

# N32G451 Series Errata Sheet

## Table of Contents

<b>1 ERRATA LIST</b> .....	<b>4</b>
<b>2 POWER CONTROL (PWR)</b> .....	<b>6</b>
2.1 STOP2 MODE WAKEUP .....	6
<b>3 RESET AND CLOCK CONTROL (RCC)</b> .....	<b>7</b>
3.1 SYSTEM TIMER (SYSTICK) .....	7
<b>4 GPIO AND AFIO</b> .....	<b>7</b>
4.1 SPI1 SLAVE MODE, USART2 SYNC MODE .....	7
4.2 SPI1 MASTER MODE, USART2 SYNC MODE .....	7
4.3 SPI2 SLAVE MODE, USART3 SYNC MODE .....	8
4.4 SPI2 MASTER MODE, USART3 SYNC MODE .....	8
<b>5 ANALOG/DIGITAL CONVERSION (ADC)</b> .....	<b>9</b>
5.1 ADC DATA LEFT ALIGNMENT.....	9
5.2 ADC ANALOG WATCHDOG.....	9
5.3 ADC INJECTED CHANNEL TRIGGERS REGULAR CHANNEL CONVERSION .....	9
5.4 SLAVE ADC CONVERSION IMPACTED BY MASTER ADC CONVERSION .....	10
5.5 AFFECTING ADJACENT ADC DATA REGISTERS.....	10
<b>6 SERIAL PERIPHERAL INTERFACE (SPI)</b> .....	<b>11</b>
6.1 SPI INTERFACE.....	11
6.1.1 SPI Baud Rate Setting .....	11
6.1.2 Checking Slave Mode CRC .....	11
6.2 I2S INTERFACE.....	12
6.2.1 PCM Long Frame Mode .....	12
<b>7 I2C INTERFACE</b> .....	<b>13</b>
7.1 HANDLING SOFTWARE EVENTS BEFORE CURRENT BYTE TRANSFERRING.....	13
7.2 NOTES ON SINGLE READ OF ONE OR TWO BYTES.....	13
7.3 USING DMA SIMULTANEOUSLY WITH OTHER PERIPHERALS .....	14
<b>8 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER</b> .....	<b>15</b>
8.1 PARITY ERROR FLAG .....	15
8.2 RTS HARDWARE FLOW CONTROL.....	15

<b>9 DEBUG INTERFACE (DBG)</b> .....	<b>16</b>
9.1 DEBUG REGISTER.....	16
<b>10 TIMER (TIM)</b> .....	<b>17</b>
10.1 TIMER REPETITIVE CAPTURE DETECTION.....	17
<b>11 REAL TIME CLOCK (RTC)</b> .....	<b>18</b>
11.1 RTC PRESCALER.....	18
11.2 RTC CALIBRATION .....	18
11.3 RTC TIMING .....	18
11.4 RTC WAKEUP .....	19
<b>12 CHIP MARKING AND REVISION DESCRIPTION</b> .....	<b>20</b>
<b>13 VERSION HISTORY</b> .....	<b>21</b>
<b>14 DISCLAIMER</b> .....	<b>22</b>

# 1 Errata List

Table 1-1 Overview of Errata

Errata Link		Chip Version	
		B	
<a href="#">Ch 2: Power Control (PWR)</a>	<a href="#">2.1: Stop2 mode wakeup</a>	•	
<a href="#">Ch 3: Reset and Clock Control (RCC)</a>	<a href="#">3.1: System Timer (SysTick)</a>	•	
<a href="#">Ch 4: GPIO and AFIO</a>	<a href="#">4.1: SPI1 slave mode, USART2 sync mode</a>	•	
	<a href="#">4.2: SPI1 master mode, USART2 sync mode</a>	•	
	<a href="#">4.3: SPI2 slave mode, USART3 sync mode</a>	•	
	<a href="#">4.4: SPI2 master mode, USART3 sync mode</a>	•	
<a href="#">Ch 5: Analog/Digital Conversion (ADC)</a>	<a href="#">5.1: ADC Data Left Alignment</a>	•	
	<a href="#">5.2: ADC Analog Watchdog</a>	•	
	<a href="#">5.3: ADC Injected Channel Trigger Rule Channel Conversion</a>	•	
	<a href="#">5.4: Starting from ADC Conversion Affected by Master ADC Conversion</a>	•	
	<a href="#">5.5: Adjacent ADC Data Register Affected</a>	•	
<a href="#">Ch 6: Serial Peripheral Interface (SPI)</a>	<a href="#">6.1: SPI interface</a>	<a href="#">6.1.1: SPI Baud Rate Setting</a>	•
		<a href="#">6.1.2: 从 Slave Mode CRC Check</a>	•
	<a href="#">6.2: I2Sinterface</a>	<a href="#">6.2.1: PCM Long Frame Mode</a>	•
<a href="#">Ch 7: I2C Interface</a>	<a href="#">7.1: Software Events to be Managed Before Current Byte Transfer</a>	•	

	<a href="#">7.2: Considerations When Reading Single or Double Bytes Once</a>	•
	<a href="#">7.3: Using DMA Simultaneously with Other Peripherals</a>	•
<a href="#">Ch 8: Universal Synchronous Asynchronous Receiver (USART)</a>	<a href="#">8.1: Parity Error Flag</a>	•
	<a href="#">8.2: RTS Hardware Flow Control</a>	•
<a href="#">Ch 9: Debug Interface (DBG)</a>	<a href="#">9.1: Debug Register</a>	•
<a href="#">Ch 10: Timer (TIM)</a>	<a href="#">10.1: Timer Repeat Capture Detection</a>	•
<a href="#">Ch 11: Real-Time Clock (RTC)</a>	<a href="#">11.1: RTC Presale</a>	•
	<a href="#">11.2: RTC Calibration</a>	•
	<a href="#">11.3: RTC Timing</a>	•
	<a href="#">11.4: RTC Wakeup</a>	•

## 2 Power Control (PWR)

### 2.1 Stop2 Mode Wakeup

Description:

When the MCU is in Stop2 mode, wakeup and NRST reset occurs simultaneously, the NRST reset cannot reset the MCU.

Wakeup takes precedence, and the MCU will respond to the wakeup event first.

Workaround:

To resolve this issue, it is advised to avoid triggering an NRST reset of the MCU simultaneously with the wakeup event. Alternatively, if an NRST reset is necessary in a specific scenario, perform consecutive NRST resets to ensure the MCU is properly reset.

## 3 Reset and Clock Control (RCC)

### 3.1 System Timer (Systick)

Description:

The MCU can't be awakened when external clock source (STCLK) is chosen as clock source.

Workaround:

To address this issue, configure the SysTick control register to use the core clock as the clock source instead of the External Reference Clock (STCLK).

## 4 GPIO and AFIO

### 4.1 SPI1 Slave mode, USART2 Sync Mode

Description:

With SPI1 and USART2 clocks already enabled, when pin PA4 is configured as a multiplexed output, SPI1 operating in slave mode with NSS software mode (SSMEN=1, SSEL=0), the USART2 clock in synchronous mode cannot be transmitted.

Workaround:

No solution is provided for this issue.

### 4.2 SPI1 Master Mode, USART2 Sync Mode

Description:

With SPI1 and USART2 clocks already enabled, when pin PA4 is configured as a multiplexed output, SPI1 operating in master mode with NSS software mode (SSMEN=1, SSEL=0), the USART2 clock in synchronous mode cannot be transmitted.

Workaround:

Enable the SSOEN bit in SPI1 master mode.

### **4.3 SPI2 Slave Mode, USART3 Sync Mode**

Description:

With SPI2 and USART3 clocks already enabled, when pin PB12 is configured as a multiplexed output, SPI2 operating in slave mode with NSS software mode (SSMEN=1, SSEL=0), the USART3 clock in synchronous mode cannot be transmitted.

Workaround:

No solution is provided for this issue.

### **4.4 SPI2 Master Mode, USART3 Sync Mode**

Description:

With SPI2 and USART3 clocks already enabled, when pin PB12 is configured as a multiplexed output, SPI2 operating in master mode with NSS software mode (SSMEN=1, SSEL=0), the USART3 clock in synchronous mode cannot be transmitted.

Workaround:

Enable the SSOEN bit in SPI2 master mode.



## 5 Analog/Digital Conversion (ADC)

### 5.1 ADC Data Left Alignment

Description:

In the ADC single conversion mode, when using a non-12-bit precision with left alignment, and triggering the conversion of regular channels by software, the most significant bit of the invalid bits in the ADC\_DAT register is 1.

Workaround:

To address this issue, either retain only the valid data bits or switch to right alignment mode.

### 5.2 ADC Analog Watchdog

Description:

In ADC independent mode, with single conversion, and non-12-bit precision, enabling the analogue watchdog feature, and triggering the conversion of regular/injected channels by software, if the high threshold value of the analogue watchdog has valid bits equal to the value in the ADC data register, with all invalid bits set to 0, it may lead to a false triggering of the analogue watchdog.

Workaround:

In this situation, ensure that the most significant bit of the invalid bits in the high threshold value of the analogue watchdog is set to 1 to prevent unintended triggering.

### 5.3 ADC Injected Channel Triggers Regular Channel Conversion

Description:

In continuous conversion mode, with external triggering disabled for regular channels and only software-triggered injection channel conversions, it is possible that the regular channels may be inadvertently triggered, leading to data being generated in the ADC\_DAT register, and the corresponding status bit in ADC\_STS for regular channel conversions being set.

Workaround:

Ignore the flags and data generated by regular channel conversions.

## **5.4 Slave ADC Conversion Impacted by Master ADC Conversion.**

Description:

When the ADC operates in dual ADC mode and synchronous injected mode, and only the software triggers the regular channel conversion of the master ADC, the regular channels of the slave ADC are also triggered. Additionally, the lower 16 bits from the ADC\_DAT of the slave ADC are merged into the upper 16 bits of the master ADC\_DAT.

Workaround:

No solution is provided for this issue.

## **5.5 Affecting Adjacent ADC Data Registers**

Description:

In independent mode, when software triggers the conversion of the ADC4/2 regular channels, the lower 16 bits of the ADC4/2DAT register content are merged into the upper 16 bits of the ADC3/1DAT register.

Workaround:

No solution is provided for this issue.

## 6 Serial Peripheral Interface (SPI)

### 6.1 SPI Interface

#### 6.1.1 SPI Baud Rate Setting

Description:

CRC validation may fail if SPI master mode, when setting the bitrate control bits (BR[2:0]) to fPLCK/2

Workaround:

Avoid setting the bitrate control bits (BR[2:0]) to fPLCK/2 in this scenario.

#### 6.1.2 Checking Slave Mode CRC

Description:

In SPI operating in slave mode with CRC validation enabled, even if the NSS pin is at a high level, CRC calculation continues if the SPI receives a clock signal.

Workaround:

Before using CRC validation, clear the CRC data register to ensure synchronization between the CRC checks of the master and slave devices.

The clearing steps are as follows:

Reset the SPI enable bit (set to 0).

Reset the CRC validation bit (set to 0).

Set the CRC validation bit (set to 1).

Set the SPI enable bit (set to 1).

## 6.2 I2S Interface

### 6.2.1 PCM Long Frame Mode

Description:

When I2S is operating in master mode, PCM long frame mode, and the data format is extended from 16-bit to 32-bit or is directly set to 32-bit, the WS (word select) signal occurs every 16 bits instead of every 32 bits.

Workaround:

If I2S must operate in master mode and use long frame mode, it is recommended to use only the 16-bit data mode to ensure proper WS signal alignment.

## 7 I2C Interface

### 7.1 Handling Software Events Before Current Byte Transferring

Description:

In the occurrence of events EV7, EV7\_1, EV6\_1, EV6, EV2, EV8, and EV3, it is essential to handle the events before the current byte transfer to prevent issues such as reading an extra byte, obtaining duplicate data, or losing data. If the software fails to read the N-1 data before the stop signal generation, the data in the shift register for the Nth byte may become corrupted (shifted left by one bit).

Workaround:

When transferring more than one byte in I2C, consider using DMA if possible.

When using I2C interrupts, set the interrupt priority to the highest priority in the application.

When the read data reaches the N-1 byte:

- a) Check that BSF (Byte Shift Flag) is set to 1.
- b) Configure SCL as a GPIO open-drain output and set it to 0.
- c) Set STOPGEN to 1.
- d) Read the N-1 byte.
- e) Configure SCL back to I2C multiplex open-drain output mode.
- f) Read the last byte.

### 7.2 Notes on Single Read of One or Two bytes

Description:

In master read mode, errors in reading data may occur when reading single-byte and double-byte lengths.

Workaround:

Single byte Read:

- a) Upon receiving ADDR.F.
- b) Set ACKEN to 0.

- c) Clear ADDRFB bit (by reading STS1 first and then STS2).
- d) Set STOPGEN to 1.
- e) Read one byte of data.

Double byte Read:

- a) Upon receiving ADDRFB.
- b) Set ACKPOS to 1.
- c) Clear ADDRFB bit (by reading STS1 first and then STS2).
- d) Set ACKEN to 0.
- e) Check that BSF (Byte Shift Flag) is 1.
- f) Set STOPGEN to 1.
- g) Read two consecutive bytes of data.

### 7.3 Using DMA Simultaneously with Other Peripherals

Description:

During I2C communication using DMA, it may lead to abnormal I2C communication, if there are other peripherals also use the same DMA controller.

Workaround:

1. Use Different DMA Controllers.
2. Disable DMA for Other Peripherals During I2C DMA Communication.

## 8 Universal Synchronous Asynchronous Receiver Transmitter

### 8.1 Parity Error Flag

Description:

During the reception of a byte of data, if a parity error is detected before receiving the stop bit, the parity error flag is set. During this period, the parity error flag cannot be cleared through software (by reading the status register and then reading the data register). If the parity error interrupt is enabled, it may repeatedly enter the parity error interrupt handling function.

Workaround:

1. When the read data buffer flag is set and data is received, clear the parity error flag.
2. Disable the parity error interrupt during the first entry into the parity error interrupt, then re-enable it after receiving the data.

### 8.2 RTS Hardware Flow Control

Description:

When the RTS (Request to Send) hardware flow control is enabled, and a USART receives a frame of data, the RTS signal is automatically asserted (pulled high) upon receiving the first byte of data. If this first byte is not promptly read from the data register, upon receiving the next byte of data, the RTS signal is de-asserted (pulled low) again, and the USART waits for the reception of the next frame of data.

Workaround:

Read the data from the data register promptly after receiving the first byte.

## 9 Debug Interface (DBG)

### 9.1 Debug Register

Description:

The DBGMCU\_IDCODE debugging register can only be accessed in debug mode (not accessible to user programs). When attempting to read it in user mode, the returned value is 0xFF.

Workaround:

Avoid using IDCODE in user applications to prevent issues related to its restricted access in non-debug modes.



## 10 Timer (TIM)

### 10.1 Timer Repetitive Capture Detection

Description:

When an input capture occurs, if a new input capture is generated during the reading of TIMx\_CCxDATx (capture/compare register x) – where the read operation automatically clears the capture flag – the CCxOCF (capture/compare x repeated capture flag) might still remain set.

Workaround:

No solution is provided for this issue.

## 11 Real Time Clock (RTC)

### 11.1 RTC Prescaler

Description:

The asynchronous prescaler factor (DIVA[6:0]) and synchronous prescaler factor (DIVS[14:0]) in the RTC cannot be set to 0, as it may lead to RTC pre-division failure.

Workaround:

Avoid setting the asynchronous prescaler register (TRC\_PRE) DIVA[6:0] (asynchronous prescaler segment) and DIVS[14:0] (synchronous prescaler segment) to 0.

### 11.2 RTC Calibration

Description:

During RTC automatic calibration (when the CP bit in the RTC\_CALIB register is set to 1), the automatic calibration may not successful if the asynchronous prescaler factor (DIVA) is not set to 128/64/32/16/8.

Workaround:

To ensure successful RTC automatic calibration, choose an asynchronous prescaler (DIVA) factor of 128/64/32/16/8.

### 11.3 RTC Timing

Description:

While the RTC is active, if an NRST (external reset) occurs, the reset period will cause the RTC to stop counting.

Workaround:

No solution is provided for this issue.

## 11.4 RTC Wakeup

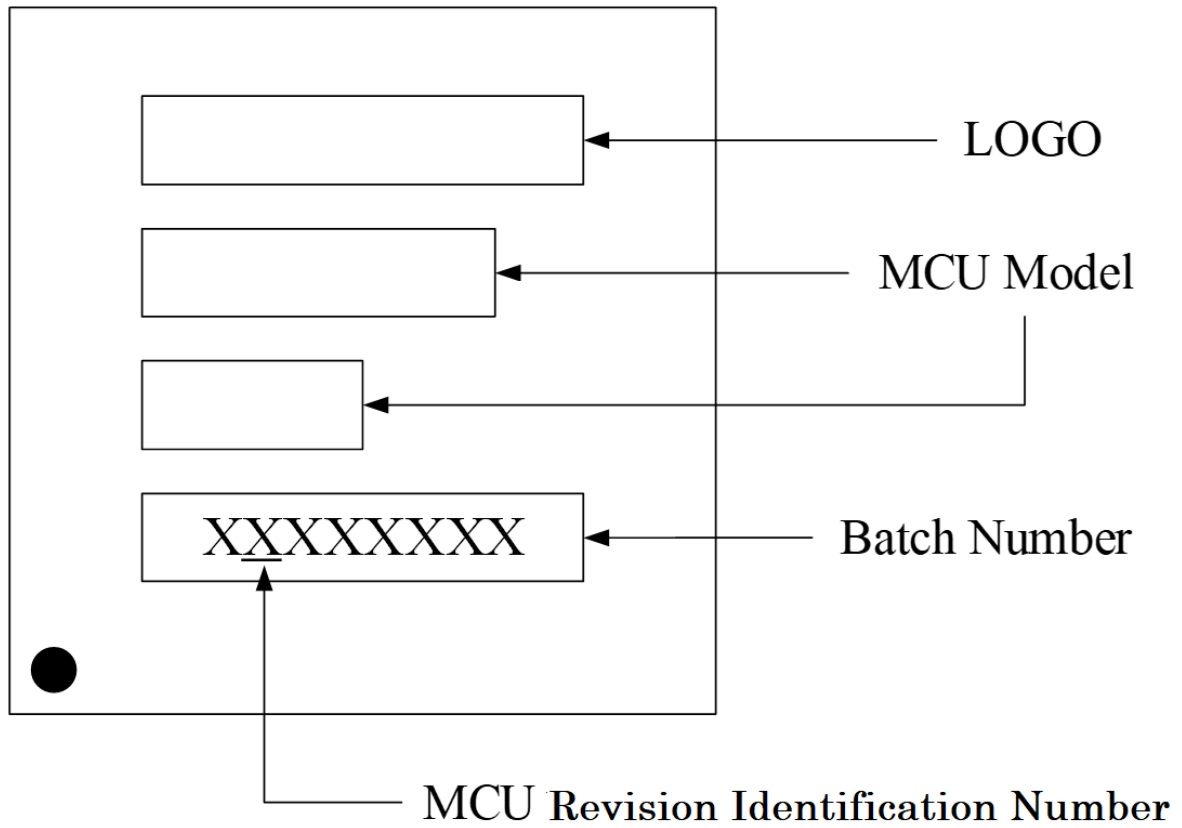
Description:

Periodic wake-up from the RTC module does not work to wake up from Standby mode.

Workaround:

Use the RTC alarm wake-up feature instead of the periodic wake-up.

## 12 Chip Marking and Revision Description



## 13 Version history

Version	Date	Changes
V1.0.0	2023.05.30	Initial release

## 14 Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD.(Hereinafter referred to as NSING).

This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to Nations Technologies Inc. and Nations Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders.

Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document. 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. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, 'Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.