

# N32G432 Series Errata Sheet

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>ERRATA LIST .....</b>  | <b>4</b>  |
| <b>2</b> | <b>POWER CONTROL (PWR).....</b>                                 | <b>6</b>  |
| 2.1      | SYSTEM CLOCK SWITCHING.....                                     | 6         |
| 2.2      | SWITCH TO STOP2 MODE FROM LPRUN MODE .....                      | 6         |
| 2.3      | POWER ON AGAIN AFTER POWER DOWN.....                            | 6         |
| <b>3</b> | <b>RESET AND CLOCK CONTROL (RCC) .....</b>                      | <b>7</b>  |
| 3.1      | ENTER STOP2 MODE FROM RUN MODE .....                            | 7         |
| 3.2      | LSE-CSS FAULT DETECTION.....                                    | 7         |
| <b>4</b> | <b>GPIO AND AFIO.....</b>                                       | <b>8</b>  |
| 4.1      | GPIO ANALOG FUNCTION .....                                      | 8         |
| <b>5</b> | <b>ANALOG/DIGITAL CONVERSION (ADC) .....</b>                    | <b>8</b>  |
| 5.1      | ADC DATA LEFT-ALIGN.....  | 8         |
| 5.2      | ADC INJECTION CHANNELS TRIGGER REGULAR CHANNEL CONVERSIONS..... | 8         |
| 5.3      | ADC ANALOG WATCHDOG.....  | 9         |
| 5.4      | DMA MOVES ADC DATA .....  | 9         |
| <b>6</b> | <b>SERIAL PERIPHERAL INTERFACE (SPI) .....</b>                  | <b>9</b>  |
| 6.1      | SPI INTERFACE.....  | 9         |
| 6.1.1    | SPI Baud Rate Setting .....                                     | 9         |
| 6.1.2    | CRC Check In Slave Mode.....                                    | 10        |
| 6.1.3    | SPI CLK GPIO Configuration.....                                 | 10        |
| 6.2      | I2S INTERFACE .....   | 11        |
| 6.2.1    | PCM Long Frame Mode.....  | 11        |
| <b>7</b> | <b>I2C INTERFACE.....</b>                                       | <b>11</b> |
| 7.1      | HANDLING SOFTWARE EVENTS BEFORE CURRENT BYTE TRANSFERRING.....  | 11        |
| 7.2      | NOTES ON SINGLE READ OF ONE OR TWO BYTES .....                  | 12        |
| 7.3      | USE DMA SIMULTANEOUSLY WITH OTHER PERIPHERALS.....              | 12        |
| 7.4      | ABNORMAL SIGNAL INTERFERENCE.....                               | 13        |
| <b>8</b> | <b>UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER (USART).....</b> | <b>13</b> |
| 8.1      | CHECK ERROR FLAG .....  | 13        |

|           |  |           |
|-----------|--|-----------|
| 8.2       | RTS HARDWARE FLOW CONTROL.....                           | 14        |
| <b>9</b>  | <b>DEBUG INTERFACE (DBG).....</b>                        | <b>15</b> |
| 9.1       | DEBUG REGISTER.....                                      | 15        |
| <b>10</b> | <b>TIMER (TIM).....</b>                                  | <b>16</b> |
| 10.1      | TIMER REPEAT CAPTURE DETECTION.....                      | 16        |
| <b>11</b> | <b>REAL TIME CLOCK (RTC) .....</b>                       | <b>17</b> |
| 11.1      | RTC PRESCALE .....                                       | 17        |
| 11.2      | RTC SUBPICOSECOND INTERRUPT.....                         | 17        |
| 11.3      | RTC INTERRUPT.....                                       | 17        |
| 11.4      | RTC PERIODIC WAKE-UP.....                                | 17        |
| 11.5      | RTC MISTAKENLY TRIGGERS TISOVF FLAG BIT .....            | 18        |
| <b>12</b> | <b>LOW POWER TIMER (LPTIM) .....</b>                     | <b>19</b> |
| 12.1      | LPTIM MAXIMUM COUNT VALUE .....                          | 19        |
| <b>13</b> | <b>CONTROLLER AREA NETWORK (CAN) .....</b>               | <b>20</b> |
| 13.1      | CAN ACTIVE ERROR.....                                    | 20        |
| <b>14</b> | <b>CHIP SCREEN PRINTING AND VERSION DESCRIPTION.....</b> | <b>21</b> |
| <b>15</b> | <b>VERSION HISTORY .....</b>                             | <b>22</b> |
| <b>16</b> | <b>DISCLAIMER.....</b>                                   | <b>23</b> |

# 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 |
| Ch 2: Power Control (PWR)               | 2.1: System clock switchover                                    | •                                 |           |           |           |           |           |
|   | 2.2: Switch to Stop2 mode from LPRUN mode                       | •                                 | •         | •         | •         | •         | •         |
|   | 2.3: Power on again after power down                            | •                                 | •         |           |           |           |           |
| Ch 3: Reset and clock control (RCC)     | 3.1: Enter STOP2 mode from Run mode                             | •                                 | •         | •         | •         | •         | •         |
|   | 3.2 LSE-CSS fault detection                                     | •                                 | •         | •         | •         | •         | •         |
| Ch 4: GPIO and AFIO                     | 4.1: GPIO analog function                                       | •                                 | •         |           |           |           |           |
| Ch 5: Analog/Digital conversion (ADC)   | 5.1: ADC data left-align  | •                                 | •         | •         | •         | •         | •         |
|   | 5.2: ADC injection channels trigger regular channel conversions | •                                 | •         | •         | •         | •         | •         |
|   | 5.3: ADC analog watchdog  | •                                 | •         | •         | •         | •         | •         |
|   | 5.4: DMA moves ADC data   | •                                 | •         | •         | •         | •         | •         |
| Ch 6: Serial peripheral interface (SPI) | 6.1: SPI interface  | 6.1.1: SPI Baud Rate Setting      | •         | •         | •         | •         | •         |
|   |   | 6.1.2: CRC check slave mode       | •         | •         | •         | •         | •         |
|   |   | 6.1.2: SPI CLK GPIO configuration | •         | •         | •         | •         | •         |
|   | 6.2: I2S interface  | 6.2.1: PCM long frame mode        | •         | •         | •         | •         | •         |

|   |  |   |   |   |   |   |   |
|---|--|---|---|---|---|---|---|
| Ch 7: I2C interface                                       | 7.1: Handling Software events before current byte transferring | • | • | • | • | • | • |
|   | 7.2: Notes on single read of one or two bytes                  | • | • | • | • | • | • |
|   | 7.3: Use DMA simultaneously with other peripherals             | • | • | • | • | • | • |
|   | 7.4: Abnormal signal interference                              | • | • | • | • | • | • |
| Ch 8: Universal synchronous asynchronous receiver (USART) | 8.1: Check error flag  | • | • | • | • | • | • |
|   | 8.2: RTS hardware flow control                                 | • | • | • | • | • | • |
| Ch 9: Debug interface (DBG)                               | 9.1: Debug register  | • | • | • | • | • | • |
| Ch 10: Timer (TIM)  | 10.1: Timer repeat capture detection                           | • | • | • | • | • | • |
| Ch 11: Real time clock (RTC)                              | 11.1: RTC prescale   | • | • | • | • | • | • |
|   | 11.2: RTC subpicosecond interrupt                              | • | • | • | • | • | • |
|   | 11.3: RTC interrupt  | • | • | • | • | • | • |
|   | 11.4: RTC Auto Wake-up   | • | • | • | • | • | • |
|   | 11.5: RTC mistakenly triggers TISOVF flag bit                  | • | • | • | • | • | • |
| Ch 12: Low Power Timer (LPTIM)                            | 12.1: LPTIM maximum count value                                | • | • | • | • | • | • |
| Ch 13: Controller Area Network (CAN)                      | 13.1: CAN active error   | • | • | • | • | • |   |

## 2 Power Control (PWR)

### 2.1 System Clock Switching

#### Description

During the process of switching from the system clock using HSI/HSE to MSI, if a system reset occurs, it may lead to chip crash

#### Workaround

To avoid using HSI/HSE directly as the system clock, it is recommended to choose PLL as the system clock.

### 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 always enabled 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, it is necessary to ensure that the chip's VDD voltage drops below 100mV before powering up the MCU.

### **3 Reset And Clock Control (RCC)**

#### **3.1 Enter STOP2 Mode From Run Mode**

##### **Description**

Dividing AHB CLK and APB CLK by certain frequency divisors may result in an extremely low probability of the chip not entering Stop2 mode in Run mode. The higher the frequency divisor, the more likely this phenomenon may occur.

##### **Workaround**

Back up AHB CLK frequency division factor and configure the AHB CLK to be undivided before entering STOP2 mode. Besides, restore the frequency division backed up after exit from STOP2 mode.

#### **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 on again.

## 4 GPIO And AFIO

### 4.1 GPIO Analog Function

#### Description

When the four GPIO PA1/PA2/PA3/PA4 output high and are switched to analog function, it will cause 30mV voltage drop in the switching process.

#### Workaround

Avoid the above methods.

## 5 Analog/Digital Conversion (ADC)

### 5.1 ADC Data Left-Align

#### Description

In ADC single conversion mode, with non-12-bit precision and left alignment, when a software-triggered conversion of the regular channel is initiated, the highest bit of the invalid bits in the ADC\_DAT register is set to 1.

#### Workaround

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

### 5.2 ADC Injection Channels Trigger Regular Channel Conversions

#### Description

In ADC continuous conversion mode, with external triggering disabled for regular channels and only software-triggered injection channel conversion, the regular channels may be triggered for conversion, resulting in data being generated in ADC\_DAT and the corresponding status bit of the regular channel conversion in ADC\_STS being set.

#### Workaround

Ignore the flags and data generated by the regular channels.



## 5.3 ADC Analog Watchdog

### Description

In ADC independent mode with single conversion and non-12-bit precision, enabled analog watchdog may be triggered by mistake if the high threshold value of the analog watchdog is equal to the value in the ADC data register with all invalid bits equal 0 when triggering the conversion of regular/injected channels by software.

### 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

In ADC independent mode with continuous conversion and DMA transfer of ADC data, when the configured number of transfers is completed, disable DMA first, then disable ADC, and then re-enable DMA. If DMA is re-enabled before ADC is enabled, the first data transferred by DMA will be the remaining ADC conversion data from before DMA was disabled.

### Workaround

In this case, if DMA is re-enabled after being disabled in a loop, an additional data transfer is added to the base transfer count N. After the first disable, the data from the first to the Nth transfer is read. Subsequent disables will read the data from the second to the N+1th transfer.

## 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  $f_{PLCK}/2$  in SPI master mode, the CRC check will fail.

### Workaround

In this case, avoid setting baud rate control bit (BR[2:0]) to  $f_{PLCK}/2$ .

### 6.1.2 CRC Check in Slave Mode

#### Description

When the SPI is operating in slave mode with CRC enabled, even if the NSS pin is at a high level, as long as the SPI receives clock signals, it will still perform CRC calculation.

#### 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 as high, after enabling the SPI, the clock of GPIO will be pulled high before outputting the clock signal. The slave may mistakenly interpret this edge as a clock signal, causing the received data to shift.

#### Workaround

Configure the clock of GPIO based on the clock polarity (pull up the clock of GPIO if the clock polarity is high, and pull down the clock of GPIO if the clock polarity is low) before enabling the SPI.

## 6.2 I2S Interface

### 6.2.1 PCM Long Frame Mode

#### Description

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

1. When transferring more than one byte using I2C, it is recommended to use DMA
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 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

1. Single byte read:
  - a) Upon receipt of ADDR<sub>F</sub>
  - b) Set the ACKEN bit to 0
  - c) Clear the ADDR<sub>F</sub> 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 ADDR<sub>F</sub>
  - b) Set the ACKPOS bit to 1
  - c) Clear the ADDR<sub>F</sub> 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 Simultaneously with Other Peripherals

### Description

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

### Workaround

During I2C communication using DMA, disable DMA for other peripherals.

## 7.4 Abnormal Signal Interference

### Description

During operation, I2C may experience communication anomalies due to glitches on SCL and SDA lines.

### 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 reception of a byte of data, if a checksum error is detected before the stop bit is received, the checksum error flag is set. During this time, the checksum error flag cannot be cleared by software means (reading the status register and then reading the data register). If the checksum error interrupt is enabled, the checksum error interrupt handler may be triggered multiple 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 event occurs, while reading TIMx\_CCDA Tx (capture/compare register x) (the read operation automatically clears the capture flag), the CCxOCF (capture/compare x overcapture flag) may still be set.

#### Workaround

None



## 11 Real Time Clock (RTC)

### 11.1 RTC Prescale

#### Description

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

#### Workaround

Avoid setting the DIVA[6:0](asynchronous prescale) and DIVS[14:0](synchronous prescale) registers of RTC\_PRE to 0.

### 11.2 RTC Subpicosecond Interrupt

#### Description

The first RTC subpicosecond interrupt will not be generated.

#### Workaround

The application waits for the second and subsequent subpicosecond interrupt.

### 11.3 RTC Interrupt

#### Description

If 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 RTC initialization should be more than 1S.

### 11.4 RTC Periodic Wake-up

#### 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.

## 11.5 RTC Mistakenly Triggers TISOVF Flag Bit

### Description

When the system wakes up from STANDBY mode or reset by IWDG timeout, TISOVF flag in RTC may be set incorrectly.

### Workaround

Before entering STANDBY mode or when the system is reset by IWDG timeout, when the SHOPF flag is 0, configure RTC\_SCTRL.SUBF[14:0] register once, and the SHOPF flag will set be to 1. When the SHOPF flag is 0 again, it should configure RTC\_SCTRL.SUBF[14:0] register for the second time. This will solve the issue.

*Note: that NRST cannot be triggered during above process.*

## 12 Low Power Timer (LPTIM)

### 12.1 LPTIM Maximum Count Value

#### Description

When selecting the internal clock source for LPTIM (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), the maximum count value of the counter is ARRVAL (automatic reload counter) -1.

#### Workaround

When the LPTIM\_CFG register CKSLE bit is 0 and the LPTIM\_CFG register CNTMEN bit is 1, the calculated target value of ARRVAL needs to be incremented by 1 for configuration.

## 13 Controller Area Network (CAN)

### 13.1 CAN Active Error

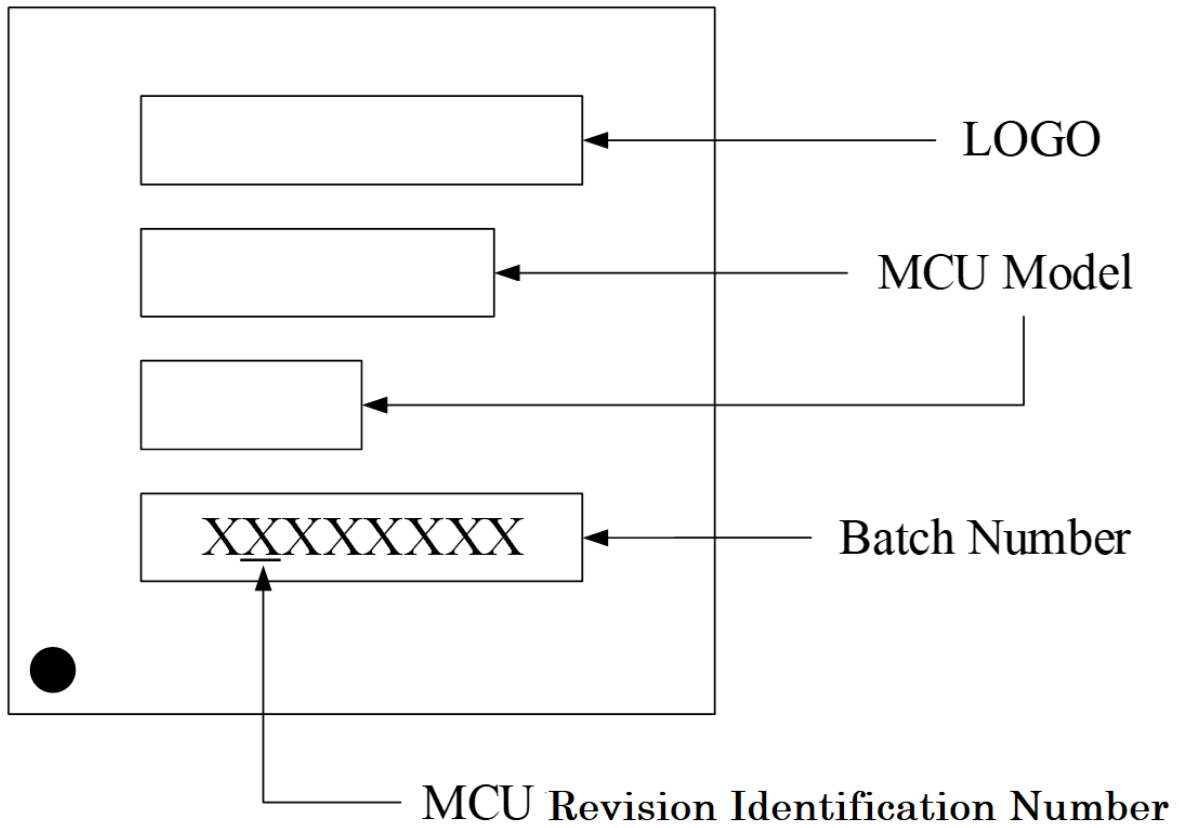
#### Description

In normal mode, if the CAN bit is hard synchronized and the bus rate deviation of other nodes on the bus is too large (approaching or exceeding the synchronization segment), the CAN module is prone to report active errors.

#### Workaround

None

### 14 Chip Screen Printing and Version Description



## 15 Version History

| Version | Date       | Changes  |
|---------|------------|--|
| V1.0.0  | 2021.09.15 | Initial release  |
| V1.0.1  | 2021.11.30 | <ol style="list-style-type: none"> <li>1. Added 5.3 ADC analog watchdog</li> <li>2. Added 5.4 DMA moving ADC data</li> <li>3. Added 7.4 Abnormal signal interference</li> <li>4. Added 11.2 RTC subsecond interrupt</li> </ol>   |
| V1.1.0  | 2022.02.22 | Add SPI CLK GPIO Configuration   |
| V1.2    | 2022.04.06 | <ol style="list-style-type: none"> <li>1. Modified Table 1-1 Errata Description Added version E</li> <li>2. Modified 11.2 chapter to change RTC sub-pico-second to RTC subsecond</li> <li>3. Added 12 chapter on Low Power Timer (LPTIM)</li> <li>4. Added 3.2 LSE-CSS Fault Detection</li> <li>5. Modified 7.3 chapter to add I2C DMA usage restrictions</li> </ol> |
| V1.2.1  | 2022.09.05 | <ol style="list-style-type: none"> <li>1. Added 13 Ch Controller Area Network(CAN)</li> <li>2. Added 11.3 RTC interrupt Ch</li> <li>3. Added 11.4 RTC Auto Wake-up Ch</li> </ol>   |
| V1.3    | 2023.02.14 | Add 11.5 Ch RTC mistakenly triggers TISOVF flag bit  |
|         |            |  |

## 16 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.