, CLOCKLO=0
SCALE=0, CLOCKHI=1, CLOCKLO=1

Workaround:

Do not use the invalid settings. The operating speeds generated by all other field combinations are valid.

MAX32665–MAX32668

REV A3 ERRATA

4) UART RECEIVER REQUIRES ONE EXTRA BIT TIME BETWEEN STOP AND START BITS

Description:

The UART requires an external transmitter to send at least one more stop bit than specified in the UART_CTRL0.stop field. (10650)

Workaround:

There are two workarounds:

- 1) Configure the transmitter to send at least one additional stop bit than specified in the UART_CTRL0.stop field.
- 2) Ensure the transmitter generates at least one idle bit time between byte transmissions.

5) DO NOT USE UART Tx AND Rx FUNCTIONALITY SIMULTANEOUSLY

Description:

The peripheral does not operate as expected if both the receive and transmit functions are used simultaneously. (10685)

Workaround:

There are two workarounds:

- 1) Use flow control to prevent an external UART from transmitting while the peripheral is transmitting.
- 2) Assign the transmit and receive functionality to different peripherals if full-duplex operation is required.

6) UART TRANSMITTER GENERATES SPURIOUS PULSE WHEN USING 7.3728MHz CLOCK SOURCE

Description:

The UART generates a negative pulse 2 PCLK periods wide on the UART_TX line 1 bit period before the falling edge of the start bit. (10689)

Workaround:

There are two workarounds:

- 1) Use PCLK as the clock source when transmitting.
- 2) Ensure the external receiver will reject the spurious pulse.

7) UART Tx BAUD RATE FORMULA INCORRECT

Description:

The baud rate formula supplied in the User Guide must be modified to achieve the required accuracy for data transmission. The baud rate formula for data reception is correct as written in the User Guide. (10682)

Workaround:

First, calculate the ibaud and dbaud values using the formula supplied in the User Guide, then apply the following correction. Note that the term UARTn_BAUD0.clkdiv refers to its decoded value (128, 64, 32, 16, 8).

```
ddiv_temp = ((128 / UARTn_BAUD0.clkdiv) * UARTn_BAUD0.ibaud) + UARTn_BAUDn.dbaud;  
ddiv_new = ddiv_temp % 128;  
idiv_new = UARTn_BAUD0.ibaud + floor(ddiv_temp / 128);
```

```
UARTn_BAUD1.dbaud = ddiv_new  
UARTn_BAUD0.ibaud = idiv_new
```

MAX32665–MAX32668

REV A3 ERRATA

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	12/20	Initial release	—
1	8/21	Added errata 4-7	2, 3

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

© 2021 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
One Analog Way, Wilmington, MA 01887 U.S.A. | Tel: 781.329.4700 | © 2021 Analog Devices, Inc. All rights reserved.