N32G435 series errata sheet V1.3

Contents

1 Errata list

Table 1-1 Overview of errata

Errata link			Chip version					
			Version B	Version C	Version D	Version E	Version G	Version H
Chapter 2: Power Control (PWR)	Section 2.1: System clock switchover		●					
	Section 2.2: Switch to Stop2 mode from LPRUN mode		●	●	●	●	●	●
	Section 2.3: Power on again after power down		●	●				
Chapter 3: Reset and clock control (RCC)	Section 3.1: Enter Stop2 mode from Run mode		●	●	●	●	●	●
	Section 3.2: LSE-CSS fault detection		●	●	●	●	●	●
Chapter 4: GPIO and AFIO	Section 4.1: GPIO analog function		●	●				
Chapter 5: Analog/Digital conversion (ADC)	Section 5.1: ADC data left-align		●	●	●	●	●	●
	Section 5.2: ADC injection channels trigger regular channel conversions		●	●	●	●	●	●
	Section 5.3: ADC analog watchdog		●	●	●	●	●	●
	Section 5.4: DMA moves ADC data		●	●	●	●	●	●
Chapter 6: Serial peripheral interface (SPI)	Section 6.1: SPI interface	Section 6.1.1: SPI Baud Rate Setting	●	●	●	●	●	●
		Section 6.1.2: CRC check slave mode	●	●	●	●	●	●

		Section 6.1.3: SPI CLK GPIO configuration	●	●	●	●	●	●
	Section 6.2: I2S interface	Section 6.2.1: PCM long frame mode	●	●	●	●	●	●
Chapter 7: I2C interface	Section 7.1: Software events that must be managed before the current byte transfer		●	●	●	●	●	●
	Section 7.2: Considerations when reading single or double bytes at a time		●	●	●	●	●	●
	Section 7.3: Use DMA in conjunction with other peripherals		●	●	●	●	●	●
	Section 7.4: Abnormal signal interference		●	●	●	●	●	●
Chapter 8: Universal synchronous asynchronous receiver (USART)	Section 8.1: Check error flag		●	●	●	●	●	●
	Section 8.2: RTS hardware flow control		●	●	●	●	●	●
Chapter 9: Debug interface (DBG)	Section 9.1: Debug register		●	●	●	●	●	●
Chapter 10: Timer (TIM)	Section 10.1: Timer repeat capture detection		●	●	●	●	●	●
Chapter 11: Comparator (COMP)	Section 11.1: Comparator INP input			●	●	●	●	●
Chapter 12: Real time clock (RTC)	Section 12.1: RTC prescale		●	●	●	●	●	●
	Section 12.2: RTC subpicosecond interrupt		●	●	●	●	●	●
	Section 12.3: RTC interrupt		●	●	●	●	●	●
	Section 12.4: RTC Auto Wake-up		●	●	●	●	●	●

	Section 12.3: RTC mistakenly triggers TISOVF flag bit	●	●	●	●	●	●
Chapter 13: Low Power Timer (LPTIM)	Section 13.1: LPTIM maximum count value	●	●	●	●	●	●
Chapter 14: Controller Area Network (CAN)	Section 14.1: CAN active error	●	●	●	●	●	

2 Power control (PWR)

2.1 System clock switchover

Description

When you select HSI/HSE as the system clock and then switch to MSI as the system clock, the chip may break down if a system reset occurs at this time.

Workaround

You are advised to use PLL as the system clock instead of HSI/HSE.

2.2 Switch to Stop2 mode from LPRUN mode

Description

When the MCU switches from LPRUN mode to Stop2 mode, a reset may occur after wakeup.

Workaround

To solve this problem, ensure that PVDBOR is normally on in Stop2 mode (clear the PBDTLPR bit of PWR_CTRL3 to 0) before entering Stop2 mode.

2.3 Power on again after power down

Description

If the power supply is powered off and drops to the range of 600 mv to 100mV, and then power on again, the power on may be unsuccessful.

Workaround

When the MCU is powered off, ensure that the VDD voltage of the chip is below 100mV before powering on the MCU.

3 Reset and clock control (RCC)

3.1 Enter Stop2 mode from Run mode

Description

Under a certain AHB CLK and APB CLK frequency division, the Run mode is extremely unlikely to enter Stop2 mode, and the larger the frequency division is, the more likely it is to occur.

Workaround

Before entering Stop2 mode, back up the AHB CLK frequency division coefficient, set the AHB CLK frequency neutral mode, and then enter Stop2 mode. When exiting Stop2 mode, restore the original frequency division.

3.2 LSE-CSS fault detection

Description

After the LSE-CSS detects that the LSE is stopped, it cannot switch to the LSI through software.

Workaround

Power cycle.

4 GPIO and AFIO

4.1 GPIO analog function

Description

When the four GPIO PA1/PA2/PA3/PA4 switch to analog function in the state of high output level, there will be a voltage drop of about 30mV output voltage in the switching process.

Workaround

Avoid the above methods.

5 Analog/Digital conversion (ADC)

5.1 ADC data left-align

Description

ADC single conversion mode, non-12bit precision and left aligned, the software triggers the conversion rule channel, in the ADC_DAT register, the highest invalid bit is 1.

Workaround

Retain only valid data bits or use right-aligned mode.

5.2 ADC injection channels trigger regular channel conversions

Description

When the software triggers the injection channel conversion, the regular channel conversion may be started. As a result, data is generated in ADC_DAT, and the corresponding status bit of ADC_STS regular channel conversion will be set.

Workaround

Flag bits and data generated by regular channels are ignored.

5.3 ADC analog watchdog

Description

When the ADC works in independent mode and converts once and the accuracy is not 12bit, the analog watchdog function is enabled, and the software triggers the conversion rule channel/injection channel. The effective bits of the analog watchdog high threshold value are set equal to the value of the ADC data register. When the invalid bits are all 0, the analog watchdog may be triggered by mistake.

Workaround

In this case, the highest position 1 of the invalid bit of the simulated watchdog high threshold is not triggered.

5.4 DMA moves ADC data

Description

When the number of ADC moves is completed, DMA is disabled first, then ADC is disabled, and DMA continues to be enabled. In the case that ADC is not enabled, the first data that DMA moves will occur is the ADC conversion data left before DMA is disabled.

Workaround

In this case, if loop enable and disable DMA, add 1 quantity on the basis of move quantity N. After the first disabled, select read DMA move data from the 1 to N, and select read DMA move data from the 2 to N +1 in the subsequent disabled.

6 Serial peripheral interface (SPI)

6.1 SPI interface

6.1.1 SPI baud rate setting

Description

When the baud rate control bit (BR[2:0]) is set to fPLCK/2 in SPI master mode, CRC check will fail.

Workaround

In this case, avoid setting baud rate control bit (BR[2:0]) to fPLCK/2.

6.1.2 CRC check slave mode

Description

The SPI operates in slave mode and CRC verification is enabled. Even if the NSS pin is high, CRC calculations are performed whenever the SPI receives a clock signal

Workaround

Before using CRC check, clear the CRC data register to synchronize the CRC check between the master and slave devices

The clearing steps are as follows:

1. Reset the SPI enable bit (set to 0)
2. Reset the CRC check bit (set to 0)
3. Set the CRC check bit (set to 1)
4. Set the SPI enable bit (set to 1)

6.1.3 SPI CLK GPIO configuration

Description

When the SPI clock polarity is configured to be high, the CLK GPIO will be pulled high after the SPI is enabled, and then the clock signal will be output. The slave may misjudge this edge as a clock signal, causing the data received by the slave to shift.

Workaround

Before the SPI is enabled, configure the CLK GPIO according to the clock polarity (when the clock polarity is high, configure the CLK GPIO as a pull-up, and when the clock polarity is low, configure the CLK GPIO as a pull-down)

6.2 I2S interface

6.2.1 PCM long frame mode

Description

When the I2S work in master mode, PCM long frame mode, and the data format is 16bit extended to 24bit or 32bit, the WS signal is cycles every 16bit instead of 24bit or 32bit.

Workaround

When I2S is in master mode and long frame mode must be used, 16bit data mode should be used.

7 I2C interface

7.1 Software events that must be managed before the current byte transfer

Description

When EV7, EV7_1, EV6_1, EV6, EV2, EV8, and EV3 events occur, the events must be processed before the current byte is transferred. Otherwise, one more byte may be read, duplicate data may be read, or data may be lost.

If the data N-1 is not read by the software before the stop signal is generated, the data N in the shift register is corrupted (moved one bit to the left).

Workaround

1. Use DMA when transferring more than one byte using I2C

2. When using I2C interrupts, the interrupt priority is set to the highest priority of the application

3. When the read data reaches the N-1 byte:

a) Check BSF is 1

b) Set SCL to GPIO open miss output and set it to 0

c) Set STOPGEN to 1

d) Read the N-1 byte

e) Set SCL to open/miss output mode for I2C multiplexing

f) Read the last byte

7.2 Considerations when reading single or double bytes at a time

Description

In host read mode, data read errors may occur when the bytes read are single or double bytes.

Workaround

1. Single byte reading:

a) Upon receipt of ADDRF

b) Set the ACKEN bit to 0

c) Clear the ADDRF bit (by reading STS1 and then STS2)

d) Set STOPGEN to 1

e) Read one byte of data.

2. Double-byte read:

a) Upon receipt of ADDRF

b) Set the ACKPOS bit to 1

c) Clear the ADDRF bit (by reading STS1 and then STS2)

d) Set the ACKEN bit to 0

e) The BSF level was 1

f) Set STOPGEN to 1

g) Read two bytes of data in a row

7.3 Use DMA in conjunction with other peripherals

Description

If other peripherals are using DMA during the I2C communication, the I2C communication will be abnormal.

Workaround

I2C turns off DMA for other peripherals during DMA communication.

7.4 Abnormal signal interference

Description

In the working process of I2C, SCL and SDA may be interfered by burrs in the communication process, resulting in abnormal communication.

Workaround

Master and slave automatic recovery:

A) Soft reset of I2C by I2C CTRL1[15](SWRST) bit

B) Restore the IIC module by controlling the RCC_APB1RSTR[21](I2CxRST) bit in the RCC module to complete the recovery

C) Restore the IIC module by calling the global soft reset NVIC_SystemReset() function for global reset

Master recovery slave machine:

A) Restore the slave machine by controlling the hardware reset pin reset the slave machine

B) Use the power management mechanism to power on the slave machine for restoration

C) Set the communication port of the IIC master to GPIO mode, and send 9 clocks on the SCL CLOCK line to recover the slave machine

8 Universal synchronous asynchronous receiver (USART)

8.1 Check error flag

Description

During the receipt of a byte of data, a checksum error is detected before the stop bit is received, and the checksum error flag bit is set, during which the checksum error flag bit cannot be cleared by software (read status register, read data register again). If checksum interrupt is enabled, the checksum interrupt handler will be entered several times.

Workaround

The read buffer flag bit is set, and the error flag bit operation is performed after the data is received.

If checksum error interrupt is enabled, to avoid entering the interrupt processing function for multiple times, the checksum error interrupt is disabled when entering the checksum error interrupt for the first time. After receiving data, the checksum error interrupt is enabled again.

8.2 RTS hardware flow control

Description

When the RTS hardware flow control is enabled, the USART receives a frame of data. When the first byte of data is received, the RTS signal is automatically pulled up. If the first byte of data is not read out of the data register in time, the RTS signal is pulled down again after the next byte of data is received, and the USART waits for the next frame of data to be received.

Workaround

Read the data from the data register in time before receiving the next new data.

9 Debug Interface (DBG)

9.1 Debug register

Description

The DBGMCU_IDCODE debug register can only be accessed in debug mode (not by user programs), and the value returned by reading in user mode is 0xFF.

Workaround

Avoid using IDCODE in user applications.

10 Timer (TIM)

10.1 Timer repeat capture detection

Description

When an input capture is generated, if a new input capture is generated during reading of the TIMx_CCDATx (capture/compare register x) (the read operation automatically clears the capture flag bit), CCxOCF(capture/compare x repeat capture flag) may still be set.

Workaround

None

11 Comparator (COMP)

11.1 Comparator INP input

Description

PA0/PA1/PA3 as a digital function pin and high power level will affect the INP input of COMP1.

Workaround

When using COMP1, do not use PA0/PA1/PA3 as digital function pins.

12 Real time clock (RTC)

12.1 RTC prescale

Description

The RTC asynchronous and synchronous prescale coefficients cannot be set to 0. Otherwise, RTC prescale fails.

Workaround

Avoid setting the DIVA[6:0](asynchronous pre-band) and DIVS[14:0](synchronous pre-band) registers of RTC_PRE to 0.

12.2 RTC subpicosecond interrupt

Description

The first RTC subpicosecond interrupt did not respond.

Workaround

The application waits for the second and subsequent subpicosecond interrupt.

12.3 RTC interrupt

Description

The interval between two RTC initialization is less than 1S, the RTC alarm and auto wake-up interrupt can’t be generated.

Workaround

The time interval for the application to initialize RTC is more than 1S.

12.4 RTC Auto Wake-up

Description

RTC Auto wake-up cannot be used as wake-up source of Standby mode.

Workaround

Select RTC ALARM clock as wake-up source.

12.5 RTC mistakenly triggers TISOVF flag bit

Description

When the system wakes up from STANDBY mode,or when the IWDG timeout generates a system reset,the RTC will probabilistically trigger the TISOVF flag bit by mistake.

Workaround

Before entering STANDBY mode or IWDG timeout,When the SHOPF flag is 0,configure the RTC_SCTRL.SUBF[14:0] register,and the flag will be set to 1.When the SHOPF flag is 0 again,configure the RTC_SCTRL.SUBF[14:0] register for the second time;note that the NRST cannot be triggered when the software executes the above process clock.

13 Low Power Timer (LPTIM)

13.1 LPTIM maximum count value

Description

The LPTIM clock source selects the internal clock source (CKSLE bit in the LPTIM_CFG register is 0), and the counter is configured to increment for each valid clock pulse on Input1 (the CNTMEN bit in the LPTIM_CFG register is 1), and the maximum count value of the counter is ARRVAL (automatic reload counter) -1.

Workaround

When the CKSLE bit of the LPTIM_CFG register is 0, and the CNTMEN bit of the LPTIM_CFG register is 1, the calculated ARRVAL target value needs to be added by 1 to configure and use.

14 Controller Area Network (CAN)

14.1 CAN active error

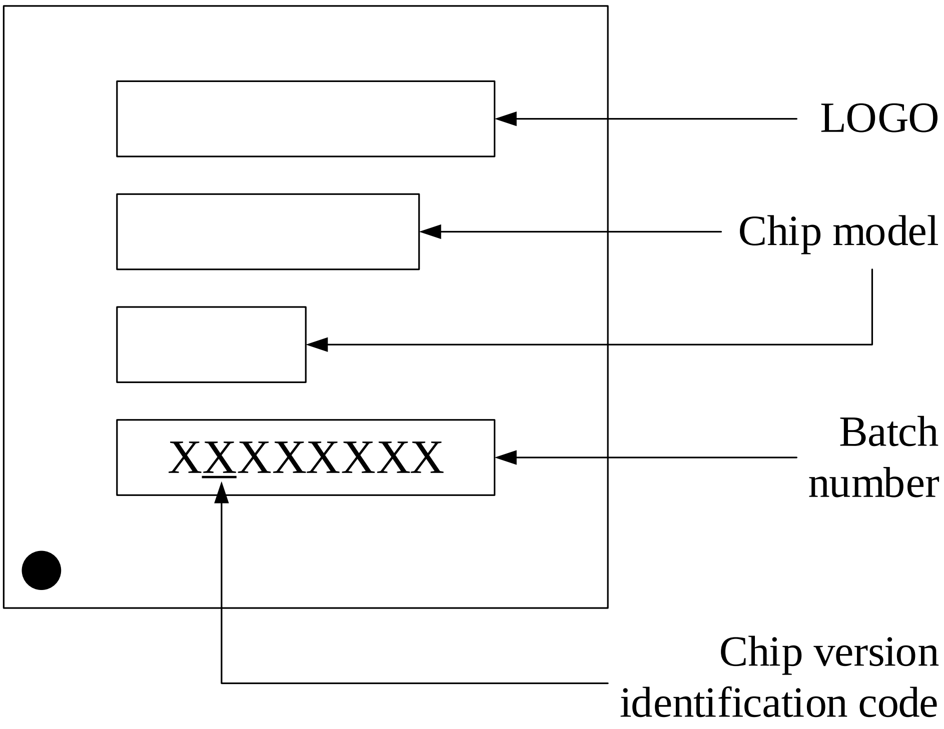
Description

When the CAN is in normal mode and the CAN bit is hard synchronized,if the baud rate deviation of other nodes is too large(approaching or exceeding the synchronization segment),the CAN module is prone to generate the active error.

Workaround

None

15 Chip screen printing and version description



16 Version history

Date	Version	Modify
2021.09.15	V1.0.0	The initial release
2021.11.30	V1.0.1	Add 5.3 ADC analog watchdog Add 5.4 DMA moving ADC data Add 7.4 Abnormal signal interference Add 11.2 RTC subsecond interrupt
2022.02.22	V1.1.0	Add SPI CLK GPIO Configuration
2022.04.06	V1.2	Modify Table 1-1 Errata Description Added version E Modify 11.2 chapter to change RTC sub-pico-second to RTC subsecond Add 12 chapter on Low Power Timer (LPTIM) Add 3.2 LSE-CSS Fault Detection Modify 7.3 chapter to add I2C DMA usage restrictions
2022.09.05	V1.2.1	Add 14 chapters Controller Area Network(CAN) Add 12.3 RTC interrupt chapter Add 12.4 RTC Auto Wake-up chapter
2023.02.14	V1.3	Add 12.5 chapter RTC mistakenly triggers TISOVF flag bit

17 Notice

This document is the exclusive property of Nations Technologies Inc. (Hereinafter referred to as NATIONS). This document, and the product of NATIONS described herein (Hereinafter referred to as the Product) are owned by NATIONS under the laws and treaties of the People’s Republic of China and other applicable jurisdictions worldwide.

NATIONS does not grant 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.

NATIONS reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NATIONS 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 support or 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.