ES0005
Errata Sheet

AT32F421 device limitations

Device identification

This errata sheet applies to ARTERY AT32F421 microcontrollers that feature an ARM™ 32-bit Cortex®-M4 core.

Table 1. Device summary

Device	Flash memory	Part number
AT32F421	64 KB	AT32F421C8T7 AT32F421K8T7 AT32F421K8U7 AT32F421K8U7-4 AT32F421F8P7 AT32F421G8U7 AT32F421PF8P7 AT32F421PF4P7 AT32F4212C8T7
	32 KB	AT32F421C6T7 AT32F421K6T7 AT32F421K6U7 AT32F421K6U7-4 AT32F421F6P7 AT32F421G6U7
	16 KB	AT32F421C4T7 AT32F421K4T7 AT32F421K4U7 AT32F421K4U7-4 AT32F421F4P7 AT32F421G4U7

Contents

List of tables

1 AT32F421 device limitations

Table 2 gives a list of limitations that have been identified so far on the AT32F421 devices.

Table 2. Summary of device limitations

Sections	Description
1.1 ERTC	1.1.1 How to update TIME and DATE register value. 1.1.2 Writing ERTC occupies APB for 4 ERTC clock cycles
1.2 I2C	1.2.1 1st clk from host becomes shorter when SCL stretching is released by slave in I2C master mode. 1.2.2 START failed to send when the slave pulls down bus during RESTART transfer in I2C master mode 1.2.3 I2C slave communication failed when APB equals or less than 4MHz
1.3 I2S	1.3.1 Unable to resume communication while I2S CK line is disturbed. 1.3.2 I2S Philips protocol SOF data error in certain conditions. 1.3.3 First data error in I2S PCM standard long frame receive-only mode. 1.3.4 UDR flag is set in I2S slave transmission mode and discontinuous communication state. 1.3.5 Data reception error when I2S 24-bit data is packed into 32-bit format.
1.4 PWC	1.4.1 PVM event generation after PVM enabled when VDD is above PVM threshold. 1.4.2 Unable to wakeup Deepsleep mode after AHB frequency division. 1.4.3 Systick interrupt wakes up Deepsleep mode . 1.4.4 SWEF flag is set when enabling a standby-mode wakeup pin. 1.4.5 Disabling Flash memory clock causes additional power consumption in Deepsleep mode. 1.4.6 Unable to select system clock source after waking up Deepsleep mode.
1.5 TMR	1.5.1 How to clear TMR-triggered DAM requests. 1.5.2 TMR overrun in encoder mode counter. 1.5.3 TMR1 accessing 0x4C address causes DMA request 1.5.4 Break input failed when TMREN=0 (TMR disabled)
1.6 CRM	1.6.1 PLL 2x or 3x multiplication factor failure.
1.7 SPI	1.7.1 CS failing edge not synchronized in slave SPI hardware CS mode.

1.1 ERTC

1.1.1 How to update TIME and DATE register values

- Description:

  If no operation were performed on the ERTC registers, the ERTC_TIME and ERTC_DATE registers will not be updated, which means they hold the values updated last time when they were accessed.

- Workaround:

  Read status register first before reading TIME and DATE registers.

1.1.2 Writing ERTC occupies APB for 4 ERTC clock cycles

- Description:

  Writing ERTC register takes approximately four ERTC CLK clock cycles to be synchronized with the battery powered domain, causing APB1 to be occupied and DMA transfer on APB1 to be halted during this period until the completion of the operation process.

- Workaround:

  After ERTC initialization, if ERTC features can satisfy users' needs, try to reduce the times of writing ERTC registers so as to reduce its impact on system.

1.2 I2C

1.2.1 1st clk from host becomes shorter when SCL stretching is released by slave in I2C master mode

- Description:

  In I2C master mode, when the slave releases the SCL bus at the place where the slave SCL stretching feature is enabled, the first clk sent by the master may be shorter so that the slave cannot receive data normally.

- Workaround:

  Add a delay to the location where SCL stretching feature is enabled to avoid sending data immediately after SCL stretching is released.

1.2.2 START failed to send when the slave pulls down bus during RESTART transfer in I2C master mode

- Description:

  In I2C master mode, if the slave pulls down the bus while RESTART signal is being sent, then the START signal may fail to be sent.

- Workaround:

  Add a delay to the place where the slave enables SCL stretching in order to avoid sending data immediately after SCL stretching is released.

1.2.3 I2C slave communication failed when APB equals or less than 4MHz

- Description:

  I2C is unable to communicate at 400kHz in slave mode when the APB clock is equal to or less than 4MHz.

- Workaround:

  Increase the APB clock to 8 MHz, or reduce the I2C speed to 100kHz.

1.2.4 BUSERR is detected by I2C before start of communication

- Description:

  When all the following conditions are present, BUSERR conditions would be detected by I2C, causing communication error.

  The three conditions are as follows:

  Condition 1: I2C is enabled

  Condition 2: Before the start of communication

  Condition 3: BUSERR timing takes place on the bus

- Workaround:

  Check if the BUSERR flag is set or not before the start of communication. If it is set, just need clear this flag to enable communication. Optionally, enable error interrupt, and clear it in the interrupt after the BUSERR flag is set.

1.3 I2S

1.3.1 Unable to resume communication while I2S CK line is disturbed

- Description:

  The CK and WS signals of the I2S are not synchronized inside the device, so that when the clock line is interfered during actual communication, such interference will be treated as a CK signal by the I2S, causing communication unable to resume automatically.

- Workaround:

  Pull-up/pull-down the WS and CK pins internally or externally (depending on the desired audio protocol and I2SCLKPOL configuration). When communication error is detected, disabling and then enabling I2S can help resume communication.

1.3.2 I2S Philips protocol SOF data error in certain conditions

- Description:

  In case of I2S Philips protocol, master receive and slave transmission mode, and I2SCLKPOL high level, the WS signal falling edge corresponding to the left channel of the first data frame would not be output effectively, causing some devices unable to receive data from the left channel.

- Workaround:

  Pull up or pull down the WS and SCK pins internally or externally, depending on the desired audio protocols and I2SCLKPOL configuration.

1.3.3 First data error in I2S PCM standard long frame receive-only mode

- Description:

  When PCLK frequency division factor is greater than 1, and I2S PCM standard long frame receive-only mode is enabled, if I2SCPOL = 0 is set and the SCK line remains high before enabling I2S, the first received data would be incorrect.

- Workaround:

  Pull up or pull down the SCK pin externally or internally, depending on the I2SCLKPOL configuration.

1.3.4 UDR flag is set in I2S slave transmission mode and discontinuous communication state

- Description:

  The UDR flag is set exceptionally in I2S slave transmission mode alongside discontinuous communication state, even if data have been written before the start of communication.

- Workaround:

  For continuous communication, it is recommended to use DMA or interrupts for fast data transfer in I2S slave transmission mode according to the protocols.

1.3.5 Data reception error when I2S 24-bit data is packed into 32-bit format

- Description:

  When I2S 24-bit data is packed into 32-bit frame format, the remaining 8 invalid CLK data would be received by the receiver as normal data.

- Workaround:

  Method 1: Both the receiver and transmitter use the same way of packing 24-bit data into 32-bit format.
  Method 2: Discard these 8 invalid CLK data in this frame format using software.

1.4 PWC

1.4.1 PVM event generation after PVM enabled when VDD is above PVM threshold

- Description:

  When the VDD is greater than PVM threshold, an unwanted PVM event is generated as soon as PWC voltage monitoring is enabled.

- Workaround:

  Clear the unwanted PVM event during PVM initialization.

1.4.2 Unable to wakeup Deepsleep mode after AHB frequency division

- Description:

  If AHB frequency is divided, no wakeup source can wake up Deepsleep mode.

- Workaround:

  Do not divide AHB frequency in Deepsleep mode.

  Remove AHB frequency division before entering Deepsleep mode. Configure then the desired AHB frequency after wakeup.

1.4.3 Systick interrupt wakes up Deepsleep mode

- Description:

  If Systick or Systick interrupt is not disabled before the Deepsleep mode is entered, the Systick then would keep running after Deepsleep mode entry, and the subsequent Systick interrupt would wake up Deepsleep mode.

- Workaround:

  Disable Systick or Systick interrupts before entering Deepsleep mode.

1.4.4 SWEF flag is set when enabling a standby-mode wakeup pin

- Description:

  If a wakeup pin (Standby-mode wakeup pin) were used as a GPIO push-pull output (high) or pull-up input Before being enabled, a SWEF flag would be set immediately once the pin is enabled.

- Workaround:

  If the Standby-mode wakeup pin was used as a GPIO before, then the IO has to be re-initialized to pull-down input or analog input mode before enabling the pin. For example:

```
gpio_init_type gpio_init_struct;

/* enable the button clock */
crm_periph_clock_enable(CRM_GPIOA_PERIPH_CLOCK, TRUE);

/* set default parameter */
gpio_default_para_init(&gpio_init_struct);

/* configure wakeup pin as input with pull-down */
gpio_init_struct.gpio_drive_strength = GPIO_DRIVE_STRENGTH_STRONGER;
gpio_init_struct.gpio_out_type  = GPIO_OUTPUT_PUSH_PULL;
gpio_init_struct.gpio_mode = GPIO_MODE_INPUT;
gpio_init_struct.gpio_pins = USER_BUTTON_PIN;
gpio_init_struct.gpio_pull = GPIO_PULL_DOWN;
gpio_init(GPIOA, &gpio_init_struct);

/* enable wakeup pin - pa0 */
pwc_wakeup_pin_enable(PWC_WAKEUP_PIN_1, TRUE);
```

1.4.5 Disabling Flash memory clock causes additional power consumption in Deepsleep mode

- Description:

  If Flash memory clock is disabled (bit 4 of the CRM_AHBEN register) before Deepsleep mode entry, then the low-power mode of Flash memory would become invalid in Deepsleep mode, causing additional power consumption of around 50uA.

- Workaround:

  Make sure that the Flash memory clock (bit 4 of the CRM_AHBEN register) is always enabled before entering Deepsleep mode.

1.4.6 Unable to select system clock source after waking up Deepsleep mode

- Description:

  When a wakeup source arrives at the very moment while the Deepsleep mode is being entered, either HEXT or PLL could no longer be selected as the clock source of system clock.

- Workaround:

  After waking up Deepsleep mode, wait around 3 LICK clock cycles before starting system clock configuration.

1.5 TMR

1.5.1 How to clear TMR-triggered DAM requests

- Description:

  TMR-induced DMA request cannot be cleared by resetting/setting the corresponding DMA request enable bit in the TMRx_IDEN register.

- Workaround:

  Before enabling DMA channel transfer, reset TMR (reset CRM clock of TMR) and initialize TMR to clear pending DMA requests.

1.5.2 TMR overrun in encoder mode counter

- Description:

  In encoder counting mode, if the counter counts back and forth between 0 and PR, the OVFIF is not set at an overrun or underrun event.

- Method 1:

  Configure the C3IF and C4IF channels of the TMR (where an encoder is being used) as output mode, C3DT = AR, C4DT = 0, and enable C3IF and C4IF interrupts.
  C3IF event & downcounting indicates an underrun;
  C4IF event & upcounting indicates an overrun;
  This method has its limitation: If the input frequency of the encoder mode counter were too fast, interrupts would occur frequently and need to be handled by software, causing not enough time to deal with interrupts. Thus this method applies to the scenario where the external input frequency of the encoder is not so fast.

- Method 2:

  Turn to a TMR with enhanced mode (the counter can be extended from 16-bit to 32-bit width) in order to expand the encoder's counting range that detects forward and reverse rotation, and configure the initial value of the counter to PR/2 so as to prevent the timer from overflowing.
  This method has its limitation: The forward and reverse rotation of the encoder must be limited to a certain range. An overflow still occurs if the encoder were always rotated in one direction. This method applies to the scenario where the rotation of the encoder is controlled at a certain range.

1.5.3 TMR1 accessing 0x4C address causes DMA request exception

- Description:

  When TMR1 issues a DMA request, the lower 8 bits of the current address bus is 0x4C, and DMA transfer doesn't change the APB bus address where the TMR is located. In this case, a single TMR DMA request would be set again after being cancelled, causing potential DMA transfer error.

- Workaround:

  User other timers than TMR1.

1.5.4 Break input failed when TMREN=0 (TMR disabled)

- Description:

  When TMREN=0 (Timer is not enabled), break input failed to work, causing it unable to trigger break event or interrupt.

  Example: in single-pulse mode, TMREN is cleared automatically at the end of one-cycle counting. But due to above-mentioned reason relating to break input, output enable bit (OEN) cannot be cleared, nor can a break flag be set.

- Workaround:

  None.

1.6 CRM

1.6.1 PLL 2x or 3x multiplication factor failure

- Description:

  PLL output clock should be greater than or equal to 16 MHz due to PLL output range limitations. A lower PLL input clock frequency may cause an error when 2x or 3x multiplication factor is used.

- Workaround:

  Try not to use 2x or 3x multiplication factor of the PLL.

1.7 SPI

1.7.1 CS failing edge not synchronized in slave SPI hardware CS mode

- Description:

  In SPI slave hardware CS mode, the initial CLK synchronization for data transfer is not performed at each CS falling edge.

- Workaround:

  Solution A: Strictly control the slave CS line, pull high the CS line as soon as the communication is complete.

  Solution B: Enable CRC check. Once a CRC error is detected, reset SPI and restart handshake communication.

2 Document revision history

Table 3. Document revision history

Date	Revision	Changes
2021.9.30	2.0.0	Initial release
2022.03.01	2.0.1	1. Added *PLL 2x or 3x multiplication factor failure.*
2022.04.15	2.0.2	1. Updated I2S content structure. 2. Added *First data error in I2S PCM standard long frame receive-only mode*. 3. Added *UDR flag is set in I2S slave transmission mode and discontinuous communication state*. 4. Added *Data reception error when I2S 24-bit data is packed into 32-bit format*. 5. Added *START failed to send when the slave pulls down bus during RESTART transfer in I2C master mode*. 6. Added *CS failing edge not synchronized in slave SPI hardware CS mode*.
2022.04.28	2.0.3	1. Added an example case in the section *1.4.4 SWEF flag is set when enabling a standby-mode wakeup pin* 2. Added *1.2.3 I2C slave communication failed when APB equals or less than 4MHz*
2022.09.06	2.0.4	Added *1.2.4 BUSERR is detected by I2C before start of communication*
2022.09.21	2.0.5	Added *1.1.2 Writing ERTC occupies APB for 4 ERTC clock cycles*
2022.11.2	2.0.6	Added the part number *AT32F4212C8T7 in Table 1 64KB.*
2023.02.08	2.0.7	Updated the description in the section *1.1.2 Writing ERTC occupies APB for 4 ERTC clock cycles*
2023.06.15	2.0.8	Added *section 1.5.4 Break input failed when TMREN=0 (TMR disabled)*

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license granted by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement on any patent, copyright or other intellectual property right.

Purchasers hereby agree that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any aircraft application; (C) any aerospace application or environment; (D) any weapon application, and/or (E) or other uses where the failure of the device or product could result in personal injury, death, property damage. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and Purchasers are solely responsible for meeting all legal and regulatory requirements in such use.

Resale of ARTERY products with provisions different from the statements and/or technical characteristics stated in this document shall immediately void any warranty grant by ARTERY for ARTERY's products or services described herein and shall not create or expand any liability of ARTERY in any manner whatsoever.