1 or 2 high-level stop bits, support odd check/even check/mark check/blank check. The baud rate error of the signal sent by the serial port is less than 0.5%, and the allowable baud rate error of the signal received by the serial port is not more than 2%.

9.3.1 Baud rate calculation

1) Calculate the internal reference clock Fuart of the serial port, set the R8_UARTx_DIV register, the maximum value is 127, usually write 1.

2) Calculate the baud rate and set the R16_UARTx_DL register.

The baud rate formula = Fsys * 2 / R8_UARTx_DIV / 16 / R16_UARTx_DL.

9.3.2 Serial port transmission

The "THR register empty" interrupt sent by the serial port UART_II_THR_EMPTY means that the current transmit FIFO is empty. After reading the IIR register, the interrupt is cleared, or when the next data is written to the THR, the interrupt can also be cleared. If only one byte is written to THR, then since the byte is quickly transferred to the transmit shift register TSR to start transmission, the transmission will occur again soon THR register empty interrupt request, at this time, the next data to be sent can be written. When all the data in the TSR register is removed, the serial port transmission is truly completed, and the RB_LSR_TX_ALL_EMP bit of the LSR register becomes 1 valid.

In the interrupt trigger mode, after receiving the interrupt of the serial port sending the holding register THR empty, if the FIFO is enabled, you can write up to 8 bytes to the THR register and FIFO at a time, and then the controller will automatically send in order; if Disable FIFO, then you can only write one byte at a time; if there is no data to send, you can exit directly (it has been automatically cleared when reading IIR before Off).

In query mode, you can judge whether the transmit FIFO is empty according to the RB_LSR_TX_FIFO_EMP bit of the LSR register. When this bit is 1, you can write data to the THR register and FIFO. If the FIFO is enabled, you can write up to 8 bytes at a time .

You can also read the R8_UARTx_TFC register to determine the number of remaining data to be sent in the current FIFO. If it is not equal to 8, you can continue to write the data to be sent into the FIFO. This method can save filling time.

9.3.3 Serial port reception

The serial port receiving data available interrupt UART_II_RECV_RDY means that the number of data bytes in the receive FIFO has reached or exceeded the FIFO trigger point selected by the RB_FCR_FIFO_TRIG setting of the FCR register. This interrupt is cleared when the data read from RBR makes the FIFO word count lower than the FIFO trigger point.

Serial port receiving data timeout interrupt UART_II_RECV_TOUT means that there is at least one byte of data in the receive FIFO, and since the last time the serial port received the data and the last time the data was removed by the system, it has waited for the time equivalent to receiving 4 data. When again After receiving a new data, the interrupt is cleared, or when the RBR register is read once in a single chip, the interrupt can also be cleared. When the receive FIFO is completely empty, the RB_LSR_DATA_RDY bit of the LSR register is 0, and when there is data in the receive FIFO, the RB_LSR_DATA_RDY bit is 1 valid.

In the interrupt trigger mode, after receiving the interrupt of the serial port receiving data timeout, you can read the R8_UARTx_RFC register to query the current Count the remaining data in the front FIFO, read all the data directly, or continuously query the RB_LSR_DATA_RDY of the LSR register, if this bit is valid, read the data until this bit is invalid. After receiving the interrupt available for the serial port to receive data, you can first read the RBR register one-time Read RB_FCR_FIFO_TRIG to set the number of bytes of data, or you can read all the data in the current FIFO according to the RB_LSR_DATA_RDY bit and R8_UARTx_RFC register.

In the query mode, you can judge whether the receive FIFO is empty according to the RB_LSR_DATA_RDY bit of the LSR register, or read the R8_UARTx_RFC register to get the data count in the current FIFO to get all the data received by the serial port.

9.3.4 Hardware flow control

Hardware flow control includes automatic CTS (RB_MCR_AU_FLOW_EN is set to 1) and automatic RTS (RB_MCR_AU_FLOW_EN and RB_MCR_RTS are both set to 1)

If auto CTS is enabled, then the CTS pin must be valid before the serial port sends data. The serial port transmitter will detect the CTS pin before sending the next data. When the CTS pin status is valid, the transmitter will send the next data. In order to make the transmitter stop sending the following data, the CTS pin must be deactivated before the middle of the last stop bit of the current transmission. The automatic CTS function reduces

Interruption of the system application. When the hardware flow control is enabled, the controller will automatically control the transmitter according to the state of the CTS pin, so the change of the CTS pin level will not trigger the MODEM interrupt. If automatic RTS is enabled, the RTS pin output is enabled only when there is enough space in the FIFO to receive data, and the RTS pin output is disabled when the receive FIFO is full. If all the data in the receive FIFO is removed or emptied, then the RTS pin output is valid. When the trigger point of the receive FIFO is reached (the number of bytes in the receive FIFO is not less than the number of bytes set by the RB_FCR_FIFO_TRIG of the FCR register), the RTS pin output is invalid, and the other transmitter is allowed to wait until the RTS pin is invalid. Send an additional data. Once the receive FIFO is empty of data, the RTS pin will be automatically re-enabled, so that the other party's transmitter resumes sending. If both automatic CTS and automatic RTS are enabled (RB_MCR_AU_FLOW_EN and RB_MCR_RTS in the MCR register are both 1), then when one's RTS pin is connected to the other party's CTS pin, unless there is enough space in the other party's receive FIFO, the other party No data will be sent. Therefore, through this hardware flow control, it is possible to avoid FIFO overflow and

Timeout error.

Chapter 10 Serial Peripheral Interface SPI

10.1 Introduction to SPI

SPI is a full-duplex serial interface. There is a master and several slaves connected to the bus. At the same time, only a pair of masters and slaves are communicating. News. Usually SPI interface consists of 4 pins: SPI chip select pin SCS, SPI clock pin SCK, SPI serial data pin MISO (master input/slave output pin) and SPI serial data pin MOSI (master Output/slave input pin).

10.1.1 Main features

The chip provides 2 SPI interfaces, the characteristics are as follows:

- SPI0 supports master mode (Master) and slave mode (Slave) , SPI1 only supports the master mode (Master).
- Compatible with Serial Peripheral Interface (SPI) specification.
- Support mode 0 and mode 3 data transmission methods.
- 8-bit data transmission mode, data bit sequence is optional: low byte first or high byte first.
- The clock frequency can reach up to half of the system's main frequency Fsys.
- 8-byte FIFO.
- SPI0 slave mode supports command mode or data stream mode for the first byte.
- SPI0 supports DMA, and the data transmission efficiency is higher.

10.2 Register description

Table 10-1 List of SPI0 related registers

name	address	description	Reset value
R8_SPI0_CTRL_MOD	0x40004000	SPI0 mode control register	0x02
R8_SPI0_CTRL_CFG	0x40004001	SPI0 configuration register	0x00
R8_SPI0_INTER_EN	0x40004002	SPI0 interrupt enable register	0x00
R8_SPI0_CLOCK_DIV R8_SPI0_SLAVE_PRE	0x40004003	SPI0 master mode clock divider register SPI0 slave mode preset data register	0x10
R8_SPI0_BUFFER	0x40004004	SPI0 data buffer	0xXX
R8_SPI0_RUN_FLAG	0x40004005	SPI0 working status register	0x00
R8_SPI0_INT_FLAG	0x40004006	SPI0 interrupt flag register	0x40
R8_SPI0_FIFO_COUNT	0x40004007	SPI0 Transceiver FIFO count register	0x00
R16_SPI0_TOTAL_CNT	0x4000400C	SPI0 Total Length of Transceived Data Register	0x0000
R8_SPI0_FIFO	0x40004010	SPI0 data FIFO register	0xXX
R8_SPI0_FIFO_COUNT1	0x40004013	SPI0 Transceiver FIFO count register	0x00
R16_SPI0_DMA_NOW	0x40004014	SPI0 DMA buffer current address	0xXXXX
R16_SPI0_DMA_BEG	0x40004018	SPI0 DMA buffer start address	0xXXXX
R16_SPI0_DMA_END	0x4000401C	SPI0 DMA buffer end address	0xXXXX

Table 10-2 SPI1 related register list

name	address	description	Reset value
R8_SPI1_CTRL_MOD	0x40004400	SPI1 mode control register	0x02
R8_SPI1_CTRL_CFG	0x40004401	SPI1 configuration register	0x00
R8_SPI1_INTER_EN	0x40004402	SPI1 interrupt enable register	0x00
R8_SPI1_CLOCK_DIV	0x40004403	SPI1 master mode clock divider register	0x10
R8_SPI1_BUFFER	0x40004404	SPI1 data buffer	0xXX
R8_SPI1_RUN_FLAG	0x40004405	SPI1 working status register	0x00

R8_SPI1_INT_FLAG	0x40004406	SPI1 interrupt flag register	0x40
R8_SPI1_FIFO_COUNT	0x40004407	SPI1 Transceiver FIFO count register SPI1	0x00
R16_SPI1_TOTAL_CNT	0x4000440C	Transceive data total length register	0x00
R8_SPI1_FIFO	0x40004410	SPI1 Data FIFO Register SPI1	0xXX
R8_SPI1_FIFO_COUNT1	0x40004413	Transceiver FIFO Count Register	0x00

SPI Mode control system Register (R8_SPIx_CTRL_M OD) (x=0/1)

Bit	name	access	description	Reset value
7	RB_SPI_MISO_OE	RW	MISO pin output enable bit (can be used in 2-wire mode data line switching direction): 1: MISO output is enabled; 0: MISO output is disabled.	0
6	RB_SPI_MOSI_OE	RW	MOSI pin output enable bit: 1: MOSI output is enabled; 0: MOSI output is disabled.	0
5	RB_SPI_SCK_OE	RW	SCK pin output enable bit: 1: SCK output is enabled; 0: SCK output is disabled.	0
4	RB_SPI_FIFO_DIR	RW	FIFO direction setting bit: 1: Input mode (indicating to receive data); 0: Output mode (indicating sending data).	0
3	RB_SPI_SLV_CMD_MOD	RW	The first byte mode selection in SPI0 slave mode (only SPI0 support): 1: First byte command mode; 0: Data flow mode. In the first byte command mode, when the SPI chip is received After selecting the valid first byte of data, it will be regarded as a command Code, and RB_SPI_IF_FST_BYTE will be set to 1.	0
3	RB_SPI_MST_SCK_MOD	RW	Master mode clock idle mode selection: 1: Mode 3 (SCK is high when idle); 0: Mode 0 (SCK is low when idle). Slave mode 2-wire or	0
2	RB_SPI_2WIRE_MOD	RW	3-wire SPI mode selection (only SPI0 supports, SPI1 does not need this control bit) ： 1: 2-wire mode/half duplex (SCK/MISO) ； 0: 3-wire mode/full duplex (SCK/MOSI/MISO) .	0
1	RB_SPI_ALL_CLEAR	RW	SPI's FIFO/counter/interrupt flag is cleared: 1: Forced clearing and clearing; 0: Uncleared.	1
0	RB_SPI_MODE_SLAVE	RW	SPI0 master-slave mode selection (only supported by SPI0): 1: slave mode; 0: master mode.	0

SPI Configuration send Save (R8_SPIx_CTRL_CFG) (x=0/1)

Bit	name	access	description	Reset value
7	Reserved	R0	Reserved.	0
6	RB_SPI_MST_DLY_EN	RW	Input delay enable in host mode: 1: Enable, used for high-speed applications such as SPI clock close to half of Fsys; 0: Prohibited, normal application.	0
5	RB_SPI_BIT_ORDER	RW	SPI data bit sequence selection: 1: Low order first; 0: High order first.	0
4	RB_SPI_AUTO_IF	RW	Enable to automatically clear the flag when accessing BUFFER/FIFO Function of bit RB_SPI_IF_BYTE_END:	0

			1: enable; 0: Prohibited.	
3	Reserved	R0	Reserved.	0
2	RB_SPI_DMA_LOOP	RW	DMA address cycle function enable bit (only supported by SPI0): 1: Enable address loop; 0: Disable address loop. If the DMA address loop is enabled, when the DMA address increases to the end address set, it will automatically loop to Set the first address.	0
1	Reserved	R0	Reserved.	0
0	RB_SPI_DMA_ENABLE	RW	DMA function enable bit (only supported by SPI0) ： 1: Enable DMA; 0: Disable DMA.	0

SPI Interrupt enable can Register (R8_SPIx_INTER_ EN) (x=0/1)

Bit	name	access	description	Reset value
7	RB_SPI_IE_FST_BYTE	RW	In the first byte command mode of the slave mode, the first byte Receive interrupt enable bit (only supported by SPI0)： 1: Enable to receive the first byte interrupt; 0: Disable receiving the first byte interrupt.	0
[6:5]	Reserved	R0	Reserved.	00b
4	RB_SPI_IE_FIFO_OV	RW	FIFO overflow (FIFO is full when receiving or FIFO when sending Empty) Interrupt enable bit (only supported by SPI0)： 1: Enable interrupt; 0: Disable interrupts.	0
3	RB_SPI_IE_DMA_END	RW	DMA end interrupt enable bit (only supported by SPI0)： 1: Enable interrupt; 0: Disable interrupts.	0
2	RB_SPI_IE_FIFO_HF	RW	FIFO uses more than half of the interrupt enable bits: 1: Enable interrupt; 0: Disable interrupts.	0
1	RB_SPI_IE_BYTE_END	RW	SPI single byte transfer complete interrupt enable bit: 1: Enable interrupt; 0: Disable interrupts.	0
0	RB_SPI_IE_CNT_END	RW	SPI all byte transfer complete interrupt enable bit: 1: Enable interrupt; 0: Disable interrupts.	0

SPI Host model formula Clock divider register (R8_SPIx_CLOC K_DIV) (x=0/1)

Bit	name	access	description	Reset value
[7:0]	R8_SPI_CLOCK_DIV	RW	The frequency division coefficient of the host mode, the minimum value is 2. Fsck = system main frequency Fsys / frequency division coefficient.	10h

SPI Slave model formula Preset data register (R8_SPI0_SLAV E_PRE) (only supported by SPI0)

Bit	name	access	description	Reset value
[7:0]	R8_SPI0_SLAVE_PRE	RW	In slave mode, the first preset return data. Used to return data after receiving the first byte of data.	10h

SPI Data slow Bash area (R8_SPIx_BUFFER) (x =0/1)

Bit	name	access	description	Reset value
[7:0]	R8_SPIx_BUFFER	RW	SPI data transmission and reception buffer.	XXh

SPI Work status state Register (R8_SPIx_RUN_FL AG) (x=0/1)

Bit	name	access	description	Reset value
7	RB_SPI_SLV_SELECT	RO	Slave mode is chip select status bit (only supported by SPI0):	0

			1: Being selected; 0: Not selected by chip.	
6	RB_SPI_SLV_CS_LOAD	RO	Load the status bit for the first time after the slave mode is chip selected (only SPI0 support) 1: Loading R8_SPI0_SLAVE_PRE; 0: Not yet loaded or completed.	0
5	RB_SPI_FIFO_READY	RO	FIFO ready status bit: 1: FIFO is ready (R16_SPIx_TOTAL_CNT is not 0, and the FIFO is not full when receiving or the FIFO is not empty when sending); 0: FIFO is not ready.	0
4	RB_SPI_SLV_CMD_ACT	RO	In the slave mode, the command receiving completion status bit, that is, the After changing the first byte of data (only supported by SPI0) 1: Indicates that the first byte that has just been exchanged; 0: The first byte has not been exchanged or is not the first byte.	0
[3:0]	Reserved	R0	Reserved.	0000b

SPI Interrupt mark Zhi Register (R8_SPIx_INT_FL AG) (x=0/1)

Bit	name	access	description	Reset value
7	RB_SPI_IF_FST_BYTE	RW1	In slave mode, the flag bit of the first byte is received (only SPI0 support) 1: The first byte has been received; 0: The first byte has not been received.	0
6	RB_SPI_FREE	RO	Current SPI idle status bit: 1: The current SPI is idle; 0: The current SPI is not idle.	1
5	Reserved	RO	Reserved.	0
4	RB_SPI_IF_FIFO_OV	RW1	FIFO overflow (FIFO is full when receiving or FIFO when sending Empty) flag bit, write 1 to clear: 1: FIFO overflow; 0: FIFO has not overflowed.	0
3	RB_SPI_IF_DMA_END	RW1	DMA complete flag (only supported by SPI0) , Write 1 Cleared: 1: Completed; 0: Not completed.	X
2	RB_SPI_IF_FIFO_HF	RW1	FIFO used more than half (FIFO>=4 when receiving or FIFO<4 when sending) flag bit, write 1 to clear: 1: FIFO is used more than half; 0: FIFO is not more than half used.	0
1	RB_SPI_IF_BYTE_END	RW1	SPI single byte transfer complete flag bit, write 1 to clear: 1: Single byte transmission is completed; 0: transmission is not completed.	0
0	RB_SPI_IF_CNT_END	RW1	SPI all bytes transfer completed flag bit, write 1 to clear: 1: Transmission of all bytes is completed; 0: Transmission is not completed.	1

SPI Transceiver FIFO Counting register (R8_SPIx_FIFO_CO UNT) (x=0/1)

Bit	name	access	description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT	RW	The current byte count in the FIFO.	00h

SPI Transceiver FIFO Counting register (R8_SPIx_FIFO_CO UNT1) (x=0/1)

Bit	name	access	description	Reset value
[7:0]	R8_SPIx_FIFO_COUNT1	RW	The current byte count in the FIFO. Same as R8_SPIx_FIFO_COUNT.	00h

SPI Number of sending and receiving according to Total length register (R16_SPIx_TOTAL _CNT) (x=0/1)

Bit	name	access	description	Reset value
[15:0]	R16_SPIx_TOTAL_CNT	RW	The total number of bytes of SPI data sent and received in the host mode, low 12 bits are valid. When using DMA, up to 4095 bytes can be sent and received at a time. Slave mode is not supported.	0000h

SPI data FIFO Register (R8_SPIx_FIFO) (x=0/ 1)

Bit	name	access	description	Reset value
[7:0]	R8_SPIx_FIFO	RO/ WO	Data FIFO register.	XXh

The registers R8_SPIx_BUFFER and R8_SPIx_FIFO are SPI data related registers, the main difference is:

Reading R8_SPIx_BUFFER is taken from the last data exchanged by SPI, and does not affect FIFO and R8_SPIx_FIFO_COUNT. Writing R8_SPIx_BUFFER in master mode is to send the byte directly, and the write operation in slave mode is undefined;

Reading R8_SPIx_FIFO is taken from the earliest exchanged data in FIFO, which will reduce FIFO and R8_SPIx_FIFO_COUNT,

Writing R8_SPIx_FIFO is to temporarily store the data in the FIFO. In the slave mode, the external SPI master decides when to take it. In the master mode, the transmission is automatically started when R16_SPIx_TOTAL_CNT is not 0.

SPI0 DMA buffer Area current address (R16_SPI0_DMA_NO W)

Bit	name	access	description	Reset value
[15:0]	R16_SPI0_DMA_NOW	RW	The current address of the DMA data buffer. Can be used to calculate the number of conversions, calculation method: COUNT=SPI0_DMA_NOW-SPI0_DMA_BEG.	XXXXh

SPI0 DMA buffer Zone start address (R16_SPI0_DMA_BE G)

Bit	name	access	description	Reset value
[15:0]	R16_SPI0_DMA_BEG	RW	The starting address of the DMA data buffer, only the lower 15 bits are valid.	XXXXh

SPI0 DMA buffer Zone end address (R16_SPI0_DMA_EN D)

Bit	name	access	description	Reset value
[15:0]	R16_SPI0_DMA_END	RW	DMA data buffer end address (not included) , Only low 15 bits are valid.	XXXXh

10.3 SPI transmission format

SPI supports mode 0 and mode 3 transmission formats, which can be selected by setting RB_SPI_MST_SCK_MOD in R8_SPIx_CTRL_MOD. The serial data input is always sampled on the rising edge of SCK, and serial data is output on the falling edge.

The data transmission format is shown in the figure below:

Mode 0: RB_SPI_MST_SCK_MOD = 0

Mode 0 timing diagram

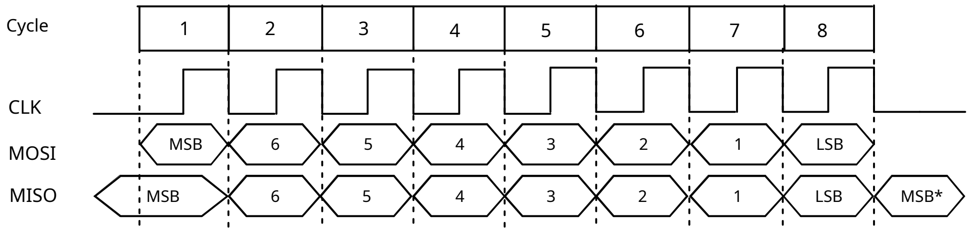


Figure 10-1 SPI mode 0 transmission format

Mode 3: RB_SPI_MST_SCK_MOD = 1

Mode 3 timing diagram

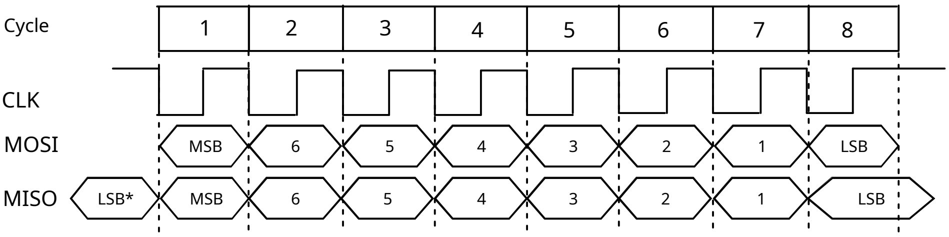


Figure 10-2 SPI mode 3 transmission format

10.4 SPI configuration

10.4.1 SPI master mode

In SPI master mode, the serial clock is generated on the SCK pin, and the chip select pin can be designated as any I/O pin.

Configuration steps:

(1), set R8_SPIx_CLOCK_DIV, configure the SPI clock frequency;

(2) Set RB_SPI_MODE_SLAVE of R8_SPIx_CTRL_MOD to 0, configure SPI as the master mode; (3) Set RB_SPI_MST_SCK_MOD of R8_SPIx_CTRL_MOD, select clock idle mode 0 or mode 3;

(4). Set the RB_SPI_FIFO_DIR of R8_SPIx_CTRL_MOD to configure the FIFO direction. If it is 1, the FIFO is used for receiving, and if it is 0, the FIFO is used for sending.

(5) Set the RB_SPI_MOSI_OE and RB_SPI_SCK_OE of R8_SPIx_CTRL_MOD to 1, RB_SPI_MISO_OE to 0, and set the GPIO direction configuration register (R32_PA/PB_DIR) to make the MOSI pin and SCK pin output, and the MISO pin as input; (6), 2 SCK remains unchanged in wire mode, RB_SPI_MOSI_OE=0, MOSI is not used, input is realized by MISO half-duplex (same as 3-wire mode, RB_SPI_MISO_OE=0 and the pin is set as input) and output (RB_SPI_MISO_OE=1 and the pin is set as output) , Switch the direction manually; (7). Optional step. If DMA is enabled, write the start address of the transceiver buffer to R16_SPI_DMA_BEG, and write the end address (not included) to R16_SPI_DMA_END. It is recommended to set RB_SPI_DMA_ENABLE after setting RB_SPI_FIFO_DIR. If you confirm that R16_SPIx_TOTAL_CNT is 0, you can also set RB_SPI_DMA_ENABLE to 1 here to enable the DMA function.

Data sending process:

(1) Set RB_SPI_FIFO_DIR to 0, and the current FIFO direction is output; (2) Write to the R16_SPIx_TOTAL_CNT register to set the length of the data to be sent;

(3) Write the R8_SPIx_FIFO register to write the data to be sent to the FIFO. If R8_SPIx_FIFO_COUNT is less than the FIFO capacity, you can continue to write to the FIFO. If DMA is enabled, the DMA will automatically load the FIFO to complete this step;

(4) As long as R16_SPIx_TOTAL_CNT is not 0 and there is data in the FIFO, the SPI host will automatically send the data, otherwise it will pause; (5) Wait for the R16_SPIx_TOTAL_CNT register to be 0, indicating that the data transmission is complete. If only one byte is sent, you can also query Wait for RB_SPI_FREE to be free, or wait for R8_SPIx_FIFO_COUNT to be 0.

Data receiving process:

(1) Set RB_SPI_FIFO_DIR to 1, and the current FIFO direction is input; (2) Write to the R16_SPIx_TOTAL_CNT register to set the data length to be received;

(3) As long as R16_SPIx_TOTAL_CNT is not 0 and the FIFO is not full, the SPI host will automatically receive data, otherwise it will pause; (4) Wait for the R8_SPIx_FIFO_COUNT register to be non-zero, indicating that the return data is received. Read the value in R8_SPIx_FIFO. If the received data is DMA enabled, DMA will automatically read the FIFO to complete this step.

10.4.2 SPI slave mode

SPI0 supports the slave mode. In the slave mode, the SCK pin is used to receive the serial clock of the externally connected SPI master.

Configuration steps:

(1) Set the RB_SPI_MODE_SLAVE of R8_SPI0_CTRL_MOD to 1, and configure SPI0 as the slave mode; (2) Set the RB_SPI_SLV_CMD_MOD of R8_SPI0_CTRL_MOD as needed to select the first byte mode or data stream mode of the slave; (3) Set the RB_SPI_DIR of R8_SPI0_CTRL_MOD_DIR, Configure the FIFO direction, if it is 1, FIFO is used for receiving, and if it is 0, FIFO is used for sending;

(4) Set the RB_SPI_MOSI_OE and RB_SPI_SCK_OE of R8_SPI0_CTRL_MOD to 0, RB_SPI_MISO_OE to 1, and set the GPIO direction configuration register (R32_PA/PB_DIR) to make the MOSI pin, SCK pin and SCS pin as input, and MISO pin as input (support Multiple slaves are connected under the bus, MISO will automatically switch to output after chip selection, and also supports one master and one slave) or output (only Used for a master-slave connection)In SPI slave mode, the I/O pin direction of MISO can be set by the GPIO direction configuration register. In addition to output, it also supports automatic switching to output during SPI chip selection, but its output data is selected by RB_SPI_MISO_OE. When it is 1, it outputs SPI data, and when it is 0, it outputs GPIO data output register data. It is recommended to set the MISO pin as input, so that MISO does not output when the chip select is invalid, so that it is convenient to share the SPI bus during multi-machine operation;

(5). Optionally, set the SPI0 slave mode preset data register R8_SPI0_SLAVE_PRE to be automatically loaded into the buffer for the first time after chip selection for external output. When 8 clocks have passed (that is, the first data byte is exchanged between the master and the slave), the control The device gets the first byte of data (command code) from the external SPI host, The external SPI host exchanges the data in R8_SPI0_SLAVE_PRE Preset data (status value). Bit 7 of R8_SPI0_SLAVE_PRE will be automatically loaded during the SCK low level after the SPI chip select is valid. On the MISO pin, for SPI mode 0 (CLK defaults to low level) , If bit 7 of R8_SPI0_SLAVE_PRE is preset, then external When the SPI chip select is valid but the SPI master has not yet transmitted data,You can get R8_SPI0_SLAVE_PRE by querying the MISO pin The preset value of bit 7 of R8_SPI0_SLAVE_PRE can be quickly obtained by only validating the SPI chip select (usually to provide a busy state to the host for quick query by the host) ;

(6). Optional step. If DMA is enabled, write the start address of the transceiver buffer to R16_SPI_DMA_BEG, and write the end address (not included) to R16_SPI_DMA_END. You must set RB_SPI_DMA_ENABLE after setting RB_SPI_FIFO_DIR.

Data sending process:

(1). Set the RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to 0, and the current FIFO direction is output; (2). Optional step. If DMA is enabled, then set RB_SPI_DMA_ENABLE to 1 to enable the DMA function; (3) Send multiple data Write to the FIFO register R8_SPI0_FIFO, and the external SPI host decides when to take it. If DMA is enabled, DMA will automatically load FIFO to complete this step;

(4). Query R8_SPI0_FIFO_COUNT, if it is not full, continue to write the data to be sent to the FIFO.

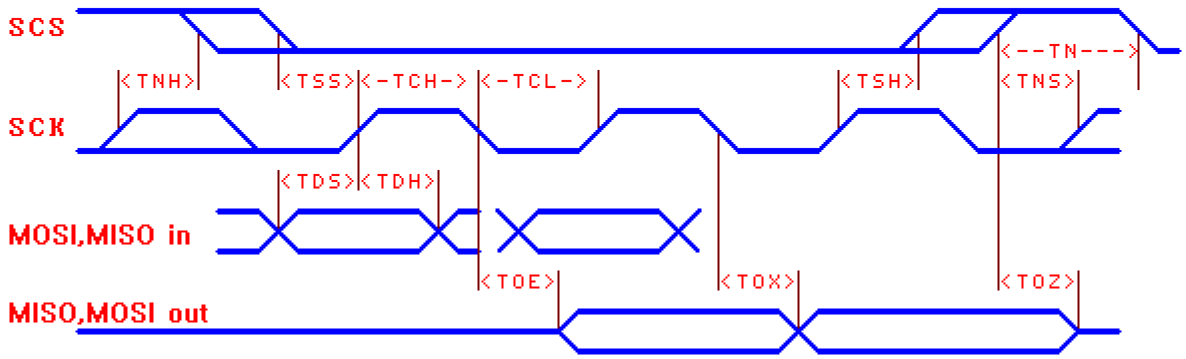
Data receiving process:

(1) Set the RB_SPI_FIFO_DIR of R8_SPI0_CTRL_MOD to 1, and the current FIFO direction is input; (2) Optional step, if DMA is enabled, then set RB_SPI_DMA_ENABLE to 1 to enable the DMA function;

(3) Query R8_SPI0_FIFO_COUNT. If it is not empty, it means that the data has been received. The data is taken by reading R8_SPI0_FIFO. If DMA is enabled, DMA will automatically read FIFO to complete this step;

(4). For single byte data reception, FIFO is not required, and R8_SPIx_BUFFER can be read directly.

10.5 SPI timing



name	Parameter description (TA=25 °C, VIO33=3.3V)	Minimum	Typical value	Max	unit
TSS	SCS valid setup time before SCK rising edge	Tsys*1.05			nS
TSH	SCK valid hold time after rising edge SCK	Tsys*1.05			nS
TNS	SCS invalid setup time before rising edge	15			nS
TNH	SCK invalid hold time after SCK rising edge	15			nS
TN	SCS invalid time (SPI operation interval time)	Tsys*2			nS
TCH	SCK clock high time	Tsys*0.55			nS
TCL	SCK clock low time	Tsys*0.55			nS
TDS	MOSI/MISO input setup time before SCK rising edge	8			nS
TDH	SCK MOSI/MISO input hold time after SCK rising edge	5			nS
TOE	SCK falling edge to MISO/MOSI output valid	0		18	nS
TOX	SCK falling edge to MISO/MOSI output change	0	5	16	nS
TOZ	SCS invalid to MISO/MOSI output invalid	2		twenty four	nS

Note: Tsys is the system clock cycle (1/Fsys).

Chapter 11 PWM

11.1 Introduction to PWM Controller

In addition to the 4-channel 26-bit PWM output provided by the timer, the system also provides 8 8-bit PWM outputs (PWM4 ～PWM11) ， The duty cycle is adjustable, the PWM cycle is fixed and 8 cycles can be selected, and the operation is simple.

11.2 Register description

Table 11-1 List of PWMx related registers

name	address	description	Reset value
R8_PWM_OUT_EN	0x40005000	PWMx output enable register	0x00
R8_PWM_POLAR	0x40005001	PWMx output polarity configuration register	0x00
R8_PWM_CONFIG	0x40005002	PWMx configuration control register	0x0X
R8_PWM_CLOCK_DIV	0x40005003	PWMx clock divider register	0x00
R32_PWM4_7_DATA	0x40005004	PWM4/5/6/7 data holding register	0xXXXXXXXX
R8_PWM4_DATA	0x40005004	PWM4 data holding register	0xXX
R8_PWM5_DATA	0x40005005	PWM5 data holding register	0xXX
R8_PWM6_DATA	0x40005006	PWM6 data holding register	0xXX
R8_PWM7_DATA	0x40005007	PWM7 data holding register	0xXX
R32_PWM8_11_DATA	0x40005008	PWM8/9/10/11 data holding register	0xXXXXXXXX
R8_PWM8_DATA	0x40005008	PWM8 data holding register	0xXX
R8_PWM9_DATA	0x40005009	PWM9 data holding register	0xXX
R8_PWM10_DATA	0x4000500A	PWM10 data holding register	0xXX
R8_PWM11_DATA	0x4000500B	PWM11 data holding register	0xXX

PWMx output enable Register (R8_PWM_OUT_EN)

Bit	name	access	description	Reset value
7	RB_PWM11_OUT_EN	RW	PWM11 output enable bit: 1: enable; 0: Prohibited.	0
6	RB_PWM10_OUT_EN	RW	PWM10 output enable bit: 1: enable; 0: Prohibited.	0
5	RB_PWM9_OUT_EN	RW	PWM9 output enable bit: 1: enable; 0: Prohibited.	0
4	RB_PWM8_OUT_EN	RW	PWM8 output enable bit: 1: enable; 0: Prohibited.	0
3	RB_PWM7_OUT_EN	RW	PWM7 output enable bit: 1: enable; 0: Prohibited.	0
2	RB_PWM6_OUT_EN	RW	PWM6 output enable bit: 1: enable; 0: Prohibited.	0
1	RB_PWM5_OUT_EN	RW	PWM5 output enable bit: 1: enable; 0: Prohibited.	0
0	RB_PWM4_OUT_EN	RW	PWM4 output enable bit: 1: enable; 0: Prohibited.	0

PWMx output polarity Configuration register (R8_PWM_PO LAR)

Bit	name	access	description	Reset value

7	RB_PWM11_POLAR	RW	PWM11 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
6	RB_PWM10_POLAR	RW	PWM10 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
5	RB_PWM9_POLAR	RW	PWM9 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
4	RB_PWM8_POLAR	RW	PWM8 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
3	RB_PWM7_POLAR	RW	PWM7 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
2	RB_PWM6_POLAR	RW	PWM6 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
1	RB_PWM5_POLAR	RW	PWM5 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0
0	RB_PWM4_POLAR	RW	PWM4 output polarity control bit: 1: Default high level, low effective; 0: Default low level, high effective.	0

PWMx configuration control Register (R8_PWM_CONFIG)

Bit	name	access	description	Reset value
7	RB_PWM10_11_STAG_EN	RW	PWM10/11 interleaved output enable bit: 1: Interleaved output; 0: Independent output.	0
6	RB_PWM8_9_STAG_EN	RW	PWM8/9 interleaved output enable bit: 1: Interleaved output; 0: Independent output.	0
5	RB_PWM6_7_STAG_EN	RW	PWM6/7 interleaved output enable bit: 1: Interleaved output; 0: Independent output.	0
4	RB_PWM4_5_STAG_EN	RW	PWM4/5 interleaved output enable bit: 1: Interleaved output; 0: Independent output.	0
[3:2]	RB_PWM_CYC_MOD	RW	PWM data width selection: 00: 8-bit data width; 01: 7-bit data width; 10: 6-bit data width; 11: 5-bit data width.	00b
1	RB_PWM_STAG_ST	RO	PWM interleaving flag: 1: Indicate that PWM5/7/9/11 is allowed to output; 0: Indicate that PWM4/6/8/10 is allowed to output;	X
0	RB_PWM_CYCLE_SEL	RW	PWM period selection, matching with PWM data width: 1:255/127/63/31 clock periods; 0: 256/128/64/32 clock cycles.	0

PWMx clock divider Register (R8_PWM_CLOCK_ DIV)

Bit	name	access	description	Reset value
[7:0]	R8_PWM_CLOCK_DIV	RW	The frequency division factor of the PWM base clock.	00h

			Fpwm = Fsys / R8_PWM_CLOCK_DIV.	

PWM Data retention Register group 1 (R32_PWM4_7_DATA)

Bit	name	access	description	Reset value
[31:24]	R8_PWM7_DATA	RW	PWM7 data holding register.	XXb
[23:16]	R8_PWM6_DATA	RW	PWM6 data holding register.	XXb
[15:8]	R8_PWM5_DATA	RW	PWM5 data holding register.	XXb
[7:0]	R8_PWM4_DATA	RW	PWM4 data holding register.	XXb

PWM Data retention Register Group 2 (R32_PWM8_11_DATA)

Bit	name	access	description	Reset value
[31:24]	R8_PWM11_DATA	RW	PWM11 data holding register.	XXb
[23:16]	R8_PWM10_DATA	RW	PWM10 data holding register.	XXb
[15:8]	R8_PWM9_DATA	RW	PWM9 data holding register.	XXb
[7:0]	R8_PWM8_DATA	RW	PWM8 data holding register.	XXb

11.3 PWM configuration

(1) Set the register R8_PWM_CLOCK_DIV to configure the reference clock frequency of PWM;

(2) Set the PWM output polarity configuration register R8_PWM_POLAR, configure the output polarity corresponding to PWMx;

(3) Set the PWM configuration control register R8_PWM_CONFIG, set the PWM mode, data width, and period; (4) Set the PWM output The enable register R8_PWM_OUT_EN turns on the corresponding PWMx output enable; (5). Calculate the data according to the required duty cycle and write it into the corresponding data holding register R8_PWMx_DATA;

(6). Set the required PWM pin direction in PWM4-PWM11 as output. Optionally, set the drive capability of the corresponding I/O;

(7). Update the data in R8_PWMx_DATA as needed and update the output duty cycle.

Adjust the R8_PWMx_DATA register to modify the duty cycle of the output PWM. The duty cycle calculation formula: Ncyc = RB_PWM_CYCLE_SEL? (2^n̂-1): (2^n)

(Where n=data bit width)，Ncyc results in 63 ～256.

The duty cycle of PWMx output effective level = R8_PWMx_DATA / Ncyc * 100%

PWMx output frequency Fpwmout = Fpwm / Ncyc = Fsys / R8_PWM_CLOCK_DIV / Ncyc

If you need to generate a DC signal through PWM, you can use R/C to filter the PWMx output. It is recommended to use a two-stage RC with a time constant much greater than 4 / Fpwmout, or a one-stage RC with a time constant much greater than 100 / Fpwmout.

Chapter 12 LED Screen Controller

12.1 Introduction to LED Controller

The chip provides an LED screen control card interface, built-in 2 bytes FIFO, supports DMA and interrupts, saves CPU processing time, and supports 1/2/4 data line interfaces.

12.2 Register description

Table 12-1 LED related register list

name	address	description	Reset value
R8_LED_CTRL_MOD	0x40006400	LED mode configuration register	0x02
R8_LED_CLOCK_DIV	0x40006401	LED serial clock divider register	0x10
R8_LED_STATUS	0x40006404	LED status register	0xX0
R16_LED_FIFO	0x40006408	LED data FIFO register	0xXXXX
R16_LED_DMA_CNT	0x40006410	LED DMA remaining count register	0x0000
R16_LED_DMA_MAIN	0x40006414	LED main buffer DMA address	0xXXXX
R16_LED_DMA_AUX	0x40006418	LED auxiliary buffer DMA address	0xXXXX

led Mode configuration Set Register (R8_LED_CTRL_MO D)

Bit	name	access	description	Reset value

[7:6]	RB_LED_CHAN_MOD	RW	LED channel mode selection:
			00: LED0, single channel output;
			01: LED0/1, dual-channel output;
			10: LED0 ～3, 4 channel output;
			11: LED0 ～3, 4-channel output, among which LED2/3 Channel data source auxiliary buffer.
				00b

5	RB_LED_IE_FIFO	RW	FIFO counting over half interrupt enable: 1: FIFO count<=2 interrupt trigger; 0: Disable interrupts.	0
4	RB_LED_DMA_EN	RW	LED DMA function and DMA interrupt enable: 1: enable; 0: Prohibited.	0
3	RB_LED_OUT_EN	RW	LED signal output enable: 1: enable; 0: Prohibited.	0
2	RB_LED_OUT_POLAR	RW	LED data output polarity control bit: 1: Reverse, data 0 output 1, data 1 output 0; 0: pass-through, data 0 output 0, data 1 output 1. Clear	0
1	RB_LED_ALL_CLEAR	RW	the FIFO/counter/interrupt flag of the LED: 1: Forced clearing and clearing; 0: Uncleared.	1
0	RB_LED_BIT_ORDER	RW	LED serial data bit sequence selection: 1: Low order first; 0: High order first.	0

led Serial bell Frequency divider register (R8_LED_CLOCK_DIV)

Bit	name	access	description	Reset value
[7:0]	R8_LED_CLOCK_DIV	RW	LED output clock division factor. LEDC frequency = Fsys / R8_LED_CLOCK_DIV.	10h

led Status post Register (R8_LED_STATUS)

Bit	name	access	description	Reset value
7	RB_LED_IF_DMA_END	RW1	DMA complete flag bit, write 1 to clear or write R16_LED_DMA_CNT to clear: 1: Completed; 0: Not completed.	0
6	RB_LED_FIFO_EMPTY	RO	FIFO empty status bit: 1: FIFO is empty; 0: There is data in the FIFO.	1
5	RB_LED_IF_FIFO	RW1	FIFO counting over half interrupt flag bit, write 1 to clear or write R16_LED_FIFO to clear: 1: FIFO count<=2; 0: FIFO count>2.	0
4	RB_LED_CLOCK	O	Current LED clock signal level status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved.	0
[2:0]	RB_LED_FIFO_COUNT	RO	The byte count value in the current FIFO must be an even number.	000b

LED data FIFO Register (R16_LED_FIFO)

Bit	name	access	description	Reset value
[15:0]	R16_LED_FIFO	WO	LED data FIFO entry, 16-bit write.	XXXXb

LED DMA remaining Counting register (R16_LED_DMA_CNT)

Bit	name	access	description	Reset value
[15:0]	R16_LED_DMA_CNT	RW	LED_DMA_MAIN The current DMA remaining words (16 bits) count in the main buffer area, which will automatically decrement after DMA is started. Only the lower 12 bits are valid. Does not include auxiliary buffers.	0000b

LED main buffer DMA address (R16_LED_DMA_MAIN)

Bit	name	access	description	Reset value
[15:0]	R16_LED_DMA_MAIN	RW	Main buffer DMA start address/current address, preset After the initial value, it is automatically incremented and must be aligned with 2 bytes.	XXXXb

LED auxiliary buffer DMA address (R16_LED_DM A_AUX)

Bit	name	access	description	Reset value
[15:0]	R16_LED_DMA_AUX	RW	Auxiliary buffer DMA start address/current address, pre After the initial value is set, it will be automatically incremented and must be aligned with 2 bytes.	XXXXb

12.3 LED configuration

(1) Set R8_LED_CLOCK_DIV to select the LED output clock frequency;

(2). Set R16_LED_DMA_MAIN to point to the buffer that is ready to output data, that is, the main buffer;

(3) If you choose LED channel mode 3, then you must also set R16_LED_DMA_AUX to point to the auxiliary buffer; (4) Set R8_LED_CTRL_MOD, select the channel mode, output polarity, bit sequence, enable interrupt and DMA functions, etc.; (5), Set LEDC and the necessary LED0 ～LED3 pin direction as output, optionally, set corresponding I/O drive capability; (6) Set DMA count register R16_LED_DMA_CNT, start DMA transmission, or send data by writing to FIFO.

Chapter 13 Segment LCD

13.1 Introduction to Segment LCD Control

The chip provides a segment LCD controller interface, which can support 96 dots (24×4) LCD panels. Support adjustable drive voltage, adjustable scan frequency, support 1/2 duty, 1/3 duty, 1/4 duty and 1/3 bias, 1/2 bias LCD.

13.2 Register description

Table 13-1 List of LCD related registers

name	address	description	Reset value
R8_LCD_CTRL_MOD	0x40006000	LCD mode configuration register	0x00
R32_LCD_RAM0	0x40006004	LCD data register group 0	0x00000000
R32_LCD_RAM1	0x40006008	LCD data register group 1	0x00000000
R32_LCD_RAM2	0x4000600C	LCD data register group 2	0x00000000

LCD Mode configuration Set Register (R8_LCD_CTRL_MO D)

Bit	name	access	description	Reset value
7	RB_LCD_V_SEL	WO	LCD drive voltage selection: 1: VIO33 * 76% (2.5V); 0: VIO33 * 100% (3.3V). LCD scan	0
[6:5]	RB_LCD_SCAN_CLK	RW	clock frequency selection: 00: 256Hz; 01: 512Hz; 10: 1KHz; 11: 128 Hz.	00b

[4:3]	RB_LCD_DUTY	RW	LCD driver duty selection:
			00: 1/2 duty (COM0-COM1);
			01: 1/3 duty (COM0-COM2);
			10: 1/4 duty (COM0-COM3).
				00b

2	RB_LCD_BIAS	RW	LCD driver bias selection bit: 1: 1/3 bias; 0: 1/2 bias.	0
1	RB_LCD_POWER_ON	RW	LCD analog circuit enable bit: 1: enable; 0: Closed.	0
0	RB_SYS_POWER_ON	RW	LCD logic circuit enable bit: 1: enable; 0: Closed.	0

LCD Data Send Memory bank 0 (R32_LCD_RAM0)

Bit	name	access	description	Reset value
[31:0]	R32_LCD_RAM0	RW	SEG0-SEG7 segment data, each segment has 4 bits.	00000000h

LCD Data Send Memory bank 1 (R32_LCD_RAM1)

Bit	name	access	description	Reset value
[31:0]	R32_LCD_RAM1	RW	SEG8-SEG15 segment data, each segment has 4 bits.	00000000h

LCD Data Send Memory bank 2 (R32_LCD_RAM2)

Bit	name	access	description	Reset value
[31:0]	R32_LCD_RAM2	RW	SEG16-SEG23 segment data, each segment has 4 bits.	00000000h

13.3 Segment LCD configuration

(1). Select and turn on the 32KHz clock source;

(2). Set the LCD drive pin to be used as a floating input. It is optional. Set R16_PIN_ANALOG_IE to save system power consumption. Note that the external reset RST# multiplexes the LCD drive pin SEG23. If SEG23 is to be used, then To cancel the external reset function; (3) Load the segment code data to be displayed to the LCD data register R32_LCD_RAM0/1/2;

(4) Configure R8_LCD_CTRL_MOD, set the drive voltage, scan frequency, bias, duty and other parameters, and set RB_LCD_POWER_ON and RB_SYS_POWER_ON to turn on the segment LCD driver;

(5) The data in the LCD data register can be updated at any time, and the display content can be changed.

Chapter 14 Passive Parallel Port

14.1 Introduction to Passive Parallel Port

The chip provides a passive parallel port, including 8 bidirectional data lines PB7 ～PB0, 4 input control lines SLVS/SLVW/SLVR/SLVA and 1 output interrupt request line SLVI. Among them, SLVS is chip select, low level is active, SLVW/SLVR control write and read respectively, SLVA=0 means the operation is data; SLVA=1 means the operation is command or status.

Table 14-1 Passive parallel port I/O operation table

SLVS	SLVW	SLVR	SLVA	PB7 ～**PB0**	operating
1	X	X	X	Three-state prohibition	Not selected, no action
0	1	1	X	Three-state prohibition	Although selected, but no operation
0	0	1/X	1	Input state	Write command to CH579, command exists R8_SLV_WR_DATA to write data
0	0	1/X	0	Input state	to CH579, data exists R8_SLV_WR_DATA read status from CH579, status
0	1	0	1	Output state	comes from R8_SLV_RD_STAT read data from CH579, data comes from
0	1	0	0	Output state	R8_SLV_RD_DATA

The register R8_SLV_WR_DATA is the multiplexing of the high byte of R16_PB_INT_MODE. The registers R8_SLV_RD_STAT and R8_SLV_RD_DATA are the multiplexing of the low byte of R16_PB_INT_MODE and R8_PB_OUT_0. Therefore, after the passive parallel port is enabled, the function of the module whose registers are multiplexed cannot be used.

The interrupt request output SLVI is the "logical OR" of R8_SLV_RD_STAT[7] and R8_PA_OUT_0[3]. When either of them is 1, the default high level is output, and both of them are 0 to output the low-level active interrupt request.

14.2 Register description

Table 14-2 Passive parallel port related register list

name	address	description	Reset value
R8_SLV_CONFIG	0x4000101C	Passive parallel port configuration register	0x00
R8_SLV_RD_DATA	0x400010C8	Passive parallel port read data register	0x00
R8_SLV_RD_STAT	0x40001096	Passive parallel port readout status register	0x00
R8_SLV_WR_DATA	0x40001097	Passive parallel port write data/command register	0x00

Be Parallel port Set Register (R8_SLV_CONFIG)

Bit	name	access	description	Reset value
7	RB_IF_SLV_RD	RW1	Data is read interrupt flag bit, write 1 to clear 0: 1: detected event being read; 0: no event.	0
6	RB_IF_SLV_WR	RW1	Data is written into the interrupt flag bit, write 1 to clear 0: 1: detected event being written; 0: no event.	0
5	RB_IF_SLV_CMD	R0	The command is written into the flag; 1: What is currently being written is a command; 0: There is no write currently or data is written.	0
4	Reserved	RO	Reserved.	0
3	RB_SLV_IE_RD	RW	Data read interrupt enable bit: 1: enable; 0: Closed.	0
2	RB_SLV_IE_WR	RW	Data write interrupt enable bit: 1: enable; 0: Closed.	0
1	RB_SLV_IE_CMD	RW	Command write interrupt enable bit: 1: enable; 0: Closed.	0
0	RB_SLV_ENABLE	RW	Passive parallel port enable bit:	0

			1: enable; 0: Closed.	

14.3 Passive Parallel Port Configuration

(1) Optionally, if you want to enable the SLVI interrupt output, you must first preset the status value in R8_SLV_RD_STAT, especially R8_SLV_RD_STAT[7] related to the SLVI interrupt pin, which should be set to 1, and R8_PA_OUT_0[3];
(2) Set the GPIO pins corresponding to SLVA (PA0), SLVS (PA1), SLVR (PB8), SLVW (PB9) to pull-up input (recommended) or floating input mode, and set PB7 ～PB0 pins to be up Pull input (recommended) or floating input mode;
(3). Set RB_SLV_ENABLE of R8_SLV_CONFIG to enable passive parallel port;
(4) Optionally, if you want to enable SLVI interrupt output, set the corresponding GPIO pin of SLVA (PA3) as output; (5) Optionally, enable passive parallel port interrupt response as needed, combining RB_SLV_IE_RD/RB_SLV_IE_WR /RB_SLV_IE_CMD, and configure the corresponding channel enable of the core NVIC, it is recommended to enable only RB_SLV_IE_CMD;
(6). Optionally, preset the data to be sent in R8_SLV_RD_DATA.
(7). Optionally, if the SLVI interrupt output has been enabled and a low-level interrupt request needs to be output through SLVI, then the preset status value in R8_SLV_RD_STAT can be updated and R8_SLV_RD_STAT[7] is 0, if you need to cancel If the interrupt request restores SLVI to the default high level, you can set R8_SLV_RD_STAT[7] to 1, or set R8_PA_OUT_0[3] to 1; (8) After being interrupted by a write command or querying that RB_IF_SLV_WR is valid, when RB_IF_SLV_CMD is 1 When RB_SLV_WR_DATA gets the written command code, when RB_IF_SLV_CMD is 0, get the written data from R8_SLV_WR_DATA; (9) After querying that RB_IF_SLV_RD is valid, the next data or state can be preset.

Chapter 15 Analog-to-digital converter ADC

15.1 Introduction to ADC

The chip provides a 12-bit successive approximation analog-to-digital converter ADC, providing up to 16 channels, supporting 14 external signal sources and 2 internal signal sources.

15.1.1 Main Features

- 12-bit resolution.
- 14 external voltage sampling channels, internal temperature detection channels, internal battery voltage detection channels.
- Single-ended input mode and differential input mode detection.
- The sampling clock frequency is optional.
- ADC input voltage range is 0V ～VIO33.
- Optional PGA, provide gain adjustment options.
- Optional input buffer BUF to support high-impedance signal source.

15.1.2 Function description

The following figure is a block diagram of an ADC module.

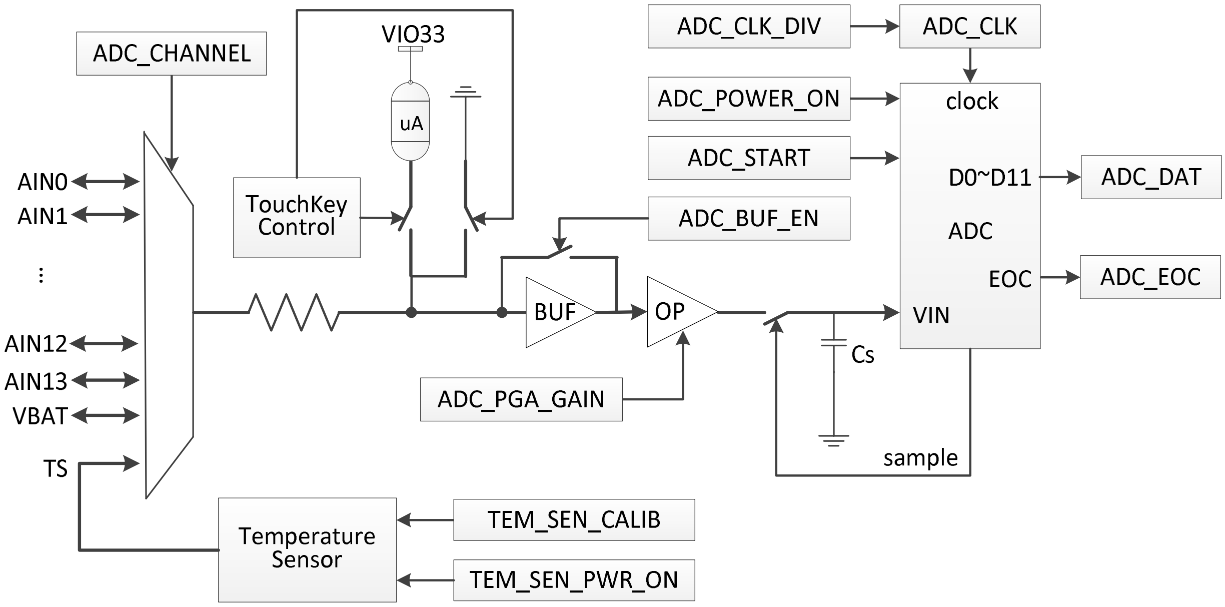


Figure 15-1 ADC structure diagram

15.2 Register description

Table 15-1 ADC related register list

name	address	description	Reset value
R8_ADC_CHANNEL	0x40001058	ADC input channel selection register	0x0F
R8_ADC_CFG	0x40001059	ADC configuration register	0xA0
R8_ADC_CONVERT	0x4000105A	ADC conversion control register	0x00
R8_TEM_SENSOR	0x4000105B	Temperature sensor control register	0x04
R16_ADC_DATA	0x4000105C	ADC data register	0x0XXX
R8_ADC_INT_FLAG	0x4000105E	ADC interrupt flag register	0x00

ADC Input pass Tao Select register (R8_ADC_CHANNEL)

Bit	name	access	description	Reset value
[7:4]	Reserved	RO	Reserved.	0000b

[3:0]	RB_ADC_CH_INX	RW	ADC channel index number, 16 channels in total:
			00h-0Dh: external signal channel AIN0 ～AIN13;
			0Eh: battery voltage VBAT;
			0Fh: Built-in temperature sensor TS.
				1111b

ADC Configuration send Register (R8_ADC_CFG)

Bit	name	access	description	Reset value
[7:6]	RB_ADC_CLK_DIV	RW	ADC clock frequency selection: 00: Divide by 10 based on CK32M, 3.2MHz; 01: Divide by 12 based on CK32M, 2.67MHz; 10: Divide by 6 based on CK32M, 5.33MHz; 11: Divide by 8, based on CK32M, 4MHz. The actual sampling rate is approximately 1/16 of the clock frequency.	10b

[5:4]	RB_ADC_PGA_GAIN	RW	ADC input PGA gain selection: 00: -12dB, 1/4 times;
			01: -6dB, 1/2 times;
			10: 0dB, 1 times, no gain;
			11: 6dB, 2 times.
				10b

3	RB_ADC_OFS_TEST	RW	ADC offset error test mode: 1: Test mode (only supported by channel 1), test mode Under the formula, the lower 12-bit data of the data register R16_ADC_DATA will be inverted by bit (0x0579 is inverted to 0x0A86); 0: Normal mode.	0
2	RB_ADC_DIFF_EN	RW	ADC channel signal input mode: 1: Differential input; 0: Single-ended input.	0
1	RB_ADC_BUF_EN	RW	ADC input buffer BUF enable: 1: open; 0: Closed.	0
0	RB_ADC_POWER_ON	RW	ADC module power enable control: 1: enable; 0: Closed.	0

Table 15-2 PGA Gain selection And input voltage range table

PGA gain selection	Data converted by ADC Calculate the sampled voltage Vi	Theoretically measurable Voltage upper limit	Theoretical measurable voltage range (Assuming Vref=1.05V)	Suggest actually available Measuring voltage range
-12dB (1/4 times)	(ADC/512-3)*Vref	5*Vref	-0.2V ～ VIO33+0.2V	2.9V ～ VIO33
-6dB (1/2 times)	(ADC/1024-1)*Vref	3*Vref	-0.2V ～ 3.15V	1.9V ～ 3V
0dB (1 times)	(ADC/2048)*Vref	2*Vref	0V ～ 2.1V	0V ～ 2V
6dB (2 times)	(ADC/4096+0.5)*Vref	1.5*Vref	0.525V ～ 1.575V	0.6V ～ 1.5V

ADC: The digital quantity after ADC conversion, namely R16_ADC_DATA.

Vref: The actual voltage value of the power node VINTA of the internal analog circuit, usually 1.05V±0.015V. Note: If the lower voltage is sampled after negative gain (signal reduction), it may cause a large error within a certain voltage range;

However, if a higher voltage is sampled after positive gain (signal amplification), it may cause the ADC conversion value to overflow, so it is recommended to

Select a reasonable gain mode to measure the signal voltage range.

By default, it is recommended to turn on the input buffer. Only when the internal resistance of the external signal source is less than 1KΩ, the input buffer can be turned off for ADC

When using differential input, it is recommended to turn off the input buffer. When used for TouchKey detection, the input buffer must be turned on, and

It is recommended to select one of two gains, 0dB (priority) or -6dB.

ADC Conversion control system Register (R8_ADC_CONVERT)

Bit	name	access	description	Reset value
7	RB_ADC_EOC_X	RO	ADC conversion end flag (asynchronous signal): 1: Complete; 0: In progress.	0
6	Reserved	RO	Reserved.	0
5	RB_TKEY_CHG_ACT	RO	TouchKey capacitor charging status: 1: Capacitor is being charged; 0: Not being charged.	0
4	RB_TKEY_ACTION	RO	TouchKey operation status: 1: Charging or discharging or ADC conversion; 0: Idle state.	0
3	RB_TKEY_PWR_ON	RW	TouchKey module power enable control: 1: enable; 0: Closed.	0
[2:1]	Reserved	RO	Reserved.	00b
0	RB_ADC_START	RW	ADC conversion start control and status, automatically cleared: 1: Start conversion/conversion; 0: Stop conversion.	0

temperature Degree sensor control Control register (R8_TEM_SENS OR)

Bit	name	access	description	Reset value
7	RB_TEM_SEN_PWR_ON	RW	TS temperature sensor power supply enable control: 1: enable; 0: Closed.	0
[6:3]	Reserved	RW	Reserved.	0000b
[2:0]	RB_TEM_SEN_CALIB	RW	The calibration value of the TS temperature sensor must remain unchanged.	100b

ADC Data Deposit (R16_ADC_DATA)

Bit	name	access	description	Reset value
[15:12]	Reserved	RW	Reserved.	0000b
[11:0]	RB_ADC_DATA	RO	ADC converted data.	XXXh

ADC Interrupt flag send Register (R8_ADC_INT_FLA G)

Bit	name	access	description	Reset value
7	RB_ADC_IF_EOC	RO	ADC conversion complete interrupt flag. Write to register R8_ADC_CONVERT This flag can be cleared.	0
[6:0]	Reserved	RW	Reserved.	0000000b

15.3 ADC configuration

15.3.1 External channel sampling

(1) Set RB_ADC_POWER_ON of R8_ADC_CFG to 1 to turn on the ADC, select the sampling frequency through RB_ADC_CLK_DIV, and enable the input buffer and select signal gain through RB_ADC_BUF_EN and RB_ADC_PGA_GAIN;

(2) Set the R8_ADC_CHANNEL register to select the external or internal signal channel;

(3) Set the R8_ADC_CONVERT register, set RB_ADC_START, and start ADC conversion;

(4). Query and wait for RB_ADC_START to be automatically cleared or wait for RB_ADC_IF_EOC to be set to 1, indicating that the conversion is complete. You can read R16_ADC_DATA to obtain the 12-bit ADC conversion data. If time is sufficient, it is recommended to convert again and discard the first ADC data;

(5), repeat 2, Steps 3 and 4, you can continue to sample another channel or the next set of data.

(6) Single ADC conversion cycle: ADC sampling (4 clocks) + conversion time (12 clocks) ≈ 16 Tadc, continuous ADC

1 time interval should be added when time, where Tadc = Tsys @ RB_ADC_CLK_DIV.

(7) If using differential input:

Enable differential, select 0# channel: actually perform differential conversion on the voltage of AIN0 (positive end) and AIN2 (negative end);

enable differential, select 1# channel: actually perform differential conversion on AIN1 (positive end) and AIN3 (negative end) Terminal) voltage for differential conversion;

The result of ADC conversion, if the data is greater than 0x800, it means that the voltage of the differential positive terminal is higher than that of the differential negative terminal; if the data is less than 0x800, it means the voltage of the differential positive terminal is lower than the voltage of the differential negative terminal. Taking PGA gain selection 0dB as an example, the theoretically measurable voltage range is -1.05V ～1.05V, 0x400 means that the voltage at the pos

15.3.2 Temperature sensor sampling

(1) Set the RB_TEM_SEN_PWR_ON of the R8_TEM_SENSOR register to 1 to turn on the temperature sensor, set R8_ADC_CHANNEL to 15, and select the temperature sensor signal to connect to the ADC input;

(2) Set RB_ADC_POWER_ON to 1 to turn on the ADC, clear RB_ADC_DIFF_EN, set RB_ADC_CLK_DIV, set RB_ADC_BUF_EN to 1, set RB_ADC_PGA_GAIN to 0dB gain;

(3) Set R8_ADC_CONVERT register, set RB_ADC_START to 1, start ADC conversion;

(4) Query and wait for RB_ADC_START to be automatically cleared or wait for RB_ADC_IF_EOC to be set to 1, read R16_ADC_DATA to obtain 12-bit ADC conversion data. When the accuracy is high, it is recommended to repeat steps 3 and 4 several times to calculate the average value of ADC data; (5))、According to the conversion relationship between voltage and temperature, the temperature value is obtained. For details, please refer to the evaluation board example program.

Chapter 16 Touch-Key

16.1 Introduction to Touch-Key

The chip provides a capacitance detection module, which can be used with the ADC module to realize capacitance touch button detection. There are a total of 14 input channels, and the capacitance value range of touch buttons is 10pF ～100pF.

16.2 Register description

Table 16-1 TouchKey related register list

name	address	description	Reset value
R8_TKEY_CTRL	0x4000105A	TouchKey control register	0x00
R8_TKEY_CNT	0x4000105F	TouchKey charge and discharge time configuration register	0x00

Register R8_TKEY_CTRL is an alias of R8_ADC_CONVERT.

TouchKey control Register (R8_TKEY_CTRL)

Bit	name	access	description	Reset value
7	RB_ADC_EOC_X	RO	ADC conversion end flag. 1: Complete; 0: In progress.	0
6	Reserved	RO	Reserved.	0
5	RB_TKEY_CHG_ACT	RO	TouchKey capacitor charging status: 1: Capacitor is being charged; 0: Not being charged.	0
4	RB_TKEY_ACTION	RO	TouchKey operation status: 1: Charging or discharging or ADC conversion; 0: Idle state.	0
3	RB_TKEY_PWR_ON	RW	TouchKey module power enable control: 1: enable; 0: Closed.	0
[2:1]	Reserved	RO	Reserved.	00b
0	RB_ADC_START	RW	ADC conversion start control and status, automatically cleared: 1: Start conversion/conversion; 0: Stop conversion.	0

TouchKey charging and discharging Power time configuration register (R8_TKEY_C NT)

Bit	name	access	description	Reset value
[7:0]	R8_TKEY_CNT	W0	The number of Touch-Key charge and discharge cycles (in Tsys). The full 8 bits are the number of charge cycles. It is recommended not to be less than 0x10. The number of discharge cycles is automatically generated based on the number of charge cycles. When the number of charge cycles is less than 128, the number of discharge cycles When the number of charge cycles is greater than or equal to 128, the number of discharge cycles is 14.	00h

16.3 Touch-Key configuration

(1) Set RB_ADC_POWER_ON to 1 to turn on the ADC, clear RB_ADC_DIFF_EN, set RB_ADC_CLK_DIV, set RB_ADC_BUF_EN to 1, set RB_ADC_PGA_GAIN to 0dB or -6dB gain, the latter is used for smaller capacitors; (2) Set R8_TKEY_CTRL's RB_T1_CTRL_ON to KEY_T1_KEY To enable the TouchKey module;

(3) Set R8_ADC_CHANNEL to select the channel where the target button is located;

(4). Optionally, if you use the interrupt mode, you must write R8_TKEY_CTRL to clear RB_ADC_IF_EOC, the same as step 2; (5) Estimate the target button capacitance, calculate its charging time, write the number of charging cycles into the R8_TKEY_CNT register, and calculate :

The number of charging cycles R8_TKEY_CNT = (Ckey + 5pF parasitic) * Vkey / Ikey / Tsys,

(Among them, Vkey is recommended to be 1.6V at 0dB gain, Vkey is recommended to be 2.4V at -6dB gain, and Ikey is about 35uA) Assuming the external capacitance of the touch button Ckey=35pF, Fsys=16MHz, then R8_TKEY_CNT=29,

Use 16 when the calculation result is less than 16, and use 128 when it is between 110 and 128;

(6). Query and wait for RB_TKEY_ACTION to be 0 (or, if you have cleared it before, you can wait for RB_ADC_IF_EOC to be set to 1), read R16_ADC_DATA to obtain 12-bit ADC conversion data, compare it with the history value of the same channel, analyze and determine whether the button is pressed , It is recommended to use software analysis to remove interference;

(7) Repeat steps 3, 4, 5, and 6 to continue to detect the buttons of the next channel.

(8) If the interrupt mode is used, the following 4 steps will be repeated in the interrupt service routine: 6 read data and judge, 4 or 2 clear interrupt flag, 3 select channel, 5 start detection.

Chapter 17 USB Controller

17.1 Introduction to USB Controller

The chip is embedded with a USB master-slave controller and transceiver, the characteristics are as follows:

l Support USB Host host function and USB Device device function.

l Support USB2.0 full speed 12Mbps or low speed 1.5Mbps.

l Support USB control transmission, batch transmission, interrupt transmission, synchronous/real-time transmission.

l Supports data packets up to 64 bytes, built-in FIFO, support interrupt and DMA.

17.2 Register description

USB related registers are divided into 3 parts, some of which are multiplexed in host and device modes.

(1), USB global register

(2) USB device control register

(3) USB host control register

17.2.1 Global Register Description

Table 17-1 USB List of related registers (The standard gray is controlled by RB_UC_RESET_SIE reset)

name	address	description	Reset value
R8_USB_TYPE_C_CTRL	0x40008038	USB type-C configuration register	0x00
R8_USB_CTRL	0x40008000	USB control register	0x06
R8_USB_INT_EN	0x40008002	USB interrupt enable register	0x00
R8_USB_DEV_AD	0x40008003	USB device address register	0x00
R32_USB_STATUS	0x40008004	USB status register	0xXX20XXXX
R8_USB_MIS_ST	0x40008005	USB Miscellaneous Status Register	0xXX
R8_USB_INT_FG	0x40008006	USB interrupt flag register	0x20
R8_USB_INT_ST	0x40008007	USB interrupt status register	0xXX
R8_USB_RX_LEN	0x40008008	USB receive length register	0xXX

USB type-C Match Set the register (R8_USB_ T YPE_C_C TRL)

Bit	name	access	description	Reset value
7	RB_UTCC_GP_BIT	RW	USB general flag bit, user-defined.	0
6	RB_UCC2_PD_EN	RW	The internal 5.1K pull-down resistor of UCC2 pin: 1: enable; 0: Prohibited.	0
[5:4]	RB_UCC2_PU_EN	RW	UCC2 pin internal pull-up resistor control bit: 00: Disable internal pull-up resistor; 01: Turn on the internal 36K pull-up and provide the default USB current; 10: Turn on the internal 12K pull-up and provide 1.5A current; 11: Turn on the internal 4.7K pull-up and provide 3A current. The	00b
3	RB_VBUS_PD_EN	RW	internal 10K pull-down resistor of VBUS pin: 1: enable; 0: Prohibited.	0
2	RB_UCC1_PD_EN	RW	The internal 5.1K pull-down resistor of UCC1 pin: 1: enable; 0: Prohibited.	0
[1:0]	RB_UCC1_PU_EN	RW	UCC1 pin internal pull-up resistor control bit: 00: Disable internal pull-up resistor;	00b

			01: Turn on the internal 36K pull-up and provide the default USB current; 10: Turn on the internal 12K pull-up and provide 1.5A current; 11: Turn on the internal 4.7K pull-up and provide 3A current.	

The above USB type-C pull-up resistance and pull-down resistance are independent of the port pull-up resistance controlled by the GPIO port direction control and pull-up resistance enable register. When a pin is used for USB type-C, the corresponding pin should be disabled It is recommended to enable the high-impedance input mode for this pin (to avoid the pin outputting low or high level) .

USB Control post Memory (R8_USB_CTRL)

Bit	name	access	description	Reset value
7	RB_UC_HOST_MODE	RW	USB working mode selection bit: 1: Host mode (HOST) ; 0: Device mode (DEVICE).	0
6	RB_UC_LOW_SPEED	RW	USB bus signal transmission rate selection bit: 1: 1.5Mbps; 0: 12Mbps.	0
5	RB_UC_DEV_PU_EN	RW	In the USB device mode, the USB device is enabled and internally The pull-up resistor control bit, set to 1, enables USB device transmission and enables the internal pull-up resistor.	0
[5:4]	MASK_UC_SYS_CTRL	RW	See the table below to configure the USB system.	00b
3	RB_UC_INT_BUSY	RW	The USB transfer completes the interrupt flag before it is cleared. Stop enable bit: 1: Before the interrupt flag UIF_TRANSFER is not cleared Automatically pause, automatically answer busy NAK in device mode, Automatically suspend subsequent transmissions in host mode; 0: Do not pause.	0
2	RB_UC_RESET_SIE	RW	USB protocol processor software reset control bit: 1: Force reset of the USB protocol processor (SIE), required To be cleared by software; 0: Do not reset.	1
1	RB_UC_CLR_ALL	RW	The USB FIFO and interrupt flag are cleared: 1: Forced clearing and clearing; 0: Uncleared.	1
0	RB_UC_DMA_EN	RW	USB DMA and DMA interrupt control bits: 1: Enable DMA function and DMA interrupt; 0: Disable DMA.	0

by RB_UC_HOST_MODE with MASK_UC_SYS_CTRL Form a USB system control combination:

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB system control description
0	00	Disable the USB device function and turn off the internal pull-up resistor.
0	01	To enable the USB device function and turn off the internal pull-up resistor, an external pull-up is required.
0	1x	Enable the USB device function and enable the internal 1.5K pull-up resistor. The pull up The resistor has priority over the pull-down resistor and can also be used in GPIO mode.
1	00	USB host mode, normal working state.
1	01	In USB host mode, DP/DM is forced to output SE0 status.
1	10	In USB host mode, DP/DM is forced to output J state.
1	11	In USB host mode, DP/DM is forced to output K state/wake-up.

USB Interrupt enable can Register (R8_USB_INT_EN)

Bit	name	access	description	Reset value

7	RB_UIE_DEV_SOF	RW	USB device mode, receiving SOF packet interrupt: 1: enable interrupt; 0: Disable interrupts.	0
6	RB_UIE_DEV_NAK	RW	In USB device mode, NAK interrupt is received: 1: Enable interrupt; 0: Disable interrupts.	0
5	Reserved	RO	Reserved.	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Disable interrupts.	0
3	RB_UIE_HST_SOF	RW	USB host mode, SOF timer interrupt: 1: Enable interrupt; 0: Disable interrupts.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wake-up event interrupt: 1: Enable interrupt; 0: Disable interrupts.	0
1	RB_UIE_TRANSFER	RW	USB transfer completed interrupted: 1: Enable interrupt; 0: Disable interrupts.	0
0	RB_UIE_DETECT	RW	USB host mode, USB device connection or disconnection event Interrupt: 1: Enable interrupt; 0: Disable interrupts.	0
	RB_UIE_BUS_RST	RW	USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupts.	0

USB Equipment location site Register (R8_USB_DEV_AD)

Bit	name	access	description	Reset value
7	RB_UDA_GP_BIT	RW	USB general flag bit, user-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: the address of the currently operating USB device; Device Mode: The USB's own address.	0000000b

USB Miscellaneous state Register (R8_USB_MIS_ST)

Bit	name	access	description	Reset value
7	RB_UMS_SOF_PRES	RO	In USB host mode, the SOF packet indicates the status bit: 1: SOF packet is about to be sent, at this time, if there are other USB data packets, it will be automatically delayed; 0: No SOF packet is sent.	X
6	RB_UMS_SOF_ACT	RO	SOF packet transmission status bit in USB host mode: 1: SOF packet is being sent out; 0: Sending completed or idle.	X
5	RB_UMS_SIE_FREE	RO	Idle status bits of the USB protocol processor: 1: The protocol device is idle; 0: Busy, USB transfer is in progress.	1
4	RB_UMS_R_FIFO_RDY	RO	USB receive FIFO data ready status bit: 1: receive FIFO is not empty; 0: The receive FIFO is empty.	0
3	RB_UMS_BUS_RESET	RO	USB bus reset status bit: 1: The current USB bus is in the reset state; 0: The current USB bus is in a non-reset state.	X
2	RB_UMS_SUSPEND	RO	USB suspend status bit: 1: The USB bus is in a suspended state and has not been Have USB activity; 0: The USB bus is in non-suspended state.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, the device has just been connected to the USB port	0

			It is the level state of the DM pin, used to judge the speed: 1: High level/low speed; 0: Low level/full speed.	
0	RB_UMS_DEV_ATTACH	RO	USB device connection status bits of the port in USB host mode: 1: The port has been connected to a USB device; 0: No USB device is connected to the port.	0

USB Interrupt mark Zhi Register (R8_USB_INT_FG)

Bit	name	access	description	Reset value
7	RB_U_IS_NAK	RO	In USB device mode, NAK response status bit: 1: Response to NAK during the current USB transmission; 0: No NAK response.	0
6	RB_U_TOG_OK	RO	Current USB transmission DATA0/1 synchronization flag matching status bit: 1: Synchronization; 0: Not synchronized.	0
5	RB_U_SIE_FREE	RO	USB protocol processor idle status bit: 1: USB is idle; 0: Busy, USB transfer is in progress.	1
4	RB_UIF_FIFO_OV	RW	USB FIFO overflow interrupt flag bit, write 1 to clear it: 1: FIFO overflow trigger; 0: no event.	0
3	RB_UIF_HST_SOF	RW	SOF timer interrupt flag bit in USB host mode, write 1 to clear: 1: SOF packet transmission complete trigger; 0: no event.	0
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag bit, Write 1 to clear: 1: USB suspend event or wake-up event triggered; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transfer complete interrupt flag bit, write 1 to clear it: 1: a USB transfer complete trigger; 0: no event.	0
0	RB_UIF_DETECT	RW	In the USB host mode, the USB device connection or disconnection event interrupt flag bit, write 1 to clear: 1: USB device connection or disconnection trigger is detected; 0: No event.	0
	RB_UIF_BUS_RST	RW	In the USB device mode, the USB bus reset event interrupt flag bit, write 1 to clear: 1: USB bus reset event triggered; 0: No event.	0

USB Interrupted state Register (R8_USB_INT_ST)

Bit	name	access	description	Reset value
7	RB_UIS_IS_NAK	RO	In USB device mode, NAK response status bit is the same as RB_U_IS_NAK: 1: Response to NAK during the current USB transmission; 0: No NAK response.	0
6	RB_UIS_TOG_OK	RO	Current USB transmission DATA0/1 synchronization flag matching status bit, same as RB_U_TOG_OK: 1: Synchronization; 0: Not synchronized.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token of the current USB transfer transaction	XXb

			PID identification.	
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint of the current USB transfer transaction number.	XXXXb
	MASK_UIS_H_RES	RO	In host mode, the response of the current USB transfer transaction PID identification, 0000 means the device has no response or timeout; other values of XXXXb means the response PID.	

MASK_UIS_TOKEN is used to identify the current USB transfer transaction token PID in USB device mode: 00 means OUT packet; 01 means SOF packet; 10 means IN packet; 11 means SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In host mode, if the host sends an OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID is the PID of the data packet (DATA0/DATA1) or the handshake packet PID.

USB Receiver degree Register (R8_USB_RX_LEN)

Bit	name	access	description	Reset value
[7:0]	R8_USB_RX_LEN	RO	The number of data bytes currently received by the USB endpoint.	XXh

17.2.2 Device register description

In the USB device mode, the chip provides 5 groups of bidirectional endpoints 0, 1, 2, 3, 4, and the maximum data packet length of all endpoints is 64 bytes.

Endpoint 0 is the default endpoint and supports control transmission. Sending and receiving share a 64-byte data buffer.
Endpoint 1, endpoint 2, and endpoint 3 each include a sending endpoint IN and a receiving endpoint OUT, each with an independent 64-byte or double 64-byte data buffer for sending and receiving, and supports batch transmission, interrupt transmission, and real-time/synchronization transmission.

Endpoint 4 includes a sending endpoint IN and a receiving endpoint OUT, each with an independent 64-byte data buffer for sending and receiving, which supports batch transmission, interrupt transmission and real-time/synchronous transmission.

Each group of endpoints has a control register R8_UEPn_CTRL and a send length register R8_UEPn_T_LEN (n=0/1/2/3/4), which are used to set the synchronization trigger bit of the endpoint, respond to OUT transactions and IN transactions, and send data The length and so on.

The USB bus pull-up resistor necessary as a USB device can be set by software at any time. When the RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller will be the DP/DM pin of the USB bus according to the speed setting of RB_UD_LOW_SPEED. Connect the pull-up resistor and enable the USB device function.

When the USB bus reset, USB bus hang or wake-up event is detected, or when the USB successfully processes the data transmission or data reception, the USB protocol processor will set the corresponding interrupt flag. If the interrupt enable is turned on, it will also generate a corresponding interrupt flag. Interrupt request. should The program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND; and, if RB_UIF_TRANSFER is valid, then continue to analyze the USB interrupt status register R8_USB_INT_ST, according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID mark MASK_UIS_TOKEN perform corresponding processing. If the synchronization trigger bit RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, you can use RB_U_TOG_OK or RB_UIS_TOG_OK to determine whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint. If the data is synchronized, the data Valid; if the data is out of sync, the data should
thrown away. Every time the USB transmission or reception interrupt is processed, the synchronization trigger bit of the corresponding endpoint should be modified correctly for the next t
Whether the sent data packet or the data packet received next time is synchronously detected; in addition, setting RB_UEP_AUTO_TOG can be realized in the sending After successful or successful reception, the corresponding synchronization trigger bit is automatically flipped.

The data to be sent by each endpoint is in their own buffer, and the length of the data to be sent is independently set in R8_UEPn_T_LEN Medium; the data received by each endpoint is in their own buffer, but the length of the received data is in the USB receive length register In R8_USB_RX_LEN, it can be distinguished according to the current endpoint number when the USB receives an interrupt.

Table 17-2 USB settings Prepare related register list Table (marked gray is controlled by RB_UC_RESET_SIE reset)

name	address	description	Reset value
R8_UDEV_CTRL	0x40008001	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x4000800c	Endpoint 1/4 mode control register	0x00

R8_UEP2_3_MOD	0x4000800d	Endpoint 2/3 mode control register	0x00
R16_UEP0_DMA	0x40008010	Endpoint 0 buffer start address	0xXXXX
R16_UEP1_DMA	0x40008014	Endpoint 1 buffer start address	0xXXXX
R16_UEP2_DMA	0x40008018	Endpoint 2 buffer start address	0xXXXX
R16_UEP3_DMA	0x4000801c	Endpoint 3 buffer start address	0xXXXX
R8_UEP0_T_LEN	0x40008020	Endpoint 0 send length register	0xXX
R8_UEP0_CTRL	0x40008022	Endpoint 0 control register	0x00
R8_UEP1_T_LEN	0x40008024	Endpoint 1 send length register	0xXX
R8_UEP1_CTRL	0x40008026	Endpoint 1 control register	0x00
R8_UEP2_T_LEN	0x40008028	Endpoint 2 send length register	0xXX
R8_UEP2_CTRL	0x4000802a	Endpoint 2 control register	0x00
R8_UEP3_T_LEN	0x4000802c	Endpoint 3 send length register	0xXX
R8_UEP3_CTRL	0x4000802e	Endpoint 3 control register	0x00
R8_UEP4_T_LEN	0x40008030	Endpoint 4 send length register	0xXX
R8_UEP4_CTRL	0x40008032	Endpoint 4 control register	0x00

USB Equipment Reason Port control register (R8_UDEV_CTRL)

Bit	name	access	description	Reset value
7	RB_UD_PD_DIS	RW	USB device port UD+/UD- pin internal pull-down Resistance control bit: 1: Disable internal pull-down; 0: Enable internal pull-down. Can be used in GPIO mode to provide pull-down resistors.	1
6	Reserved	RO	Reserved.	0
5	RB_UD_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	X
4	RB_UD_DM_PIN	RO	Current UD-pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved.	0
2	RB_UD_LOW_SPEED	RW	USB device physical port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full speed mode.	0
1	RB_UD_GP_BIT	RW	USB device mode general flag bit, user-defined.	0
0	RB_UD_PORT_EN	RW	USB device physical port enable bit: 1: Enable the physical port; 0: Disable the physical port.	0

End 1/4 mode Control register (R8_UEP4_1 _MOD)

Bit	name	access	description	Reset value
7	RB_UEP1_RX_EN	RW	1: Enable endpoint 1 to receive (OUT); 0: Disable endpoint 1 to receive.	0
6	RB_UEP1_TX_EN	RW	1: Enable endpoint 1 to send (IN) ; 0: Disable endpoint 1 sending.	0
5	Reserved	RO	Reserved.	0
4	RB_UEP1_BUF_MOD	RW	Endpoint 1 data buffer mode control bit.	0
3	RB_UEP4_RX_EN	RW	1: Enable endpoint 4 to receive (OUT); 0: Disable endpoint 4 to receive.	0
2	RB_UEP4_TX_EN	RW	1: Enable endpoint 4 to send (IN) ; 0: Disable endpoint 4 to send.	0
[1:0]	Reserved	RO	Reserved.	0

The combination of bUEP4_RX_EN and bUEP4_TX_EN configures the data buffer mode of USB endpoints 0 and 4. For details, refer to the following table:

Table 17-3 Endpoint 0 and 4 buffer mode

bUEP4_RX_EN	bUEP4_TX_EN	Description: Take UEP0_DMA as the starting address and arrange from low to high
0	0	Endpoint 0 single 64-byte transceiver shared buffer (IN and OUT) .
1	0	Endpoint 0 single 64-byte transceiver shared buffer; Endpoint 4 single 64-byte receive buffer (OUT).
0	1	Endpoint 0 single 64-byte transmit and receive shared buffer; Endpoint 4 single 64-byte transmit buffer (IN).
1	1	Endpoint 0 single 64-byte transceiver shared buffer; Endpoint 4 single 64-byte receive buffer (OUT); Endpoint 4 single 64-byte receive buffer (IN). A total of 192 bytes are arranged as follows: UEP0_DMA+0 address: endpoint 0 transmit and receive shared buffer 64-byte start address; UEP0_DMA+64 address: endpoint 4 receive buffer 64-byte start address; UEP0_DMA+128 address: endpoint 4 send The 64-byte start address of the buffer.

Endpoint 2/3 mode Control register (R8_UEP2_3 _MOD)

Bit	name	access	description	Reset value
7	RB_UEP3_RX_EN	RW	1: Enable endpoint 3 to receive (OUT); 0: Disable endpoint 3 to receive.	0
6	RB_UEP3_TX_EN	RW	1: Enable endpoint 3 to send (IN) ; 0: Disable endpoint 3 to send.	0
5	Reserved	RO	Reserved.	0
4	RB_UEP3_BUF_MOD	RW	Endpoint 3 data buffer mode control bit.	0
3	RB_UEP2_RX_EN	RW	1: Enable endpoint 2 to receive (OUT); 0: Disable endpoint 2 to receive.	0
2	RB_UEP2_TX_EN	RW	1: Enable endpoint 2 to send (IN) ; 0: Disable endpoint 2 sending.	0
1	Reserved	RO	Reserved.	0
0	RB_UEP2_BUF_MOD	RW	Endpoint 2 data buffer mode control bit.	0

By RB_UEPn_RX_EN and RB_UEPn_TX_EN and RB_UEPn_BUF_MOD (n=1/2/3) Combine to configure USB endpoint 1, respectively For the data buffer mode of 2, 3, refer to the table below for details. Among them, in the dual 64-byte buffer mode, USB data transmission will select the first 64-byte buffer according to RB_UEP_*_TOG=0, and the last 64-byte buffer according to RB_UEP_*_TOG=1. Set RB_UEP_AUTO_TOG=1. Realize automatic switching.

Table 17-4 Endpoint n buffer mode (n=1/2/3)

RB_UEPn_ RX_EN	RB_UEPn_ TX_EN	RB_UEPn_ BUF_MOD	Description: Take R16_UEPn_DMA as the starting address and arrange from low to high
0	0	X	The endpoint is disabled and the R16_UEPn_DMA buffer is not used.
1	0	0	Single 64-byte receive buffer (OUT).
1	0	1	Dual 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG. 0
0	1		Single 64-byte transmit buffer (N).
0	1	1	Dual 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG. 0
1	1		Single 64-byte receive buffer (OUT), single 64-byte transmit buffer (IN).
1	1	1	The dual 64-byte receive buffer (OUT) is selected by RB_UEP_R_TOG, and the dual 64-byte transmit buffer (IN) is selected by RB_UEP_T_TOG. All 256 bytes are arranged as follows: UEPn_DMA+0 address: the endpoint receiving address when RB_UEP_R_TOG=0; UEPn_DMA+64 address: the endpoint receiving address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint send address when RB_UEP_T_TOG=0; UEPn_DMA+192 address: when RB_UEP_T_TOG=1, the endpoint sends the address.

Endpoint n buffer Start address (R16_UEPn_DM A) (n=0/1/2/3)

Bit	name	access	description	Reset value
[15:0]	R16_UEPn_DMA	RW	Endpoint n buffer start address. The lower 15 bits are valid, and the address must be aligned with 4 bytes.	XXXXh

Note: The length of the buffer for receiving data >= min (the maximum packet length that may be received + 2 bytes, 64 bytes) .

Endpoint n send length Degree register (R8_UEPn_T_L EN) (n=0/1/2/3/4)

Bit	name	access	description	Reset value
[7:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes to be sent by the USB endpoint n.	XXh

Endpoint n control register (R8_UEPn_CTRL) (n =0/1/2 /3/4)

Bit	name	access	description	Reset value
7	RB_UEP_R_TOG	RW	Receiver of USB endpoint n (processing OUT transaction) Expected synchronization trigger bit: 1: Expect DATA1; 0: Expect DATA0.	0
6	RB_UEP_T_TOG	RW	The synchronization trigger bit prepared by the transmitter of USB endpoint n (processing IN transaction): 1: Send DATA1; 0: Send DATA0.	0
5	Reserved	RO	Reserved.	0
4	RB_UEP_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: After the data is sent or received successfully, it will be automatically turned over Sync trigger bit; 0: Do not turn over automatically, you can switch manually. Only endpoint 1/2/3 is supported.	0
[3:2]	MASK_UEP_R_RES	RW	The receiver of endpoint n responds to the OUT transaction control: 00: Reply ACK; 01: timeout/no response, used for real-time non-endpoint 0 /Synchronous transmission; 10: Answer NAK or busy; 11: Answer STALL or error.	00b
[1:0]	MASK_UEP_T_RES	RW	The sender of endpoint n responds to the control of IN transaction: 00: DATA0/DATA1 data is ready and expects ACK; 01: Reply to DATA0/DATA1 and expect no response, Used for real-time/synchronous transmission of non-endpoint 0; 10: Answer NAK or busy; 11: Answer STALL or error.	00b

17.2.3 Host register description

In USB host mode, the chip provides a set of two-way host endpoints, including a sending endpoint OUT and a receiving endpoint IN. The maximum length of a data packet is 64 bytes, and it supports control transmission, interrupt transmission, batch transmission and real-time/synchronous transmission. .

Every USB transaction initiated by the host endpoint will always automatically set the RB_UIF_TRANSFER interrupt flag after the end of the processing. The application program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to each interrupt flag; and, if RB_UIF_TRANSFER is valid, then continue to analyze the USB interrupt status register R8_USB_INT_ST, according to the current USB The response PID of the transmission transaction identifies MASK_UIS_H_RES for corresponding processing.

If the synchronization trigger bit RB_UH_R_TOG of the IN transaction of the host receiving endpoint is set in advance, you can use RB_U_TOG_OK or RB_UIS_TOG_OK to determine whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the host receiving endpoint. If the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After processing the USB send or receive

After receiving the interrupt, the synchronization trigger bit of the corresponding host endpoint should be modified correctly to synchronize the data packet sent next time and detect the r Whether the received data packet is synchronized; in addition, by setting RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG, the corresponding synchronization trigger bit can be automatically flipped after successful transmission or reception.

The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device to be operated and the token PID packet identification of this USB transfer transaction. The data corresponding to the SETUP token and OUT token is provided by the host sending endpoint. The data to be sent is in the R16_UH_TX_DMA buffer, and the length of the data to be sent is set in R16_UH_TX_LEN; the data corresponding to the IN token is returned by the target device to the host to receive End point, the received data is stored in the R16_UH_RX_DMA buffer, and the received data length is stored in R8_USB_RX_LEN.

Table 17-5 USB host Machine-related register list Table (marked gray is controlled by RB_UC_RESET_SIE reset)

name	address	description	Reset value
R8_UHOST_CTRL	0x40008001	USB host physical port control register	0xX0
R8_UH_EP_MOD	0x4000800d	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x40008018	USB host receive buffer start address	0xXXXX
R16_UH_TX_DMA	0x4000801c	USB host send buffer start address	0xXXXX
R8_UH_SETUP	0x40008026	USB host auxiliary setting register	0x00
R8_UH_EP_PID	0x40008028	USB host token setting register	0x00
R8_UH_RX_CTRL	0x4000802a	USB host receiving endpoint control register	0x00
R8_UH_TX_LEN	0x4000802c	USB host send length register	0xXX
R8_UH_TX_CTRL	0x4000802e	USB host sends endpoint control register	0x00

USB Host object Reason Port control register (R8_UHOST_CTR L)

Bit	name	access	description	Reset value
7	RB_UH_PD_DIS	RW	USB host port UD+/UD- pin internal pull-down Resistance control bit: 1: Disable internal pull-down; 0: Enable internal pull-down. Can be used in GPIO mode to provide pull-down resistors.	1
6	Reserved	RO	Reserved.	0
5	RB_UH_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	X
4	RB_UH_DM_PIN	RO	Current UD-pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved.	0
2	RB_UH_LOW_SPEED	RW	USB host port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full speed mode.	0
1	RB_UH_BUS_RESET	RW	USB host mode bus reset control bit: 1: Force output USB bus reset; 0: End output.	0
0	RB_UH_PORT_EN	RW	USB host port enable bit: 1: Enable the host port; 0: Disable the host port. When the USB device is disconnected, it should be automatically cleared to 0.	0

USB Host side point Mode control register (R8_UH_EP_MOD)

Bit	name	access	description	Reset value
7	Reserved	RO	Reserved.	0
6	RB_UH_EP_TX_EN	RW	Host sending endpoint sending (SETUP/OUT) enable bit: 1: Enable endpoint transmission;	0

			0: Disable endpoint sending.	
5	Reserved	RO	Reserved.	0
4	RB_UH_EP_TBUF_MOD	RW	Host send endpoint send data buffer mode control Bit.	0
3	RB_UH_EP_RX_EN	RW	Host receiving endpoint receiving (IN) enable bit: 1: Enable endpoint reception; 0: Disable endpoint reception.	0
[2:1]	Reserved	RO	Reserved.	00b
0	RB_UH_EP_RBUF_MOD	RW	USB host receiving endpoint receiving data buffer mode Control bit.	0

The combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD controls the host sending endpoint data buffer mode, refer to the table below.

Table 17-6 Host send buffer mode

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: Take R16_UH_TX_DMA as the starting address
0	X	The endpoint is disabled and the R16_UH_TX_DMA buffer is not used.
1	0	Single 64-byte send buffer (SETUP/OUT).
1	1	Double 64-byte sending buffer, select by RB_UH_T_TOG: When RB_UH_T_TOG=0, select the first 64-byte buffer; when RB_UH_T_TOG=1, select the last 64-byte buffer.

The combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD controls the host receiving endpoint data buffer mode, refer to the table below.

Table 17-7 Host receive buffer mode

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	Structure description: Take R16_UH_TX_DMA as the starting address
0	X	The endpoint is disabled and the R16_UH_RX_DMA buffer is not used.
1	0	Single 64-byte receive buffer (IN).
1	1	Dual 64-byte receive buffer, select by RB_UH_R_TOG: When RB_UH_R_TOG=0, select the first 64-byte buffer; when RB_UH_R_TOG=1, select the last 64-byte buffer.

USB Host connection Receive Buffer start address (R16_UH_RX_DM A)

Bit	name	access	description	Reset value
[15:0]	R16_UH_RX_DMA	RW	The start address of the host endpoint data receiving buffer. The lower 15 bits are valid, and the address must be aligned with 4 bytes.	XXXXb

USB Host sends give away Buffer start address (R16_UH_TX_DM A)

Bit	name	access	description	Reset value
[15:0]	R16_UH_TX_DMA	RW	The start address of the host endpoint data sending buffer. The lower 15 bits are valid, and the address must be aligned with 4 bytes.	XXXXb

USB Host auxiliary help Set register (R8_UH_SETU P)

Bit	name	access	description	Reset value
7	RB_UH_PRE_PID_EN	RW	Low-speed preamble packet PRE PID enable bit: 1: Enable, used to communicate with low-speed USB devices through an external HUB. 0: Disable the low-speed preamble packet.	0
6	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable bit: 1: The host automatically generates SOF packets; 0: Not automatically generated, but can be generated manually.	0
[5:0]	Reserved	RO	Reserved.	000000b

USB Host order brand Setting register (R8_UH_EP_P ID)

Bit	name	access	description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID packet identification of this USB transmission transaction.	0000b
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device being operated this time.	0000b

USB Host connection Receive Endpoint control register (R8_UH_RX_CTR L)

Bit	name	access	description	Reset value
7	RB_UH_R_TOG	RW	Synchronous trigger bits expected by the USB host receiver (handling IN transactions): 1: Expect DATA1; 0: Expect DATA0.	0
[6:5]	Reserved	RO	Reserved.	00b
4	RB_UH_R_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding expectations after the data is received successfully Synchronous trigger bit (RB_UH_R_TOG); 0: Do not turn over automatically, you can switch manually.	0
3	Reserved	RO	Reserved.	0
2	RB_UH_R_RES	RW	The response control bit of the host receiver to the IN transaction: 1: No response, used for real-time/synchronization of non-zero endpoints transmission; 0: Acknowledge ACK.	0
[1:0]	Reserved	RO	Reserved.	00b

USB Host sends give away Length register (R8_UH_TX_L EN)

Bit	name	access	description	Reset value
[7:0]	R8_UH_TX_LEN	RW	Set the USB host to send the data that the endpoint is ready to send The number of bytes.	XXh

USB Host sends give away Endpoint control register (R8_UH_TX_CTR L)

Bit	name	access	description	Reset value
7	Reserved	RO	Reserved.	0
6	RB_UH_T_TOG	RW	Synchronous trigger bits prepared by the USB host transmitter (processing SETUP/OUT transactions): 1: means sending DATA1; 0: means sending DATA0.	0
5	Reserved	RO	Reserved.	0
4	RB_UH_T_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: Automatically flip the corresponding synchronization after the data is sent successfully Trigger bit (RB_UH_T_TOG); 0: Do not turn over automatically, you can switch manually.	0
[3:1]	Reserved	RO	Reserved.	000b
0	RB_UH_T_RES	RW	The response control bit of the USB host transmitter to the SETUP/OUT transaction: 1: Expect no response, used for real-time non-zero endpoints/ Synchronous transmission 0: Expect to answer ACK.	0

Chapter 18 Ethernet Controller ETH

18.1 Introduction to Ethernet Controller

The chip integrates the Ethernet controller MAC, PHY and DMA, and is compatible with the IEEE802.3 protocol. The internal DMA transfers and receives data to the system RAM. The PHY physical layer is a 10Mbit/S Ethernet transceiver, which provides part of the network PHY register to set the transmission and reception performance.

The network bottom operation mainly provides application support with subroutine library, and does not introduce the register in detail. Main features:

- Supports full-duplex and half-duplex.
- Support short packet filling settings.
- Support CRC setting and filling.
- Support jumbo frame reception.
- Support different combinations of filtering modes.
- Support pause frame sending and setting.
- Support auto-negotiation mechanism.
- Support DMA for sending and receiving data.
- The PHY transceiver is compatible with 10BASE-T, and the transmitting module supports an energy-saving mode.
- Built-in 50 ohm transmission impedance matching resistance, can also choose to connect externally.
- Provide a globally unique MAC address assigned by IEEE.

18.2 Register description

Table 18-1 List of related registers of the Ethernet controller

name	Offset address	description	Reset value
R8_ETH_EIE	0x40009003	Interrupt enable register	0x00
R8_ETH_EIR	0x40009004	Interrupt flag register	0x00
R8_ETH_ESTAT	0x40009005	Status register	0x00
R8_ETH_ECON2	0x40009006	PHY analog parameter setting register	0x06
R8_ETH_ECON1	0x40009007	Transceiver control register	0x00
R32_ETH_TX	0x40009008	Send DMA control register	0xXXXXXXXX
R16_ETH_ETXST	0x40009008	Send DMA buffer start address register	0xXXXX
R16_ETH_ETXLN	0x4000900A	Send length register	0xXXXX
R32_ETH_RX	0x4000900C	Receive DMA control register	0x00000000
R16_ETH_ERXST	0x4000900C	Receive DMA buffer start address register	0x0000
R16_ETH_ERXLN	0x4000900E	Receive length register	0x0000
R32_ETH_HTL	0x40009010	Hash table low byte register	0x484EA033
R32_ETH_HTH	0x40009014	Hash table high byte register	0x5000EF97
R32_ETH_MACON	0x40009018	Acceptance filter setting register	0x10000000
R8_ETH_ERXFCON	0x40009018	Receive packet filter control register	0x00
R8_ETH_MACON1	0x40009019	Mac laminar flow control register	0x00
R8_ETH_MACON2	0x4000901A	Mac layer packet control register	0x00
R8_ETH_MABBIPG	0x4000901B	Minimum interval between packets register	0x10
R32_ETH_TIM	0x4000901C	Flow control pause frame time register	0xXXXXXXXX
R16_ETH_EPAUS	0x4000901C	Flow control pause frame time register	0xXXXX
R16_ETH_MAMXFL	0x4000901E	Maximum received packet length register	0x0000
R16_ETH_MIRD	0x40009020	MII read data register	0x1100
R32_ETH_MIWR	0x40009024	MII write register	0x00000000
R8_ETH_MIREGADR	0x40009024	MII address register	0x00

R8_ETH_MISTAT	0x40009025	MII status register	0x00
R16_ETH_MIWR	0x40009026	MII write data register	0x0000
R32_ETH_MAADRL	0x40009028	MAC address low byte register	0xXXXXXXXX
R16_ETH_MAADRH	0x4000902C	MAC address high byte register	0xXXXX

in Disable enable Register (R8_ETH_EIE)

Bit	name	access	description	Reset value
7	RB_ETH_EIE_INTIE	RW	Ethernet interruption enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0
6	RB_ETH_EIE_RXIE	RW	Receive completion interrupt enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0
5	Reserved	RO	Reserved.	0
4	RB_ETH_EIE_LINKIE	RW	Link change interrupt enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0
3	RB_ETH_EIE_TXIE	RW	Send completion interrupt enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0
2	RB_ETH_EIE_R_EN50	RW	The built-in 50 ohm impedance matching resistance enable: 0: On-chip resistance is disconnected; 1: On-chip resistance is connected.	0
1	RB_ETH_EIE_TXERIE	RW	Send error interrupt enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0
0	RB_ETH_EIE_RXERIE	RW	Receive error interrupt enable: 0: Turn off the interrupt; 1: Turn on interrupt.	0

in Broken sign post Register (R8_ETH_EIR)

Bit	name	access	description	Reset value
7	Reserved	RO	Reserved.	0
6	RB_ETH_EIR_RXIF	RW1	Receive complete flag.	0
5	Reserved	RO	Reserved.	0
4	RB_ETH_EIR_LINKIF	RW1	Link change sign.	0
3	RB_ETH_EIR_TXIF	RW1	Send completed flag.	0
2	Reserved	RO	Reserved.	0
1	RB_ETH_EIR_TXERIF	RW1	Send error flag.	0
0	RB_ETH_EIR_RXERIF	RW1	Receive error flag.	0

shape State deposit Device (R8_ETH_ESTAT)

Bit	name	access	description	Reset value
7	RB_ETH_ESTAT_INT	RW1	Interrupted.	0
6	RB_ETH_ESTAT_BUFER	RW1	Buffer error.	0
5	RB_ETH_ESTAT_RXCRCER	RO	Receive CRC error.	0
4	RB_ETH_ESTAT_RXNIBBLE	RO	Receive nibble error.	0
3	RB_ETH_ESTAT_RXMORE	RO	Receive more than the set maximum data packet.	0
2	RB_ETH_ESTAT_RXBUSY	RO	Receiving is in progress.	0
1	RB_ETH_ESTAT_TXABRT	RO	The transmission was interrupted by the MCU.	0
0	Reserved	RO	Reserved.	0

PHY Simulation parameters number Setting register (R8_ETH_ECO N2)

Bit	name	access	description	Reset value

[7:4]	Reserved	RO	Reserved.	0000b
[3:1]	RB_ETH_ECON2_RX RB_ETH_ECON2_MUST	RW	Reserved, 011 must be written.	011b
0	RB_ETH_ECON2_TX	RW	Energy-saving drive control at the sender: 0: Rated drive; 1: Energy-saving drive.	0

Receive Send control Register (R8_ETH_ECON1)

Bit	name	access	description	Reset value
7	RB_ETH_ECON1_TXRST	RW	Send module reset: 0: Do not reset; 1: Reset the sending module.	0
6	RB_ETH_ECON1_RXRST	RW	Receiving module reset: 0: Do not reset; 1: Reset the receiving module.	0
[5:4]	Reserved	RO	Reserved.	00b
3	RB_ETH_ECON1_TXRTS	RW	Sending starts, and it is automatically cleared after sending: 1: Start sending; 0: No action.	0
2	RB_ETH_ECON1_RXEN	RW	Receive enable control: 0: Turn off receiving; 1: Turn on receiving.	0
[1:0]	Reserved	RO	Reserved.	00b

Send DMA buffer ground Address register (R16_ETH_E TXST)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_ETXST	RW	Send the start address of the DMA buffer. The lower 15 bits are valid, and the address must be aligned with 4 bytes.	XXXXh

hair Delivery length (R16 _ETH_ETXLN)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_ETXLN	RW	Send length.	XXXXh

Receive DMA buffer ground Address register (R16_ETH_E RXST)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_ERXST	RW	The start address of the receiving DMA buffer. The lower 15 bits are valid, and the address must be aligned with 4 bytes.	XXXXh

Pick up Receive length deposit Device (R16_ETH_ERXLN)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_ERXLN	RO	Receive length.	0000h

Hash table low byte Register (R32_ETH_HTL)

Bit	name	access	description	Reset value
[31:24]	R8_ETH_EHT3	RW	Hash Table byte 3.	48h
[23:16]	R8_ETH_EHT2	RW	Hash Table byte 2.	4Eh
[15:8]	R8_ETH_EHT1	RW	Hash Table byte 1.	A0h
[7:0]	R8_ETH_EHT0	RW	Hash Table byte 0.	33h

Hash table high byte Register (R32_ETH_HTH)

Bit	name	access	description	Reset value
[31:24]	R8_ETH_EHT7	RW	Hash Table byte 7.	50h

[23:16]	R8_ETH_EHT6	RW	Hash Table byte 6.	00h
[15:8]	R8_ETH_EHT5	RW	Hash Table byte 5.	EFh
[7:0]	R8_ETH_EHT4	RW	Hash Table byte 4.	97h

Pick up Filter Assume Set the register (R8_ETH_ERXFCO N)

Bit	name	access	description	Reset value
7	RB_ETH_ERXFCON_UCEN	RW	Target address matching filter settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 target address The match will be received, RB_ETH_ERXFCON_ANDOR =1 Mismatched destination address will be filtered.	0
6	RB_ETH_ERXFCON_ANDOR	RW	Filter conditions AND and OR settings: 0: Any packet that satisfies the filter conditions will be received; 1: The packet is received only when all filter conditions are met.	0
5	RB_ETH_ERXFCON_CRCEN	RW	CRC check filter settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 CRC check Correct will be received, RB_ETH_ERXFCON_ANDOR =1 CRC check errors will be filtered.	0
4	Reserved	RO	Reserved.	0
3	RB_ETH_ERXFCON_MPEN	RW	Magic packet filtering settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 magic packet will Is received, RB_ETH_ERXFCON_ANDOR=1 is not a magic The French packet will be filtered.	0
2	RB_ETH_ERXFCON_HTEN	RW	Hash table matching filter settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 hash table match The configuration will be received, RB_ETH_ERXFCON_ANDOR =1 The hash table does not match and will be filtered.	0
1	RB_ETH_ERXFCON_MCEN	RW	Multicast packet matching filtering settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 multicast packet matching The configuration will be received, RB_ETH_ERXFCON_ANDOR=1 Multicast packets that do not match will be filtered.	0
0	RB_ETH_ERXFCON_BCEN	RW	Broadcast packet matching filter settings: 0: Do not use the filter condition; 1: RB_ETH_ERXFCON_ANDOR=0 The broadcast packet will Was received, RB_ETH_ERXFCON_ANDOR=1 non-wide Broadcast packets will be filtered.	0

Mac Laminar flow control register (R8_ETH_MACON1)

Bit	name	access	description	Reset value
[7:6]	Reserved	RO	Reserved.	00b
[5:4]	RB_ETH_MACON1_FCEN	RW	Pause frame setting, valid under full duplex: 00: Stop sending pause frames; 01: Send a pause frame once, then stop sending; 10: Periodically send pause frames; 11: Send 0 timer pause frame, then stop sending	00b

			give away.	
3	RB_ETH_MACON1_TXPAUS	RW	Send pause frame enable control: 0: Do not send pause frames; 1: Enable sending.	0
2	RB_ETH_MACON1_RXPAUS	RW	Receiving pause frame enable: 0: Do not receive pause frames; 1: Enable reception.	0
1	RB_ETH_MACON1_PASSALL	RW	Control frame settings: 0: The control frame will be filtered; 1: Control frames that are not filtered will be written to the buffer.	0
0	RB_ETH_MACON1_MARXEN	RW	MAC layer receiving enable: 0: MAC does not receive data; 1: MAC reception is enabled.	0

Mac Layer packet control register (R8_ETH_MACON2)

Bit	name	access	description	Reset value
[7:5]	RB_ETH_MACON2_PADCFG	RW	Short package filling settings: 7: All short packets are filled with 0 to 64 bytes, and then 4 bytes of CRC; 6: Do not fill short bags; 5: It is detected that the VLAN network packet with the field of 8100h is automatically filled with 0 to 64 bytes, otherwise the short packet is filled with 60 bytes of 0, and then 4 bytes of CRC are filled; 4: The short packet is not filled; 3: All short packets are filled with 0 to 64 bytes, and then 4 bytes of CRC; 2: Do not fill short bags; 1: All short packets are filled with 0 to 60 bytes, and then 4 bytes of CRC; 0: Do not fill short packets.	000b
4	RB_ETH_MACON2_TXCRCEN	RW	Send to add CRC control: 0: Hardware does not fill CRC; 1: Hardware fills CRC.	0
3	RB_ETH_MACON2_PHDREN	RW	The special 4 bytes do not participate in CRC	0
2	RB_ETH_MACON2_HFRMEN	RW	check. Allow to receive jumbo frames: 0: Do not allow receiving jumbo frames; 1: Allow receiving.	0
1	Reserved	RO	Reserved.	0
0	RB_ETH_MACON2_FULDPX	RW	Ethernet communication mode: 0: half duplex; 1: Full duplex.	0

most Small private room between Interval register (R8_ETH_MABBIP G)

Bit	name	access	description	Reset value
7	Reserved	RO	Reserved.	0
[6:0]	R8_ETH_MABBIPG	RW	The minimum number of bytes between packets.	0010000b

flow Control pause frame Time register (R16_ETH_E PAUS)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_EPAUS	RW	Flow control pause frame time.	XXXXh

most Big receive packet long Degree register (R16_ETH_MAM XFL)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_MAMXFL	RW	Maximum received packet length.	0000h

MII Read data Register (R16_ETH_MIRD)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_MIRD	RW	MII reads the data register.	1100h

MII Address Register (R8_ETH_MIREGADR)

Bit	name	access	description	Reset value
[7:5]	Reserved	RO	Reserved.	000b
[4:0]	RB_ETH_MIREGADR_MIRDL	RW	PHY register address.	00000b

MII Status post Register (R8_ETH_MISTAT)

Bit	name	access	description	Reset value
[7:1]	Reserved	RO	Reserved.	0000000b
0	R8_ETH_MII_STA	RO	MII register operation status: 1: Write MII register; 0: Read MII register	0

MII Write data Register (R16_ETH_MIWR)

Bit	name	access	description	Reset value
[15:0]	R16_ETH_MIWR	WO	MII write data register.	0000h

MAC Address hosting Device (R32_ETH_MAADRL, R16_ETH_ MAADRH)

Bit	name	access	description	Reset value
[31:0]	R32_ETH_MAADRL	RW	MAC Address byte 1 ～4.	XXXXXXXXh
[15:0]	R16_ETH_MAADRH	RW	MAC Address byte 5～6.	XXXXh

18.3 Operation Guide

1. Initialization

(1) Configure the security register to enter the security mode, and turn on the clock and power supply of the Ethernet network;

(2) Turn on the corresponding interrupt, optionally, enable impedance matching resistance;

(3) Configure the receiving filter mode, CRC function, and MAC address;

(4) Set the cache;

(5) Start receiving and enable interrupt.

2. Send data

(1), write R16_ETH_ETXLN data length;

(2) Write the data address of R16_ETH_ETXST;

(3) Enable the RB_ETH_ECON1_TXRTS flag to start sending.

3. Receive data

(1) Pre-set the receiving address and enable receiving;

(2) Use interrupt or query to receive completion status;

(3) Read R16_ETH_ERXLN receiving length;

(4) Update R16_ETH_ERXST receiving address

For specific applications, please use the Ethernet protocol stack library, and refer to the network application examples provided.

Chapter 19 Wireless Communication

19.1 Introduction

The chip integrates a low-power 2.4-GHz wireless communication module, including RF transceiver, baseband and link control, and antenna matching network, and supports low-power Bluetooth BLE. More than one hundred registers are provided internally for adjusting parameters and controlling the process and status. This manual do For a detailed introduction, only part of the commonly used registers of the application layer are listed. The underlying operation of wireless communication mainly provides application su

Main features:

- Integrated 2.4GHz radio frequency transceiver, BaseBand baseband and LLE link control. Supports low-power
- Bluetooth BLE and complies with Bluetooth Low Energy 4.2 specifications. The single-ended RF interface
- eliminates the need for external inductor and capacitor matching filter networks, simplifying board-level design.
- Receiving sensitivity -93dBm.
- Programmable -20dBm to +3dBm transmit power, support dynamic adjustment.
- The wireless communication distance using the PCB on-board antenna is about 100 meters when the
- transmit power is 0dBm. After using the built-in DC-DC conversion, the current is more than 6mA at 0dBm
- transmission power. Support AES encryption and decryption.
- Support DMA.
- Provide optimized protocol stack and application layer API to support networking.

19.2 LLE module

The LLE module supports automatic sending and receiving mode and manual sending and receiving mode, 5 groups of independent hardware timers, which can control the sendin The time point of the process.

19.2.1 Register description

Table 19-1 List of related registers of LLE module

name	Offset address	description	Reset value
R32_LLE_CTRL_CMD	0x4000C200	LLE command	0x00000000
R32_LLE_CTRL_CFG	0x4000C204	LLE configuration	0x00000F01
R32_LLE_STATUS	0x4000C208	LLE status	0x00000000
R32_LLE_INT_EN	0x4000C20C	LLE interrupt enable	0x00003F1F
R32_LLE_CTRL_MOD	0x4000C250	LLE mode control	0x00000011

19.2.2 Function and configuration

Initialization process:

(1) Set R32_LLE_CTRL_MOD register, write 58H, configure to LLE mode, and turn on the power to enable DMA and BB module function.

(2) Set the R32_LLE_CTRL_CFG register, configure LLE to automatically send and receive data mode, and manually send and receive data mode. (3) Set the R32_LLE_CTRL_CMD register, select 4 working states, send, receive, stop, and close. details as follows:

CMD_RX Use: Set the current mode to receive, it will be cleared to 0 automatically after receiving in manual mode, and it will continue to cycle in automatic mode. The process of ring sending and receiving can be stopped by CMD_STOP or CMD_SHUT.

CMD_TX Use: Set the current mode to send, it will be automatically cleared to 0 after sending is completed in manual mode, and it will continue to cycle in automatic mode. The process of ring sending and receiving can be stopped by CMD_STOP or CMD_SHUT.

CMD_STOP use: In automatic transmission mode, CMD_STOP is set to 1 during the frame interval, and the CMD_TX state will be cleared and jump to LLE_IDLE, the software cannot directly clear the CMD_TX state. In automatic receiving mode, when CMD_STOP is set to 1 during the frame interval, the CMD_TX state will be cleared and jump to LLE_IDLE, and the software cannot directly clear the CMD_RX state.

CMD_SHUT use: After CMD_SHUT is set, CMD_TX/CMD_RX/CMD_STOP/CMD_SHUT will all be cleared, and at the same time

The state machine of LLE is reset.

19.3 DMA module

The controller has 2 groups of DMA, and each group of DMA has two channels. The two channels of DMA0 are used to send data and receive data respectively. The two channels of DMA2 are used in automatic mode. In automatic transmission mode, you can configure the address of sending DMA and the address of receiving DMA at the same time, so that there is no need during the frame interval. Configure again.

19.3.1 Register description

Table 19-2 DMA module related register list

name	Offset address	description	Reset value
R32_DMA0_CTRL_CFG	0x4000C008	DMA0 configuration register	0x00002000
R32_DMA2_CTRL_CFG	0x4000C00C	DMA2 configuration register	0x00002000
R32_DMA0_TX_SRC	0x4000C010	DMA0 source address, Used to configure the starting address of sending data	0x00000000
R32_DMA2_TX_SRC	0x4000C014	DMA2 source address, automatic receiving mode, Used to configure the starting address of sending data	0x00000000
R32_DMA0_RX_DST	0x4000C018	DMA0 destination address, Used to configure the start address of received data	0x00000000
R32_DMA2_RX_DST	0x4000C01C	DMA2 destination address, automatic transmission mode, Used to configure the start address of received data	0x00000000

19.3.2 Functions and Configuration

Sending operation process:

(1) Set the R32_DMA0_TX_SRC register and write the sending data address.

(2) Set the R32_DMA0_CTRL_CFG register and write the length of the sent data.

(3) Set the IRQMASK of the R32_DMA0_CTRL_CFG register to 1.

(4) In automatic mode, set the R32_DMA2_RX_DST register and write the receive data address.

(5) In automatic mode, set the IRQMASK of the R32_DMA2_CTRL_CFG register to 1. Receiving operation process:

(1) Set the R32_DMA0_RX_DST register and write the receiving data address.

(2) Set the IRQMASK of the R32_DMA0_CTRL_CFG register to 1. (3) In automatic mode, set R32_DMA2_TX_SRC and write the sending data address.

(4) In automatic mode, set the R32_DMA2_CTRL_CFG register and write the length of the sent data.

(5) In automatic mode, set the IRQMASK of the R32_DMA2_CTRL_CFG register to 1.

19.4 BB module

19.4.1 Register description

Table 19-3 BB module related register list

name	Offset address	description	Reset value
R32_BB_CTRL_CFG	0x4000C100	BB configuration register	0x000822A7
R32_BB_TXCRC_INIT	0x4000C104	Send CRC initial value configuration register	0x0056BC9B
R32_BB_TXACCS_ADDR	0x4000C108	Send access address configuration register	0x8E89BED6
R32_BB_RXCRC_INIT	0x4000C10C	Receive CRC initial value configuration register	0x0056BC9B
R32_BB_RXACCS_ADDR	0x4000C110	Receive access address configuration register	0x8E89BED6

R32_BB_CTRL_TX	0x4000C12C	Send control register	0x00010008
R32_BB_RSSI_ST	0x4000C130	BB received signal quality status	0x00000000
R32_BB_INT_EN	0x4000C134	BB interrupt enable	0x00000003
R32_BB_INT_ST	0x4000C138	BB interrupt status	0x00000000

19.4.2 Functions and Configuration

Data whitening can be configured, sending and receiving AA can be configured, sending and receiving CRC initial value can be configured, and communication channel can be configured. Send data configuration:

(1) Configure the sending access address.

(2) Configure the initial value of sending CRC.

(3) Configure the communication channel for sending data.

Receive data configuration:

(1) Configure the receiving access address.

(2) Configure the initial value of receiving CRC.

(3) Configure the communication channel for receiving data.

19.5 AES module

19.5.1 Register description

Table 19-4 AES module related register list

name	Offset address	description	Reset value
R32_AES_CTRL_CCMMOD	0x4000C300	AES mode register	0x00000060
R32_AES_CCMINT_EN	0x4000C304	AES interrupt register	0x00000000
R32_AES_CCMVT_INIT0	0x4000C308	AES CCM mode initial vector value	0x00000000
R32_AES_CCMVT_INIT1	0x4000C30C	AES CCM mode initial vector value	0x00000000
R32_AES_PKT_CNT0	0x4000C310	AES CCM mode packet count value	0x00000000
R32_AES_PKT_CNT0	0x4000C314	AES CCM mode packet count value	0x00000000
R32_AES_DATA0	0x4000C318	AES data register	0x00000000
R32_AES_DATA1	0x4000C31C	AES data register	0x00000000
R32_AES_DATA2	0x4000C320	AES data register	0x00000000
R32_AES_DATA3	0x4000C324	AES data register	0x00000000
R32_AES_KEY0	0x4000C328	AES encryption key register 0	0x00000000
R32_AES_KEY1	0x4000C32C	AES encryption key register 1	0x00000000
R32_AES_KEY2	0x4000C330	AES encryption key register 2	0x00000000
R32_AES_KEY3	0x4000C334	AES encryption key register 3	0x00000000
R32_AES_KEY4	0x4000C338	AES encryption key register 4	0x00000000
R32_AES_KEY5	0x4000C33C	AES encryption key register 5	0x00000000
R32_AES_KEY6	0x4000C340	AES encryption key register 6	0x00000000
R32_AES_KEY7	0x4000C344	AES encryption key register 7	0x00000000
R32_AES_RAND0	0x4000C348	AES random value	0x96906220
R32_AES_RAND1	0x4000C34c	AES random value	0x8A3BBF80
R32_AES_RAND2	0x4000C350	AES random value	0xB5625304
R32_AES_RAND3	0x4000C354	AES random value	0x2A43B7E8

For specific applications, please use the BLE protocol stack library and refer to the provided BLE application examples.

Chapter 20 Parameters

20.1 Absolute maximum

Critical or exceeding the absolute maximum value may cause the chip to work abnormally or even be damaged.

Table 20-1 Absolute maximum value parameter table

name	Parameter Description	Minimum	Max	unit
TA	Ambient temperature during work	-40	85	℃
TS	Ambient temperature during storage	-40	105	℃
VIO33	System power supply voltage (VIO33 connects to the power supply, GND connects to the ground)	-0.4	3.9	V
VIO	Voltage on input or output pin	-0.4	VIO33+0.4	V
VIO5	Support 5V withstand voltage input or output pin voltage	-0.4	5.5	V
VDCI	The voltage on the VDCID/VDCIA pin (if an external DC-DC is used)	-0.4	VIO33+0.4	V
VXCK	X32MI/X32MO/PA10/PA11 voltage after enabling LSE	-0.3	1.4	V

20.2 Electrical parameters

Test conditions: TA=25 ℃, VIO33=3.3V, Fsys=16MHz.

Table 20-2 Electrical parameters table

name	Parameter Description		Minimum	Typical value	Max	unit
VIO33	System power supply voltage		2.1	3.3	3.6	V
ICC_{8}	Pass-through Quiescent power supply current	Fsys=8M	1.3	1.5		mA
ICC_{16}		Fsys=16M	1.6	1.9		mA
ICC_{32}		Fsys=32M	2.3	2.8		mA
IDDC_{8}	After enabling the built-in DC-DC Quiescent power supply current	Fsys=8M	0.8	1.1		mA
IDDC_{16}		Fsys=16M	1.0	1.3		mA
IDDC_{32}		Fsys=32M	1.5	1.8		mA
VIL	GPIO low-level input voltage		0		0.9	V
VIH	GPIO high level input voltage		2.0		VIO33	V
VIL5	Support 5V withstand voltage GPIO low-level input voltage		0		0.9	V
VIH5	Support 5V withstand voltage GPIO high-level input voltage		2.0		5.0	V
VOL	low-level output voltage (5mA/20mA sink current)		0	0.3	0.4	V
VOH	High level output voltage (5mA/20mA output current)		VIO33-0.4	VIO33-0.3	VIO33	V
IIN	GPIO floating input input current		-3	0	3	uA
IUP	The input current of the input terminal of the GPIO built-in pull-up resistor		25	60	90	uA
IDN	The input current of the input end of the GPIO built-in pull-down resistor		-90	-60	25	uA
Vref	VINTA pin voltage (ADC reference voltage)		1.035	1.05	1.065	V
Vdci	The voltage of the VDCID pin after enabling DC-DC		1.33	1.37	1.43	V

20.3 Low-power mode power consumption

Test conditions: TA=25 ℃, VIO33=3.3V, Fsys=16MHz.

Table 20-3 Low power consumption parameter table (for Reference, and Wen Degree correlation)

Low power mode	Minimum	Typical value	Max	unit
Idle mode, open the clock combination of each module	1.15	1.2	1.6	mA
Idle mode, turn off all sleep clocks		1.15		mA

Pause mode, FlashROM standby		470		uA
Pause mode, FlashROM disabled		420		uA
Sleep mode, multiple combinations, refer to Table 5-3	0.6		2.0	uA
Sleep mode, PMU+core+RAM2K, GPIO wake-up, no RTC		0.6		uA
Power-off mode, multiple combinations, refer to Table 5-3	0.2		1.3	uA
Power-down mode, only PMU, reset after GPIO wake-up, no RTC		0.2		uA

Table 20-4 Current of each module (for reference only Kao, and Wen Degree correlation)

name	Parameter Description			Minimum	Typical value	Max	unit
$I_{DD(RAM2K)}$	RAM2K: 2KB SRAM				0.3		uA
$I_{DD(RAM14K)}$	RAM14K: 14KB SRAM				0.8		uA
$I_{DD(LSI)}$	Internal LSI oscillator				0.3		uA
$I_{DD(LSE)}$	External LSE oscillator				0.4		uA
$I_{DD(HSI)}$	Internal HSI oscillator				160		uA
$I_{DD(HSE)}$	External HSE oscillator			100	200	300	uA
$I_{DD(BM)}$	Battery low voltage monitoring BM module (in pause mode)				3		uA
$I_{DD(PLL)}$	Internal PLL oscillator				1.0		mA
$I_{DD(ADC)}$	ADC module				0.27		mA
$I_{DD(TKEY)}$	TouchKey module				0.08		mA
$I_{DD(TS)}$	Temperature sensor TS module				0.1		mA
$I_{DD(USB)}$	USB module		Non-sending state	1.2	1.6	2.0	mA
			send status		3		mA
$I_{DD(ETH)}$	Ethernet ETH module		Non-sending state	2.2	2.6	3.0	mA
			Continuous transmission: rated drive	75	100		mA
			continuous hair Send: Energy-saving drive	56	75		mA
$I_{DD(BLE)}$	BLE Bluetooth	receive	Direct power supply		11		mA
			Enable DC-DC		6		mA
		-20dBm Transmit power	Direct power supply		4		mA
			Enable DC-DC		2		mA
		0dBm Transmit power	Direct power supply		10		mA
			Enable DC-DC		5		mA
		+ 3dBm Transmit power	Direct power supply		16		mA
			Enable DC-DC		8		mA

20.4 Clock source

Table 20-5 High-speed oscillator HSI and HSE

name	parameter description		Minimum	Typical value	Max	unit
F_{HSI}	Internal HSI oscillator frequency			32		MHz
A_{HSI}	HSI oscillator accuracy	TA=-40 ℃～85℃		2.0	3.0	%
		TA=-5 ℃～70℃		0.7	1.0	%
		TA=10 ℃～45℃		0.3	0.5	%
T_{SUHSI}	The internal HSI oscillator starts to stabilize time			0.2	1	uS
F_{HSE}	External HSE oscillator frequency (when wireless communication is turned off)		twenty four	32	36	MHz
T_{SUHSE}	External HSE oscillator start to stable time		200	500	1200	uS

Table 20-6 Low-speed oscillator LSI and LSE

name	Parameter Description		Minimum	Typical value	Max	unit
F_{LSIR}	Internal LSI oscillator frequency (before calibration)		20K	32K	48K	Hz
F_{LSI}	Internal LSI frequency (application After calibration while the software is running)		32700	32768	32836	Hz
A_{LSI}	LSI oscillator accuracy (After software calibration)	TA=-40 ℃～85℃		0.2	0.9	%
		TA=0 ℃～60℃		0.1	0.3	%
T_{SULSI}	Internal LSI oscillator start to stable time			40	100	uS
T_{SULSE}	External LSE oscillator start to stable time		150	450	1000	mS

Table 20-7 PLL characteristics

name	Parameter Description	Minimum	Typical value	Max	unit
F_{PLL}	PLL multiplication output clock (CK32M * 15 times)		480		MHz
T_{PLLLK}	PLL lock time		1	3	mS

20.5 Time parameters

Test conditions: TA=25 ℃, VIO33=3.3V, Fsys=6.4MHz.

Table 20-8 Time parameters

name	Parameter Description		Minimum	Typical value	Max	unit
T_{VR}	VIO33 voltage rise time (ch410 can be added for slower time)		1		50000	uS
T_{rpor}	Reset delay after power-on reset RPOR		11	15	20	mS
T_{rst}	RST# effective signal width			100		nS
T_{mr}	Reset delay after external reset MR		2	8	18	uS
T_{sr}	Reset delay after software reset SR		2	8	18	uS
T_{wtr}	Reset delay after watchdog resets WTR		10	12	18	uS
T_{WAK}	From low power state Wake-up time to exit	Idle mode	0.6	0.8	1.6	uS
		Pause mode, ROM standby	0.7	1.0	2.0	uS
		Pause mode, ROM disabled	9	11	15	uS
		Sleep mode	300	330	400	uS
		Power down mode	350	380	450	uS

Note: The delay parameters in the above table are all based on multiples of Tsys. Decreasing the main frequency will increase the delay.

The delay parameters in the above table are based on the use of the internal HSI clock source. If the external HSE clock source is used during sleep, the table will be suspended.

Mode/sleep mode/power-off mode delay parameter T_{WAK} Both will add about 1.5mS extra (start to stable).

20.6 Other parameters

Test conditions: TA=25 ℃, VIO33=3.3V, Fsys=16MHz.

Table 20-9 Other parameters

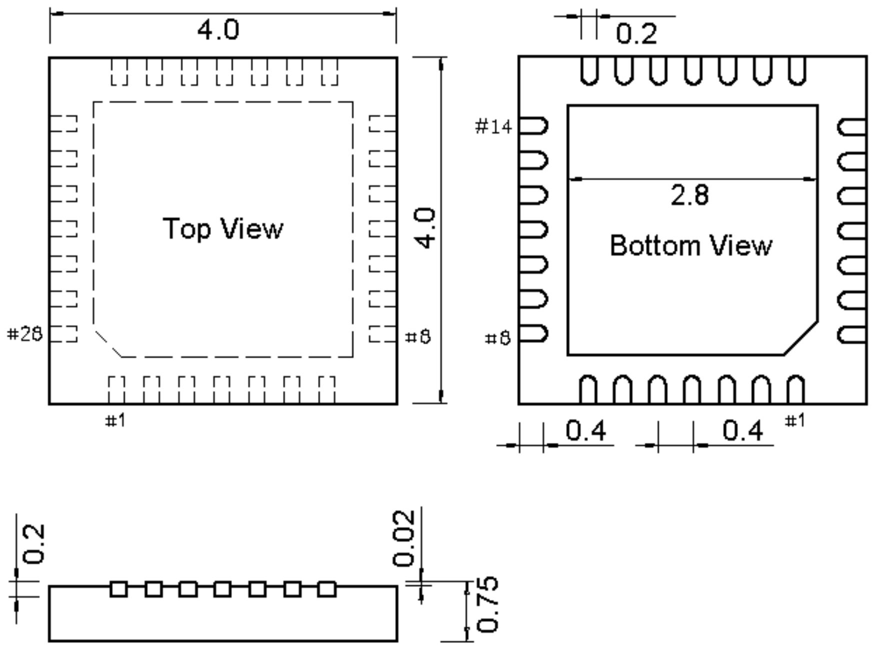
name	Parameter Description		Minimum	Typical value	Maximum	unit
R_{TS}	Measuring range of TS temperature sensor		-40		90	℃
A_{TSC}	Measurement error of temperature sensor after software calibration			±6		℃
T_{FRER}	Flash-ROM single sector erase operation time		1.1	1.4	2.4	mS
T_{FRPG}	Flash-ROM single word programming operation time		twenty two	27	36	uS
N_{EPCE}	Flash-ROM erasing times erase/program cycle endurance	5 ～45℃	100K	800K (non-guaranteed)		times
		-40 ～85℃	30K	100K (non-guaranteed)		
T_{DR}	Flash-ROM data retention capability		10			years
V_{ESD}	I/O input or output ESD withstand voltage on the pin	Antenna ANT	2K	4K (non-guaranteed)		V
		I/O pins: PA and PB	4K	6K (non-guaranteed)		V

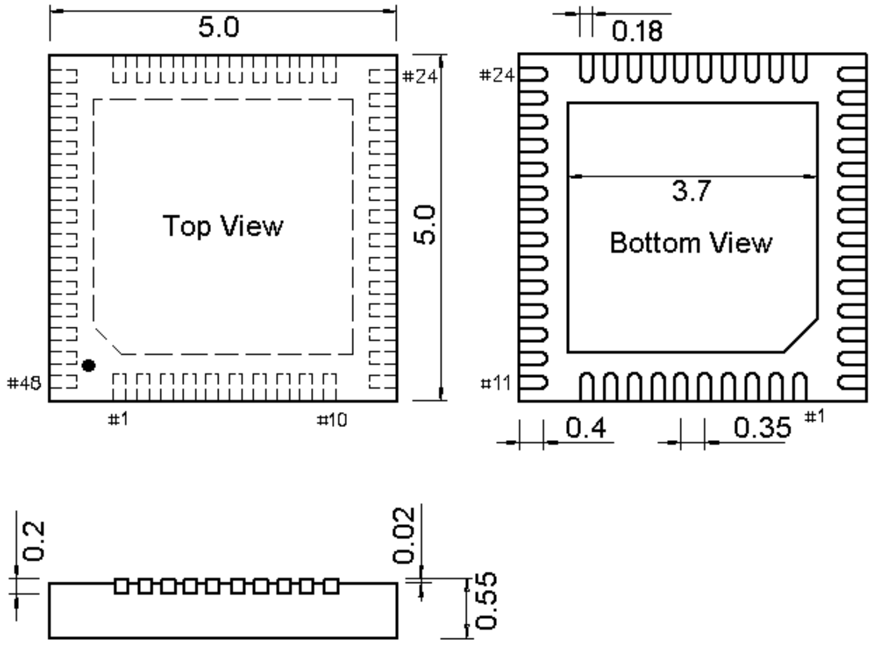
Chapter 21 Packaging

Chip packaging

Package form	Body width	Pin pitch		Package description	Order model
QFN28	4*4mm	0.4mm	15.7mil	Square without lead 28 feet	CH579F
QFN48	5*5mm	0.35mm	13.8mil	48-pin square without lead	CH579M

Note: The unit of dimensioning is mm (millimeters) , The distance between the center of the pins is the nominal value, and the size error other than that is not more than ±0.2mm.





Chapter 22 Modification Record

version	date	Description
V0.9	2018.05.09	Original Issue
V1.0	2018.09.28	Official release
V1.1	2018.10.23	Section 5.2 adds RB_PWR_MUST_0010, Section 6.4.4 is changed (1), Section 15.2 is changed to R8_ADC_CFG, 20.6 Section Change N_{EPCE} , Section 6.3 R16_OSC_CAL_CNT revised typos, Table 17-3 revised typos
V1.2	2019.03.04	Section 1.2 changes VSW to inductance, Section 5.2 changes R16_POWER_PLAN, Section 5.3 deletes low-voltage detection in sleep mode, Section 6.3 changes R16_CLK_SYS_CFG, and Sections 16.2 and 16.3 deletes RB_CHG_CNT
V1.3	2019.04.13	Table 5-2 Add WFE, Table 20-9 Adjust parameters
V1.4	2019.07.26	Section 7.4 R16_PIN_ANALOG_IE part of the control bit, Section 14.1 R8_SLV_RD_DATA address
V1.5	2020.04.15	RF only keeps BLE, adjust the steps in section 6.4.4, change the low-speed typo in section 17.2.2
V1.6	2020.12.02	Correct the typo, Section 6.3 changes R16_CLK_SYS_CFG, the timer increases the counting mode description, Table 2-3 Change the default value, add Tvr to Table 20-8, add 10.5 SPI timing