XHSC 小华半导体
XIAOHUA SEMICONDUCTOR

32-bit Microcontroller

HC32L07x_F072 Series

Errata

Rev. 1.03 September 2024

Target customers

Product range	Product Model	Product range	Product Model
L Series	HC32L072 HC32L073	**F Series**	HC32F072

Disclaimer

★ Xiaohua Semiconductor Co., Ltd. (hereinafter referred to as "XHSC") reserves the right to change, correct, enhance, and modify Xiaohua Semiconductor products and/or this document at any time without prior notice. Users can obtain the latest relevant information before placing an order. XHSC products are sold in accordance with the sales terms and conditions stated in the basic purchase and sales contract.

★ It is the customer's sole responsibility to select the appropriate XHSC product for your application and to design, validate and test your application to ensure that your application meets the applicable standards and any safety, security or other requirements.

★ XHSC hereby acknowledges that no license to any intellectual property rights is granted, either expressly or impliedly.

★ Resale of XHSC products on terms different from those set forth herein will void any warranty provided by XHSC with respect to such products.

★ Any graphics or words with "®" or "™" logo are trademarks of XHSC. All other product or service names displayed on XHSC products are the property of their respective owners.

★ The information in this notice supersedes and replaces the information in previous versions.

Table of contents

1 Summary

This document mainly introduces the usage limitations and workarounds of HC32L07x/ HC32F072 series chips.

2 Errata for HC32L07x/ HC32F072 series

Table 2-1 Errata Item Summary

Errata type	Specific errata items
2.1 Programming	2.1.1 When powered by 1.8V, the speed of programming in SWD mode needs to be reduced
2.2 ADC	2.2.1 Do not use sequential scanning and queue-jump scanning of ADC at the same time
2.3 I2C	2.3.1 The module needs to be reset before I2C initialization
	2.3.2 I2C pin configuration
	2.3.3 When I2C is the host, it cannot receive NACK in some states during the data transmission phase
2.4 SPI	2.4.1 When the slave NCS is fixed to ground, do not shut down the SPI module after it is enabled
	2.4.2 MISO pin is abnormal when SPI is used as multiple slaves
	2.4.3 When SPI is in slave mode, initialization precautions after NSS is pulled low
2.5 GPIO	2.5.1 In DeepSleep mode, a higher frequency GPIO wakeup will cause a chip reset
	2.5.2 Some IOs have a certain driving capability when powered on
	2.5.3 Initialization pull-up processing of unpackaged GPIO ports
2.6 UART_LPUART	2.6.1 The receiving completion flag RC and the parity error flag PE are set asynchronously
	2.6.2 The receiving completion flag RC and the frame error flag FE are set asynchronously
2.7 USB	2.7 USB 2.7.1 The address range of USB access to SRAM is 0x2000_0000~0x2000_1fff (8K)
2.8 CAN	2.8.1 An undefined frame is sent when the bus is disturbed
	2.8.2 An undefined waveform is sent when the bus is disturbed and occupies the bus
	2.8.3 The sampling point is inaccurate when the communication clock is not divided
2.9 DAC	2.9.1 DAC direct output has deviation at high temperature
	2.9.2 DAC generates spikes through OPA output
2.10 Clock	2.10.1 Selection of external passive clock XTH frequency

2.1 Programming

2.1.1 Under 1.8V power supply, SWD mode programming needs to reduce the speed

- Problem description

Under 1.8V power supply, GPIO speed is slow, and high-speed programming using the SWD mode of the offline programmer may fail.

- Workaround

Under 1.8V power supply, if you need to use the SWD mode of the offline programmer to program, you need to reduce the speed and use the low-speed mode to program successfully.

2.2 ADC

2.2.1 Do not use ADC sequential scanning and queue-jumping scanning at the same time

- Problem description

When the sequential scan and queue-jump scan modes of the ADC are used at the same time, queue-jump scan data may be lost, or the sequential scan and queue-jump scan may not be performed.

- Workaround

Separate the sequential scan and queue-jump scan, and only use the sequential scan or queue-jump scan mode for ADC sampling at the same time.

2.3 I2C

2.3.1 The module needs to be reset before I2C initialization

- Problem description

After power-on or reset, some registers of the I2C module are not reset.

- Workaround

Before initialization, after enabling the corresponding I2C peripheral clock, you need to use the reset register to reset the I2C module. Write 0 and then 1 to the corresponding bit of the reset register.

2.3.2 I2C pin configuration

- Problem description

If you first set the I/O to be multiplexed as an I2C function pin, then enable the I2C peripheral clock, and then use the reset register to reset the I2C module, during the period from enabling the peripheral clock to resetting the I2C module, the I2C pin may output a flip pulse signal.

- Workaround

You need to enable the I2C peripheral clock first, then call the reset register to reset the I2C module, and then multiplex the GPIO as an I2C pin.

2.3.3 When I2C is sending data as a host, it cannot receive NACK in some states

- Problem description

When I2C is used as the host, in the data transmission phase, after sending data in the status code 0x18 and 0x28, the SDA bus is not released, but the SDA bus is pulled down.

- Workaround

According to the specific application, take corresponding measures at the protocol layer or use IO to simulate I2C.

2.4 SPI

2.4.1 When the slave NCS is fixed to ground, do not turn off the SPI module after enabling it

- Problem description

When SPI communicates with a device without an NCS interface, if SCKO and CPOL are out of phase, SCK will have pulse output when SPEN is enabled and disabled, causing communication abnormality.

- Workaround

When the slave NCS is fixed to ground, do not disable the SPI module after enabling it.

2.4.2 MISO pin abnormality when SPI is used as multiple slaves

- Problem description

When SPI is used for multi-slave communication, when the last bit sent is 0 and the host CS is pulled high, MISO will remain at a low level.

- Workaround

When the last byte is sent, the software changes MISO to the input pull-up state, and when the slave encounters CS being pulled low, it will become the MISO pin again.

2.4.3 When SPI is in slave mode, initialization precautions after NSS is pulled low

- Problem description

When the SPI module is used in slave mode:

When CPHA=0, the NSS signal is pulled low before initializing the module, and the slave cannot send or receive data;

When CPHA=1, the NSS signal is pulled low before initializing the module, and the slave can send or receive data.

- Workaround

When the SPI module is used in slave mode:

When CPHA=0, the module must be initialized before the NSS signal is pulled low.

2.5 GPIO

2.5.1 In DeepSleep mode, a high-frequency GPIO wakeup will cause a chip reset

- Problem description

In DeepSleep, when the GPIO inputs a periodic wake-up signal greater than 50KHz, the chip may reset.

- Workaround

In DeepSleep, if the input wake-up signal is a periodic signal, it should not be greater than 50KHz.

2.5.2 Some IOs have certain drive capabilities when powered on

- Problem description

During the period from chip power-on to VCAP voltage stabilization, some IOs with analog output multiplexing functions may have low-voltage pulses with weak output drive capability.

- Workaround

1) When using, avoid using these pins as output functions, or avoid connecting them to devices that are sensitive to low voltage and weak drive (such as transistors, MOS tubes, etc.);

2) If they must be used as output functions, external drive can be used to ensure that IO is fixed at a low level during power-on to reduce the impact on external devices during chip power-on;

3) Increasing the MCU power-on rate can indirectly reduce the duration of the pulses that may appear on the IO and reduce the impact on external devices during chip power-on.

2.5.3 Initialization of pull-up processing for unpackaged GPIO ports

- Problem description

For the small package models of the chip, some pins are not packaged compared to the largest package. When the chip enters DeepSleep mode, if these unpackaged pins are not processed, they will also affect the chip's minimum power consumption index.

- Workaround

Referring to the largest package of the chip, these unpackaged pins can be set as input pull-up or input pull-down.
For those packaged but unused pins, the above processing can also be referred to; for packaged pins that have external circuit connections, it is necessary to avoid the voltage difference with the off-chip voltage signal to cause leakage.

2.6 UART_LPUART

2.6.1 The reception completion flag RC and the parity error flag PE are set asynchronously

- Problem description

When the UART_LPUART with parity error flag PE function is running in MODE2/3 and hardware parity function is enabled, if a parity error occurs, the PE flag and RC flag are set asynchronously, especially when the stop bit is 1.5bit or 2bit.

- Workaround

Interrupt mode: You can only enable the RC receive interrupt, and query whether the parity error flag PE is set in the RC receive interrupt.

Polling mode: it is recommended to first determine whether the receive completion flag RC is set, and then determine whether the parity error flag PE is set.

2.6.2 The reception completion flag RC and the frame error flag FE are set asynchronously

- Problem description

UART_LPUART with Frame Error Flag FE function, when running in MODE1/2/3, if a frame error occurs, the FE flag and RC flag are set asynchronously. Especially for the case where the stop bit is 1.5bit or 2bit.

- Workaround

Interrupt mode: You can only enable the RC receive interrupt, and query whether the frame error flag FE is set in the RC receive interrupt.

Polling mode: it is recommended to first determine whether the receive completion flag RC is set, and then determine whether the frame error flag FE is set.

2.7 USB

2.7.1 The address range of USB access to SRAM is 0x2000_0000~0x2000_1fff(8K)

- Problem description

The RAM space of L07x/F072 series is 16K, but the access space of USB part can only access the first 8K.

- Workaround

Specify the relevant variables on RAM that USB needs to access at the address of the first 8K of RAM.

2.8 CAN

2.8.1 Undefined frame is issued when the bus is disturbed

- Problem description

When the bus is disturbed, the CAN controller may send frames that are not defined by the application, including undefined IDs or undefined data.

- Workaround

The workaround measures include the following:

1) After enabling transmission, do not fill data into any transmission buffer and enable transmission before transmission is completed;

2) Add a message validity confirmation mechanism to the application, such as adding a handshake protocol, adding frame (including ID and data) verification, and judging whether the newly received frame is adopted based on the system status.

2.8.2 When the bus is disturbed, an undefined waveform is emitted to occupy the bus

- Problem description

When the bus is disturbed, the CAN controller may send out a waveform not defined by the CAN protocol to occupy the bus when sending.

- Workaround

It is recommended to fill and send only one frame of data each time, and perform a sending timeout. After normal sending is completed, you can continue to fill and send a new frame directly; If the sending timeout occurs, you need to reinitialize the CAN controller and wait for at least 11 CAN bit times before sending. The sending timeout can be roughly calculated based on the total number of bus nodes and the baud rate.

Take the following conditions as an example:

1) If there are 10 bus nodes;

2) The baud rate is 1Mbps;

3) The data length is 8 bytes, and the maximum time required for sending is 140us;

Theoretically, under normal circumstances, it takes at least about 1.4ms for all 10 nodes to send a frame in sequence, so the timeout can be set to 2ms or longer. However, when the bus is disturbed, the timeout should be longer, such as 5ms.

After the transmission timeout, the initialization process of the CAN controller is as follows:

1) Turn off the peripheral clock of the CAN controller (through the SYSCTRL_PERI_CLKEN1 register);

2) Enable the peripheral clock of the CAN controller (through the SYSCTRL_PERI_CLKEN1 register);

3) Initialize the registers of the CAN controller.

2.8.3 The sampling point is inaccurate when the communication clock is not divided

- Problem description

When the communication clock is not divided (CAN BT.PRESC = 0x00), the sampling point of CAN communication is inaccurate.

- Workaround

When configuring the bit time, the communication clock must be divided (CANBT.PRESC>0x1).

2.9 DAC

2.9.1 DAC direct output has deviation at high temperature

- Problem description

At temperatures of 85°C and above, the direct output of the DAC will have deviations, affecting performance.

- Workaround

Output the DAC through the buffer function of the OPA.

2.9.2 DAC generates spikes through OPA output

- Problem description

When using the DAC to output a sine wave through the OPA, a spike of several to more than ten mV will be generated near the 0 point (corresponding to CODE: 2048).

- Workaround

An external filtering circuit is required.

2.10 Clock

2.10.1 Selection of the frequency of the external passive clock XTH

- Problem description

Some 8MHz passive crystals have a large ESR. When matching such crystals, if the matching is not good, the oscillation may not start or the oscillation may be abnormal.

- Workaround

It is recommended to use a low ESR 16MHz~32MHz crystal for the external passive clock XTH, and the matching needs to be done well.

Version Revision History

Version	Date	Description
Rev1.0	2023/02/10	Initial edition released.
Rev1.01	2023/07/14	1) Modified the description of the section "2.4.2 Some IOs have certain driving capabilities when powered on"; 2) Added the section "ADC".
Rev1.02	2024/04/19	1) In the "SPI" section, the content "When SPI is in slave mode, initialization precautions after NSS is pulled low" was added; 2) In the "CAN" section, the content "The sampling point is inaccurate when the communication clock is not divided" was added.
Rev1.03	2024/09/02	Added 2.10 Clock section about the selection of external passive clock XTH frequency.