XHSC 小华半导体
XIAOHUA SEMICONDUCTOR

32-bit Microcontrollers

HC32F460_A460_F451_A452 Series

corrigenda

Rev1.0 June 2023

Applicable objects

Product Series	Product Model	Product Series	Product Model
HC32F460 Series	hc32f460jcta hc32f460jeta hc32f460jeua hc32f460kcta hc32f460keta hc32f460keua hc32f460pctb hc32f460pehb HC32F460PETB	**HC32F452 Series**	hc32f452feub hc32f452jeub hc32f452ketb hc32f452petb
HC32F451 Series	hc32f451feub hc32f451jeub hc32f451ketb HC32F451PETB	**HC32A460 Series**	HC32A460PETB

declaration Ming Dynasty (1368-1644)

★ Xiaohua Semiconductor Co., Ltd ("XHSC") reserves the right to make changes, corrections, enhancements, or modifications to the XHSC products and/or this document at any time without notice. Users are encouraged to obtain the most current information prior to placing an order.XHSC products are sold under the terms and conditions of sale set forth in the basic contract of purchase and sale.

★ Customer shall select the appropriate XHSC product for your application and design, validate and test your application to ensure that your application meets the appropriate standards and any safety, security or other requirements. The Customer shall be solely responsible for this.

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

★Resale of XHSC products under terms different from those set forth herein shall void any warranty commitment by XHSC with respect to such products.

★ Any graphics or lettering with the "®" 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 table of contents

1 summaries

This document mainly introduces the precautions and workarounds for the use of the HC32F460/ HC32F451/ HC32F452/ HC32A460 series chips.

2 caveat

2.1 System Notes

2.1.1 Abnormal operation after waking up from stop mode

- Description of the problem

When the chip wakes up from the stop mode, the Flash is not stabilized, which may cause the CPU to fetch the wrong finger and the chip to operate abnormally.

- workaround

The system clock switches to MRC before entering stop mode.

That is, before entering the stop mode, backup the user clock (non-MRC) switch the system clock to MRC and enter the stop mode; after exiting the stop mode, restore the user clock.

2.1.2 Stop Mode Wakeup Abnormal

- Description of the problem

The chip is in stop mode, and interrupt generation that is not enabled in the Stop Mode Wakeup Event Enable Register (INTC_WUPEN) may cause the chip to wake up from stop mode abnormally.

- workaround

Turn off the interrupt sources that are not enabled in the INTC_WUPEN register before entering the stop mode, and restore the turned off interrupt sources after exiting the stop mode.

2.1.3 Effect of Clock Divider Configuration Register on Power Down Mode

- Description of the problem

The reserved bits bit31~bit28 of Clock Division Frequency Configuration Register (CMU_SCFGR) set the clock division frequency smaller than that set by CMU_SCFGR.HCLK, and after exiting the power-down mode, the power-down mode cannot be re-entered.

- workaround

Reserved bits bit31~bit28 of CMU_SCFGR set the same frequency division as that set by CMU_SCFGR.HCLK.

2.1.4 CPU fetch error after wake-up from power-down mode

- Description of the problem

When the chip wakes up from the power-down mode, if the PWDN bit in the power-down mode control register 0 (PWRC0) is cleared to zero and reset again, it may cause the CPU to fetch

the wrong finger and the chip to operate abnormally.

- workaround

Places the function to enter power-down mode into RAM for execution.

2.1.5 System clock abnormality after waking up in power-down mode

■ Description of the problem

When the system clock is HRC or the system clock is MPLL and the clock source of MPLL is HRC, after the chip wakes up from power-down mode, if PWRC0.PWDN is reset after clearing (the same as the operation of [CPU Finger Fetch Error after Wake-up from Power-Down Mode]) the ICG loading action will be generated. If ICG1.HRCSTOP value is 1, i.e. HRC stop state, it will conflict with the current system clock of HRC.

■ workaround

The function to enter power-down mode is put into RAM for execution and the ICG1.HRCSTOP bit is configured to zero.

2.1.6 Port reset in power-down mode resets the RTC

■ Application Note

When the chip enters power-down mode, an external port reset is generated, which triggers a POR reset, causing the chip to initialize all the way, i.e., resetting modules such as RTCs that are not expected to be reset as well.

2.1.7 Programmable voltage detection flag bit clear

■ Application Note

The detection flag bits PVD1DETFLG and PVD2DETFLG in the PVD Detection Status Register (PWC_PVDDSR) are cleared by PVD1MON and PVD2MON, respectively, in this register (PWC_PVDDSR).

2.1.8 DMA Channel Enable

■ Description of the problem

After a DMA channel transfer is completed, the hardware automatically clears the corresponding bit of the channel enable register (DMA_CHEN) to zero. This may conflict with the software write enable DMA_CHEN.

■ workaround

Avoid performing RMW operations within this register when one or more bits of DMA_CHEN are automatically cleared to zero.

2.1.9 DMA Channel Configuration

■ Description of the problem

DMA The configuration of other channels in the same unit cannot be modified while a channel is in Block transfer.

■ workaround

After configuring the DMA registers, read the corresponding register values to determine whether the write is successful, if not, continue to write until successful or timeout processing.

2.1.10 EXTINT Usage Limitations

- Description of the problem

If the external interrupt function of the same Pin number is enabled at the same time, only the corresponding external interrupt channel with the smaller Pin number will be effective. For example, if PA0 and PB0 set PCRx0:INTE at the same time, only the external interrupt signal on PA0 can be responded.

- workaround

Each external interrupt channel can be configured with more than one I/O. Do not configure more than one pin to use external interrupts on the same channel, e.g., do not enable external interrupts for PA0 and PB0 at the same time.

2.1.11 Register Reserved Bit Considerations

- Application Note

Reserved bits in all registers (except those mentioned in this document) make sure that they remain at their default values throughout the use of the chip, otherwise unanticipated results may result.

2.2 SRAM Notes

2.2.1 SRAM3 Access Wait Cycle and ECC Functions

- Description of the problem
  - If the access wait cycle of SRAM3 is 0, a data read error occurs.
  - When the ECC function is enabled and the access wait cycle of SRAM3 is greater than 0, writing non-32-bit data to SRAM3 may cause problem 1) or problem 2)
    1) ECC False ECC Errors (1-bit ECC Errors or 2-bit ECC Errors) depending on the application setting, may generate an unintended NMI interrupt or system reset.
    2) A 1-bit ECC error occurs, the ECC function may write data without error correction to SRAM3, and the 1-bit ECC error flag bit is not set.
- workaround

Implement any of the following:

1) Set the access wait cycle of SRAM3 to at least 1 wait cycle to disable the ECC function.
2) Set the access wait cycle of SRAM3 to at least 1 wait cycle, enable the ECC function, and write data to SRAM3 with 32-bit address alignment only.

2.2.2 SRAM Cross-Region Access

- Description of the problem

Cross-region access between SRAMH (0x1FFF8000~0x1FFFFFFF) and SRAM1

(0x20000000~0x2000FFFF)

When the question is asked, a data read/write error occurs. For example, if a uint32_t* type pointer is defined to point to address 0x1FFFFFFE, a data error occurs when reading or writing a uint32_t type pointer to this address.

- workaround

Avoid cross-region access between SRAMH and SRAM1.

2.3 Timer0 Notes

2.3.1 Synchronous Mode Write Registers

- Application Note

A write operation to the CSTA/CSTB bits in the BCONR register under the synchronous clock requires a wait of 3 clock cycles for a successful write.

2.4 TimerA Notes

2.4.1 Input Capture

- Application Note

TimerA Only channel 4 can capture the rising/falling edge of TIMA_<t>_TRIG, which is enabled or disabled by channel 3's capture control register CCONR3.HICP4/HICP3.

2.4.2 PWM Output

- Description of the problem

When PCLK is used as the count clock and the number of clock divisions is not 0 (BCSTR.CKDIV[3:0]! = b'0000), it is not possible to specify the output level of the port when the counter is started (i.e., the PCONR.STAC[1:0]=b'00 or b'01 setting is invalid)

- workaround

When using PCLK as the counting clock for TimerA, to specify the output level of the port when the counter is started, you need to set the clock divider number of PCLK to 0 (BCSTR.CKDIV[3:0]=b'0000)

2.5 Timer6 Notes

2.5.1 TIM6_TRIGA~B Port Usage

- Application Note

The digital filter function of TIM6_TRIGA~B is set by FCONR of Unit 1.

The TIMER6_1 bit in function controller PWC_FCG2 needs to be cleared to zero when any unit of Timer6 uses the TIM6_TRIGA~B port.

2.6 USART Precautions

2.6.1 Notes on USART Transmit Air Disconnect

- Description of the problem

If the USART transmit function is already enabled (USART_CR1.TX=1), then enable transmit empty interrupt (USART_CR1.TXEIE=1), no transmit empty interrupt will be generated.

- workaround

If the USART transmit function is not enabled (USART_CR1.TX=0), enable the transmit function and transmit air-disconnect at the same time.
(USART_CR1|=0x00000088UL)

2.6.2 USART Single channel hardware flow control limit

- Description of the problem

Hardware flow control for CTS and RTS is not available for a single channel.

- workaround

The two USART channels use the CTS and RTS functions respectively.

2.7 I2S Notes

2.7.1 Slave receive data channel differentiation

- Application Note

When an I2S slave receives data, it cannot distinguish between data attributed to the left/right channel.

2.7.2 Slave data misalignment

- Application Note

When an I2S slave transmits data, a data shift (i.e., data bit and clock bit out of step)may occur due to data misalignment caused by external interference, andtheclock cannot be automatically synchronized via the WS line after the datashift.

2.8 QSPI Notes

2.8.1 4-wire read/write data

- Application Note

QSPI supports 4-wire read data, but not 4-wire write data, only single-wire write data.

2.8.2 Standard Read Maximum Rate

■ Description of the problem

QSPI standard read mode, read/write Flash data abnormally when QSPI clock frequency is high.

■ workaround

Select the appropriate reading mode according to the specifications of the actual connected QSPI-ROM.

2.8.3 Input Data Hold Time

■ Application Note

The QSPI interface feature requires a minimum data input hold time of 11ns under full temperature range conditions. When using QSPI to communicate with other devices, you need to check the manual of the corresponding device to see if it meets this feature.

2.9 I2C Notes

2.9.1 Host receive mode may send redundant clock signals

■ Description of the problem

After entering the host receive mode, when the DRR register or shift register is empty, the host will actively send a clock signal to read the data, and the application code may not be able to write 1 to the CR1.STOP flag bit in time to stop the clock signal from being sent, resulting in the issuance of an excess clock signal. If this situation causes the DRR register to have excess data that has not been read, a stop condition cannot be issued by writing 1 to the CR1.STOP flag bit at this time.

■ workaround

Combine the following two steps to circumvent:

1) When configured as a host, write 1 to the CR4.BUSWAIT register bit to enable the bus wait function. When this function is enabled, the host will not continue to send clock signals to read data when there is data in the DRR register that has not yet been read.
2) In the host receiving data flow, when the last 1Byte data enters the DRR register (judging that SR.RFULLF is 1) first write CR1.STOP register to send the stop condition to make the host exit the receiving mode, and then read the data in the DRR register.

2.9.2 Host receive mode cannot determine if the device address has been sent or not

■ Description of the problem

After writing the DTR register to send the device address, the I2C module immediately enters the host receive mode, at this time, the SR.TENDF register flag bit is invalid, and it is impossible to judge whether the address is sent or not by the SR.TENDF flag.

■ workaround

After writing the DTR register to send the device address, determine whether the address has

been sent by waiting for the SR.TRA flag bit to be 0.

2.9.3 Slave transmit mode may cause the bus to pull low

■ Description of the problem

In the slave transmit mode, if the slave continues to write data to the DTR register after receiving the NACK signal from the host, and this data is not read by the host, the slave will pull down the SDA signal, resulting in the host side not being able to issue a stop condition.

- workaround

Avoid writing the DTR register after the NACK signal, and if the bus cannot be released due to this operation, release the bus by I2C peripheral software reset.

2.9.4 Timeout Function Precautions

- Description of the problem

The timeout function registers SLTR.TOUTHIGH and SLTR.TOUTLOW are 16 bits wide and have a limited timeout function timing range.

- workaround

Software counting within the timeout interrupt realizes the functional requirements for longer timeout periods.

2.10 SPI Notes

2.10.1 SPI Host Mode Data Transmission Interval

- Application Note

Two consecutive data from the host will be separated by at least 3 SCK cycles and 2 PCLK cycles of time, resulting in a discontinuous SCK waveform.

2.10.2 SPI Slave Mode Data Interval

- Application Note

SPI slave mode requires that there must be 2 SCK cycles and 2 PCLK cycles between host frames or errors will occur.

2.11 SDIOC Notes

2.11.1 Long Response (136bit) Command Answer

- Description of the problem

The SDIOC hardware trims the last 1 Byte of data during a long response (136bit) command answer, causing the data to be shifted 8 bits to the right of the lower bit as a whole.

- workaround

Shift the data overall left by 8bit to the high side.

2.12 CAN Controller Precautions

2.12.1 Undefined frames are sent when the bus is disturbed

- Description of the problem

When the bus is disturbed, the CAN controller may send frames that are not defined by the

application when transmitting, including undefined IDs or undefined data.

■ workaround

Workarounds include the following:

1) Set the CAN control logic clock frequency to at least 1.5 times the communication clock frequency of CAN, for example, if the communication clock frequency is 20MHz, then the control logic clock frequency must be set to at least 30MHz. For example, if the communication clock frequency is 20MHz, then the control logic clock frequency must be set to 30MHz at the minimum, and the communication clock and control logic clock of each MCU series are shown in Table 2-4;

Table 2-1 CAN Controller Clock

MCU	communications clock	Control Logic Clock
HC32F460	External high speed crystal	EXCLK
HC32F452	External high speed crystal	EXCLK
HC32A460	External high speed crystal	EXCLK

2) After transmit enable, no data is filled into any transmit buffer and transmit is enabled until transmit is complete;
3) Add message validity confirmation mechanisms in the application program, such as adding handshake protocols, adding checksums for frames (including IDs and data), and determining whether newly received frames are adopted or not based on the system state.

2.12.2 Undefined waveforms are emitted to occupy the bus when the bus is disturbed.

■ Description of the problem

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

■ workaround

It is recommended to fill and send only one frame of data at a time, and do the send timeout processing. After the normal transmission is completed, you can directly continue to fill and send a new frame; if the transmission timeout, you need to re-initialize the CAN controller and wait for at least 11 CAN bits before sending.

The transmit timeout can be roughly calculated based on the

total number of bus nodes and the baud rate. The following

conditions are used as an example:

1) e.g. 10 bus nodes;
2) Baud rate 1Mbps;
3) The data length is 8 bytes, and the maximum time required for transmission is 140 μs.

Theoretically, under normal circumstances, it takes at least 1.4ms for all 10 nodes to send a frame

sequentially, so the timeout can be set to 2ms or longer. However, when the bus is interfered, the timeout time should be longer, such as 5ms.

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

1) Turns off the peripheral clock of the CAN controller (via the FCG register)
2) Enable the peripheral clock of the CAN controller (via the FCG register)
3) Initializes the CAN controller registers.

2.13 ADC Precautions

2.13.1 Interference with ADC input

- Description of the problem

When converting ADC channels, avoid PA0~PA7, PB0~PB2, PC4~PC5 port flip-flop, otherwise it will lead to poor conversion accuracy(The performance is: DC voltage input, the conversion result is a value that jumps within a certain range)

- workaround

Recommendation 1: Reduce the frequency and swing of the input clock to minimize the interference swing.

Recommendation 2: Multiple conversions are averaged.

Recommendation 3: Keep the input channels of the ADC away from the high-speed clock signals on the PCB.

2.14 TRNG Notes

2.14.1 At power-on reset, the random number generated by TRNG is a fixed value

- Description of the problem

At each power-on reset, the TRNG generates a random number with a fixed value.

- workaround

After power-on reset, the TRNG is activated to generate random numbers, discarding the first 10 random numbers and starting with the 11th random number.

Version Revision Record

version number	revision date	revision
Rev1.0	2023/06/05	First Edition Release.