



复旦微电子

FM3316/3313/3312

低功耗系列 MCU

固件库使用说明



前言

本说明适用于复旦微电子集团推出的 C251 低功耗系列 MCU 的编程，本说明及所提供的库函数仅供参考。

固件库及说明持续完善中，欢迎关注最新版本。



目录

FM3316/3313/3312	1
低功耗系列 MCU	1
固件库使用说明	1
1 工程使用说明	8
2 GPIO	9
2.1 GPIO 寄存器结构:	9
2.2 GPIO 函数库:	11
2.2.1 <i>GPIO_init</i>	11
2.2.2 <i>GPIO_Read_Input_Data_Bit</i>	12
2.2.3 <i>GPIO_Write_Output_Data_Bit</i>	12
2.2.4 <i>GPIO_Write_Toggle_Data_Bit</i>	13
2.2.5 <i>GPIO_Read_Output_Data_Bit</i>	13
2.2.6 <i>GPIO_Read_Input_Data</i>	13
2.2.7 <i>GPIO_Write_Output_Data</i>	14
2.2.8 <i>GPIO_Write_Toggle_Data</i>	14
2.2.9 <i>GPIO_Read_Output_Data</i>	14
2.2.10 <i>GPIO_Fout0</i>	14
2.2.11 <i>GPIO_Fout1</i>	15
3 CLK	16
3.1 CLK 寄存器结构.....	16
3.2 CLK 库函数.....	16
3.2.1 <i>CLK_init</i>	16
3.2.2 <i>CLK_change</i>	17
3.2.3 <i>CLK_Set_Peripheral_Clk</i>	17
3.2.4 <i>CLK_Clr_Peripheral_Clk</i>	18
4 CRC	20
4.1 CRC 寄存器结构.....	20
4.2 CRC 函数库.....	20
4.2.1 <i>CalCRC16_CCITT</i>	20
4.2.2 <i>BitChange</i>	20
5 EXIT	22
5.1 EXIT 寄存器结构.....	22
5.2 EXIT 函数库.....	22
5.2.1 <i>EXIT_init</i>	22
5.2.2 <i>EXIT_Close</i>	23



5.2.3	<i>EXIT_Select_Edge</i>	24
6	I2C	26
6.1	I2C 寄存器结构	26
6.2	I2C 函数库	26
6.2.1	<i>Init_I2c</i>	26
6.2.2	<i>I2C_Write_Bottom</i>	26
6.2.3	<i>I2C_Read_Bottom</i>	27
7	SPI	28
7.1	SPI 寄存器结构	28
7.2	SPI 函数库	28
7.2.1	<i>SPI_init</i>	28
7.2.2	<i>SPI_rw_byte</i>	29
7.2.3	<i>SPI_send_buf</i>	30
7.2.4	<i>SPI_recv_buf</i>	30
8	ADC	31
8.1	ADC 寄存器结构	31
8.2	ADC 函数库	31
8.2.1	<i>ADC_Init</i>	31
8.2.2	<i>ADC_Enable</i>	32
8.2.3	<i>ADC_Disable</i>	32
8.2.4	<i>ADC_Get_Done_Flag</i>	32
8.2.5	<i>ADC_Get_Value</i>	32
8.2.6	<i>ADC_Clear_Done_Flag</i>	33
8.2.7	<i>ADC_Result_Voltage_Convert</i>	33
8.2.8	<i>ADC_Result_Temperature_Convert</i>	33
9	FLASH 擦写/FLASH 模拟 EEPROM	34
9.1	FLASH 寄存器结构	34
9.2	FLASH 擦写/FLASH 模拟 EEPROM 库函数	34
9.2.1	<i>Flash_Erase_SECTOR</i>	34
9.2.2	函数 <i>Flash_Write_BYTE</i>	35
9.2.3	函数 <i>Flash_Write_SHORT</i>	35
9.2.4	函数 <i>Flash_Write_STRING</i>	35
9.2.5	函数 <i>Flash_Write_SECTOR</i>	35
9.2.6	函数 <i>Flash_Read_BYTE</i>	36
9.2.7	函数 <i>Flash_Read_SHORT</i>	36
9.2.8	函数 <i>Flash_Read_STRING</i>	36
9.2.9	函数 <i>Flash_Int_Cfg</i>	36



10	LPTIMER.....	38
10.1	LPTIMER 寄存器结构.....	38
10.2	LPTIMER 库函数	38
10.2.1	函数 <i>LPTIM_Init</i>	38
10.2.2	函数 <i>LPTIM_PWM</i>	39
10.2.3	函数 <i>LPTIM_Init_Ms_Interrupt</i>	39
10.2.4	函数 <i>LPTIM_Init_Count_Interrupt</i>	39
10.2.5	函数 <i>LPTIM_Init_Trigger_Interrupt</i>	40
10.2.6	函数 <i>LPTIM_Enable</i>	40
10.2.7	函数 <i>LPTIM_Disable</i>	40
11	定时器中断	41
11.1	定时器寄存器结构.....	41
11.2	定时器中断库函数.....	41
11.2.1	函数 <i>TIME_T0_Init_time_Interrupt</i>	41
11.2.2	函数 <i>TIME_T1_Init_time_Interrupt</i>	42
11.2.3	函数 <i>TIME_T2_Init_time_Interrupt</i>	42
12	PCA 捕获/比较.....	43
12.1	PCA 寄存器结构	43
12.2	PCA 捕获/比较库函数.....	43
12.2.1	函数 <i>PCA_PWM_Init</i>	44
12.2.2	函数 <i>PCA_Capture_Init</i>	44
12.2.3	函数 <i>PCA_Capture_Interrupt_Enable</i>	44
12.2.4	函数 <i>PCA_Capture_Interrupt_Disable</i>	45
12.2.5	函数 <i>PCA_Compare_Init</i>	45
12.2.6	函数 <i>PCA_Compare_Interrupt_Enable</i>	45
12.2.7	函数 <i>PCA_Compare_Interrupt_Disable</i>	45
12.2.8	函数 <i>PCA_Compare_Toggle_Out_Enable</i>	46
12.2.9	函数 <i>PCA_Cap_Cmp_Interrupt_Flag_Clear</i>	46
12.2.10	函数 <i>PCA_Overflow_Init</i>	46
12.2.11	函数 <i>PCA_Overflow_Interrupt_Enable</i>	46
12.2.12	函数 <i>PCA_Overflow_Interrupt_Disable</i>	47
12.2.13	函数 <i>PCA_Overflow_Interrupt_Flag_Clear</i>	47
12.2.14	函数 <i>PCA_Enable</i>	47
12.2.15	函数 <i>PCA_Disable</i>	47
13	SVD	48
13.1	SVD 寄存器结构	48
13.2	SVD 库函数	48



13.2.1	函数 SVD_Init.....	48
13.2.2	函数 SVD_Open.....	48
13.2.3	函数 SVD_Close.....	49
13.2.4	函数 SSVD_Software_Debounce_Judgement.....	49
13.2.5	函数 SVD_Software_Debounce_Judgement.....	49
14	LCD.....	50
14.1	LCD 寄存器结构.....	50
14.2	LCD 库函数.....	51
14.2.1	函数 LCD_Register_Init.....	51
14.2.2	函数 LCD_IO_Init.....	51
14.2.3	函数 LCD_Flicker_Enable.....	53
14.2.4	函数 LCD_Flicker_Disable.....	53
14.2.5	函数 LCD_Enable.....	53
14.2.6	函数 LCD_Disable.....	53
14.2.7	函数 Lcd_Display_FullScreen.....	54
14.2.8	函数 Lcd_Display_ClrScreen.....	54
15	UART.....	55
15.1	UART 寄存器结构.....	55
15.2	UART 库函数.....	55
15.2.1	函数 UART_Init.....	56
15.2.2	函数 UART_TX_Interrupt_Enable.....	56
15.2.3	函数 UART_RX_Interrupt_Enable.....	56
15.2.4	函数 UART_TX_Interrupt_Disable.....	57
15.2.5	函数 UART_RX_Interrupt_Disable.....	57
15.2.6	函数 UART_TX_Interrupt_Clear_Flag.....	57
15.2.7	函数 UART_RX_Interrupt_Clear_Flag.....	57
15.2.8	函数 UART_RX_Enable.....	58
15.2.9	函数 UART_RX_Disable.....	58
15.2.10	函数 UART_TX_Enable.....	58
15.2.11	函数 UART_TX_Disable.....	58
15.2.12	函数 UART_ReceiveData.....	59
15.2.13	函数 UART_SendData.....	59
15.2.14	函数 UART_Baudrate_Set.....	59
15.2.15	函数 UART_Ir_Init.....	59
15.2.16	函数 UART_Ir_Enable.....	60
15.2.17	函数 UART_Ir_Disbale.....	60
16	RTC.....	61
16.1	RTC 寄存器结构.....	61



16.2	RTC 库函数.....	61
16.2.1	函数 <i>RTC_Interrupt_Enable</i>	62
16.2.2	函数 <i>RTC_Interrupt_Disable</i>	62
16.2.3	函数 <i>RTC_Interrupt_Flag_Clear</i>	63
16.2.4	函数 <i>RTC_Write_Enable</i>	64
16.2.5	函数 <i>RTC_Write_Disable</i>	64
16.2.6	函数 <i>RTC_Byte_Is_BCD</i>	64
16.2.7	函数 <i>RTC_Judge_Clock</i>	64
16.2.8	函数 <i>RTC_Read_Rtc</i>	65
16.2.9	函数 <i>RTC_Set_Rtc</i>	65
16.2.10	函数 <i>RTC_Syn_Clock</i>	65
16.2.11	函数 <i>RTC_Read_Sysclock</i>	65
16.2.12	函数 <i>RTC_Fout_Select</i>	66
16.2.13	函数 <i>RTC_Set_Alarm</i>	66
16.2.14	函数 <i>RTC_Adjust_Ltbc</i>	66
17	版本说明	67

1 工程使用说明

所有程序的工程目录结构如下图：

名称	修改日期	类型
 c251	2018/11/22 15:44	文件夹
 keil37	2018/12/19 13:49	文件夹

- c251 文件夹下存放了工程使用到的 MCU 相关的底层模块驱动程序，用户程序，仿真文件，包含程序.C 文件和.H 头文件；
- keil37 文件夹下存放了 Keil 的工程配置文件，仿真设置,工程输出文件等；

2 GPIO

2.1 GPIO 寄存器结构:

寄存器	描述
PAPPEN	GPIOx 输出驱动使能寄存器
PBPPEN	
PCPPEN	
PDPPEN	
PEPPEN	
PFPPEN	
PGPPEN	
PH0PPEN	
PAPUEN	GPIOx 上拉使能寄存器
PBPUEN	
PCPUEN	
PDPUEEN	
PEPUEN	
PFPUEEN	
PGPUEN	
PH0PUEN	
PBODEN	GPIO 开漏使能寄存器
PEODEN	
PAFCR1	GPIOx 功能控制寄存器
PBFCR1	
PCFCR1	
PDFCR1	
PEFCR1	
PFFCR1	
PGFCR1	
PHFCR1	
PAFCR2	
PBFCR2	
PCFCR2	
PDFCR2	
PEFCR2	



PFFCR2	
PGFCR2	
PHFCR2	
PADATA	GPIOx 输出数据寄存器
PBDATA	
PCDATA	
PDDATA	
PEDATA	
PFDATA	
PGDATA	
PH0DATA	
PADIN	GPIOx 输入数据寄存器
PBDIN	
PCDIN	
PDDIN	
PEDIN	
PFDIN	
PGDIN	
PH0DIN	
AFSELA	GPIOx 替换功能选择寄存器
AFSELB	
AFSELC	
AFSELD	
AFSELE	
AFSELF	
AFSELG	
GPIO_EXTI_SEL0	GPIOx 中断选择寄存器
GPIO_EXTI_SEL1	
GPIO_EXTI_SEL2	
GPIO_EXTI_SEL3	
GPIO_EXTIL_ES0	GPIO 中断边沿选择寄存器
GPIO_EXTIL_ES1	
GPIO_EXTIH_ES0	
GPIO_EXTIH_ES1	
EXTILIF	GPIO 中断标志寄存器
EXTIHIF	
FOUT0_SEL	FOUT0 输出选择寄存器



FOUT1_SEL	FOUT1 输出选择寄存器
PAOS	GPIOx 输出状态寄存器
PBOS	
PCOS	
PDOS	
PEOS	
PFOS	
PGOS	
PH0OS	

2.2 GPIO 函数库:

序号	函数名	描述
1	GPIO_Init	GPIO 配置 pin 引脚功能
2	GPIO_Read_Input_Data_Bit	GPIO 读取输入 pin 信号
3	GPIO_Write_Output_Data_Bit	GPIO 写输入 pin 信号
4	GPIO_Write_Toggle_Data_Bit	GPIO 反转输出 pin 信号
5	GPIO_Read_Output_Data_Bit	GPIO 读取输出 pin 信号
6	GPIO_Read_Input_Data	GPIO 读取 port 信号
7	GPIO_Write_Output_Data	GPIO 写输入 port 信号
8	GPIO_Write_Toggle_Data	GPIO 翻转输出 port 信号
9	GPIO_Read_Output_Data	GPIO 读取输出 port 信号
10	GPIO_Fout0	GPIO FOUT0 输出频率
11	GPIO_Fout1	GPIO FOUT1 输出频率

2.2.1 GPIO_init

函数名	GPIO_init
函数原型	void GPIO_Init(unsigned char port ,unsigned char pin , unsigned char mode , unsigned char select)
功能描述	GPIO PIN 初始化
输入参数 1	port 端口： PORTA to PORTH
输入参数 2	pin 端口内引脚： PIN0 to PIN7
输入参数 3	<div>mode 端口功能：</div> <div><div>输入浮空： GPIO_MODE_IN_FLOATING</div><div>输入上拉：</div></div>



	GPIO_MODE_IN_PULL_UP 开漏输出（仅支持 PORTB 和 PORTE）： GPIO_MODE_OUT_OPEN_DRAIN 开漏输出内部上拉（仅支持 PORTB 与 PORTE）： GPIO_MODE_OUT_OPEN_DRAIN_PULL_UP 推挽输出： GPIO_MODE_OUT_PUSH_PULL 管脚不使用： GPIO_MODE_IO_OFF 模拟： GPIO_MODE_ANALOG 特殊功能选择： GPIO_MODE_ALTERNATE_FUNCTION 特殊功能选择内部上拉： GPIO_MODE_ALTERNATE_FUNCTION_PULL_UP
输入参数 4	select 功能 (mode 为特殊功能选择时有效): 特殊功能选择 0: MODE_ALTERNATE_SELECT_0 特殊功能选择 1: MODE_ALTERNATE_SELECT_1 特殊功能不选择（保持原样）： MODE_ALTERNATE_SELECT_NULL
输出参数	无
返回值	无

2.2.2 GPIO_Read_Input_Data_Bit

函数名	GPIO_Read_Input_Data_Bit
函数原型	unsigned char GPIO_Read_Input_Data_Bit(unsigned char port , unsigned char pin)
功能描述	GPIO 读取输入 pin 信号
输入参数 1	port 端口: PORTA to PORTH
输入参数 2	pin 端口内引脚: PIN0 to PIN7
输出参数	无
返回值	0 低电平 1 高电平

2.2.3 GPIO_Write_Output_Data_Bit

函数名	GPIO_Write_Output_Data_Bit
-----	----------------------------



函数原型	void GPIO_Write_Output_Data_Bit(unsigned char port , unsigned char pin ,unsigned char bitval)
功能描述	GPIO 写输出 pin 信号
输入参数 1	port 端口: PORTA to PORTH
输入参数 2	pin 端口内引脚: PIN0 to PIN7
输入参数 3	Bitval: 1 高 0 低
输出参数	无
返回值	无

2.2.4 GPIO_Write_Toggle_Data_Bit

函数名	GPIO_Write_Toggle_Data_Bit
函数原型	void GPIO_Write_Toggle_Data_Bit(unsigned char port , unsigned char pin)
功能描述	GPIO 反转输出 pin 信号
输入参数 1	port 端口: PORTA to PORTH
输入参数 2	pin 端口内引脚: PIN0 to PIN7
输出参数	无
返回值	无

2.2.5 GPIO_Read_Output_Data_Bit

函数名	GPIO_Read_Output_Data_Bit
函数原型	unsigned char GPIO_Read_Output_Data_Bit(unsigned char port , unsigned char pin);
功能描述	GPIO 读取输入 port 信号
输入参数 1	port 端口: PORTA to PORTH
输入参数 2	pin 端口内引脚: PIN0 to PIN7
输出参数	无
返回值	0 低电平 1 高电平

2.2.6 GPIO_Read_Input_Data

函数名	GPIO_Read_Input_Data
函数原型	unsigned char GPIO_Read_Input_Data(unsigned char port)
功能描述	GPIO 读取输入 port 信号
输入参数 1	port 端口: PORTA to PORTH



输出参数	无
返回值	端口电平

2.2.7 GPIO_Write_Output_Data

函数名	GPIO_Write_Output_Data
函数原型	void GPIO_Write_Output_Data(unsigned char port , unsigned char portval)
功能描述	GPIO 写输出 port 信号
输入参数 1	port 端口: PORTA to PORTH
输入参数 2	Portval 端口信号:
输出参数	无
返回值	无

2.2.8 GPIO_Write_Toggle_Data

函数名	GPIO_Write_Toggle_Data
函数原型	void GPIO_Write_Toggle_Data(unsigned char port)
功能描述	GPIO 反转输出 port 信号
输入参数 1	port 端口: PORTA to PORTH
输出参数	无
返回值	无

2.2.9 GPIO_Read_Output_Data

函数名	GPIO_Read_Output_Data
函数原型	unsigned char GPIO_Read_Output_Data(unsigned char port)
功能描述	GPIO 读取输出 port 端口
输入参数 1	port 端口: PORTA to PORTH
输出参数	无
返回值	端口电平

2.2.10 GPIO_Fout0

函数名	GPIO_Fout0
函数原型	void GPIO_Fout0(unsigned char select)
功能描述	GPIO FOUT0 输出频率
输入参数 1	select



	0: XTLF 1: RCLP 2: RCHF/64 3: LSCLK 4: CORECLK/64 5: RTCTM 6: PLLO/64 7: RTCCLK64Hz
输出参数	无
返回值	无

2.2.11 GPIO_Fout1

函数名	GPIO_Fout1
函数原型	void GPIO_Fout1(unsigned char select)
功能描述	GPIO FOUT1 输出频率
输入参数 1	select 0: LSCLK 1: PLLO 2: PLLO/4 3: RCHF
输出参数	无
返回值	无

3 CLK

3.1 CLK 寄存器结构

寄存器	描述
FDETIE	停振检测中断使能寄存器
FDETIF	停振检测中断标志寄存器
HSCLKPRES	高速时钟分频配置寄存器
MCLKSEL	系统时钟选择寄存器
LSCLKSEL	低速时钟选择寄存器
CLKSRC	时钟源控制寄存器
RCHFADJ	RCHF 调校寄存器
XTLFIPW	XTLF 电流控制寄存器
RCLPTRIM	RCLP 频率控制寄存器
PERCKEN0	外设时钟控制寄存器 0
PERCKEN1	外设时钟控制寄存器 1
PERCKEN2	外设时钟控制寄存器 2
PLLDBH	PLL 分频配置高位寄存器
PLLDBL	PLL 分频配置低位寄存器
FLSWAIT	Flash 取指等待寄存器

3.2 CLK 库函数

序号	函数名	描述
1	CLK_init	CLK 初始化
2	CLK_change	CLK 切换
3	CLK_Set_Peripheral_Clk	CLK 设置外置时钟
4	CLK_Clr_Peripheral_Clk	CLK 关闭外设时钟

3.2.1 CLK_init

函数名	CLK_init
函数原型	void CLK_Init(void)
功能描述	CLK 初始化
输入参数 1	无



输出参数	无
返回值	无

3.2.2 CLK_change

函数名	CLK_change
函数原型	void CLK_change(unsigned char clk)
功能描述	CLK 切换主频
输入参数 1	clk 主频 clk 0: 8M rchf 1: 16M rchf 2: 24M rchf 3: 16M pll 4: rclp 5: xtlf
输出参数	无
返回值	无

3.2.3 CLK_Set_Peripheral_Clk

函数名	CLK_Set_Peripheral_Clk
函数原型	void CLK_Set_Peripheral_Clk(unsigned char peripheral)
功能描述	CLK 设置外设时钟
输入参数 1	Peripheral 外设 CTRL0: PERIPHERAL_LCD_CLK PERIPHERAL_RAMBIST_CLK PERIPHERAL_FLSC_CLK PERIPHERAL_DMA_CLK PERIPHERAL_ET34_CLK PERIPHERAL_ET2_CLK PERIPHERAL_ET1_CLK PERIPHERAL_CRC_CLK CTRL1: PERIPHERAL_UART0_CLK



	PERIPHERAL_UART1_CLK PERIPHERAL_UART2_CLK PERIPHERAL_UART3_CLK PERIPHERAL_SPI_CLK PERIPHERAL_I2C_CLK PERIPHERAL_U7816_CLK PERIPHERAL_UART_PER_CLK CTRL2 PERIPHERAL_ADC_1M_CLK PERIPHERAL_RTC_CLK PERIPHERAL_ANAC_CLK PERIPHERAL_WDT_CLK PERIPHERAL_PDC_CLK PERIPHERAL_PCA_CLK PERIPHERAL_ADC_CLK PERIPHERAL_LPTIM_CLK PERIPHERAL_ALL_CLK
输出参数	无
返回值	无

3.2.4 CLK_Clr_Peripheral_Clk

函数名	CLK_Clr_Peripheral_Clk
函数原型	void CLK_Clr_Peripheral_Clk(unsigned char peripheral)
功能描述	CLK 关闭外设时钟
输入参数 1	Peripheral 外设 CTRL0: PERIPHERAL_LCD_CLK PERIPHERAL_RAMBIST_CLK PERIPHERAL_FLSC_CLK PERIPHERAL_DMA_CLK PERIPHERAL_ET34_CLK PERIPHERAL_ET2_CLK PERIPHERAL_ET1_CLK PERIPHERAL_CRC_CLK



	<p>CTRL1:</p> <p>PERIPHERAL_UART0_CLK</p> <p>PERIPHERAL_UART1_CLK</p> <p>PERIPHERAL_UART2_CLK</p> <p>PERIPHERAL_UART3_CLK</p> <p>PERIPHERAL_SPI_CLK</p> <p>PERIPHERAL_I2C_CLK</p> <p>PERIPHERAL_U7816_CLK</p> <p>PERIPHERAL_UART_PER_CLK</p> <p>CTRL2</p> <p>PERIPHERAL_ADC_1M_CLK</p> <p>PERIPHERAL_RTC_CLK</p> <p>PERIPHERAL_ANAC_CLK</p> <p>PERIPHERAL_WDT_CLK</p> <p>PERIPHERAL_PDC_CLK</p> <p>PERIPHERAL_PCA_CLK</p> <p>PERIPHERAL_ADC_CLK</p> <p>PERIPHERAL_LPTIM_CLK</p> <p>PERIPHERAL_ALL_CLK</p>
输出参数	无
返回值	无

4 CRC

4.1 CRC 寄存器结构

寄存器	描述
CRC_DRL	CRC 数据寄存器低位
CRC_DRH	CRC 数据寄存器高位
CRC_CR	CRC 控制状态寄存器
CRC_CALL	CRC 运算寄存器低位
CRC_CALH	CRC 运算寄存器高位
CRC_XORL	CRC 异或输出寄存器低位
CRC_XORH	CRC 异或输出寄存器高位
CRC_FLSCRC	CRC Flash 校验寄存器

4.2 CRC 函数库

序号	函数名	描述
1	CalCRC16_CCITT	计算 CRC16
2	BitChange	16bit 高低反转

4.2.1 CalCRC16_CCITT

函数名	CalCRC16_CCITT
函数原型	extern unsigned int CalCRC16_CCITT(unsigned int init, unsigned char *dataIn, unsigned int len)
功能描述	计算 CRC16
输入参数 1	Init 初始参数: 0x0000
输入参数 2	DataIn 数据 buffer: 如 {0x11,0x22,0x33,0x44}
输入参数 3	Len 数据长度: 如 4
输出参数	无
返回值	计算结果高低字节倒序输出

4.2.2 BitChange

函数名	BitChange
函数原型	extern unsigned int BitChange(unsigned int dataIn)



功能描述	16bit 高低反转
输入参数 1	us_Datain: 需要反转的 U16
输出参数	无
返回值	反转后的 16bit

5 EXIT

5.1 EXIT 寄存器结构

寄存器	描述
GPIO_EXTI_SEL0	GPIOx 中断选择寄存器
GPIO_EXTI_SEL1	
GPIO_EXTI_SEL2	
GPIO_EXTI_SEL3	
GPIO_EXTIL_ES0	GPIO 中断边沿选择寄存器
GPIO_EXTIL_ES1	
GPIO_EXTIH_ES0	
GPIO_EXTIH_ES1	
EXTILIF	GPIO 中断标志寄存器
EXTIHIF	

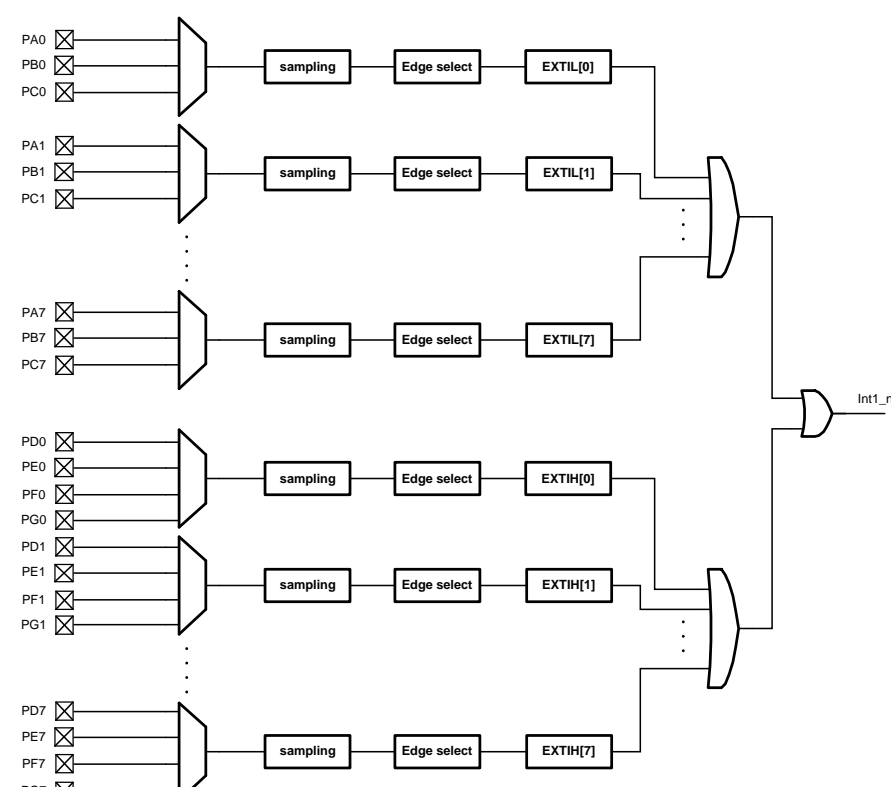
5.2 EXIT 函数库

序号	函数名	描述
1	EXIT_Init	外部中断初始化
2	EXIT_Close	外部中断关闭
3	EXIT_Select_Edge	外部中断边沿选择

5.2.1 EXIT_init

函数名	EXIT_init
函数原型	void EXIT_Init(unsigned char port,unsigned char pin,unsigned char edge)
功能描述	外部中断初始化
输入参数 1	port 端口: PORTA to PORTG
输入参数 2	pin 端口内引脚: PIN0 to PIN7
注	FM3316 的 7 组 GPIO (A~G) 最多可以产生 16 个外部引脚中断。7 组 GPIO 被分为 2 部分, 分别为 GPIOA/B/C 和 GPIOD/E/F/G, 每部分分别可以选择 8 个引脚产生独立的外部中断, 由 4 组控制寄存器 GPIOEXTI_SELx(x=0~3)进行配置, 具体参见寄存器说明。其中 GPIOA/B/C 选出的 8bit 用于产生 EXTIL[7:0], GPIOD/E/F/G 选出的 8bit 用于产生 EXTIH[7:0], 最终汇总后接入 CPU 的 int1_n 中断输入。软件可以查询 INTC 模块的 EXTIL 和 EXTIH 寄存器来确认产生中断的



	<p>具体引脚。</p> 
输入参数 3	Edge 触发沿： EXTI_TRIGGER_RISING EXTI_TRIGGER_RISING_FALLING EXTI_TRIGGER_DISABLE EXTI_TRIGGER_FALLING
输出参数	无
返回值	无

5.2.2 EXIT_Close

函数名	EXIT_Close
函数原型	void EXIT_Close(unsigned char port,unsigned char pin)
功能描述	外部中断关闭
输入参数 1	port 端口： PORTA to PORTG
输入参数 2	pin 端口内引脚： PIN0 to PIN7
注	FM3316 的 7 组 GPIO (A~G) 最多可以产生 16 个外部引脚中断。7 组 GPIO 被分为 2 部分，分别为 GPIOA/B/C 和 GPIOD/E/F/G，每部分分别可以选择 8 个引脚产生独立的外部中断，由 4 组控制寄存器 GPIOEXTI_SELx(x=0~3)进行配置，具体参见寄存器说明。其中 GPIOA/B/C 选出的 8bit 用于产生 EXTIL[7:0]，GPIOD/E/F/G 选出的 8bit 用于产生 EXTIH[7:0]，最终汇总后接入 CPU 的 int1_n 中断输入。软件可以查询 INTC 模块的 EXTIL 和 EXTIH 寄存器来确认产生中断的



	<p>具体引脚。</p> <p>PA0 <input type="checkbox"/> PA1 <input type="checkbox"/> PA7 <input type="checkbox"/> PB0 <input type="checkbox"/> PB1 <input type="checkbox"/> PB7 <input type="checkbox"/> PC0 <input type="checkbox"/> PC1 <input type="checkbox"/> PC7 <input type="checkbox"/> PD0 <input type="checkbox"/> PE0 <input type="checkbox"/> PF0 <input type="checkbox"/> PG0 <input type="checkbox"/> PD1 <input type="checkbox"/> PE1 <input type="checkbox"/> PF1 <input type="checkbox"/> PG1 <input type="checkbox"/> PD7 <input type="checkbox"/> PE7 <input type="checkbox"/> PF7 <input type="checkbox"/> PG7 <input type="checkbox"/></p> <p>sampling Edge select EXTIL[0] EXTIL[1] ... EXTIL[7] EXTIH[0] EXTIH[1] ... EXTIH[7]</p> <p>Int1_n</p>
输出参数	无
返回值	无

5.2.3 EXIT_Select_Edge

函数名	EXIT_Select_Edge
函数原型	void EXIT_Select_Edge(unsigned char port,unsigned char pin,unsigned char edge)
功能描述	外部中断边沿选择
输入参数 1	port 端口：PORTA to PORTH
输入参数 2	pin 端口内引脚： PIN0 to PIN7
注	FM3316 的 7 组 GPIO (A~G) 最多可以产生 16 个外部引脚中断。7 组 GPIO 被分为 2 部分，分别为 GPIOA/B/C 和 GPIOD/E/F/G，每部分分别可以选择 8 个引脚产生独立的外部中断，由 4 组控制寄存器 GPIOEXTI_SELx(x=0~3)进行配置，具体参见寄存器说明。其中 GPIOA/B/C 选出的 8bit 用于产生 EXTIL[7:0]，GPIOD/E/F/G 选出的 8bit 用于产生 EXTIH[7:0]，最终汇总后接入 CPU 的 int1_n 中断输入。软件可以查询 INTC 模块的 EXTIL 和 EXTIH 寄存器来确认产生中断的具体引脚。



输入参数 3	Edge 触发沿: EXTI_TRIGGER_RISING EXTI_TRIGGER_RISING_FALLING EXTI_TRIGGER_DISABLE EXTI_TRIGGER_FALLING
输出参数	无
返回值	无

6 I2C

6.1 I2C 寄存器结构

寄存器	描述
SSPCON	SSP 控制寄存器
SSPSTAT	SSP 状态寄存器
SSPBRG	波特率设置寄存器
SSPBUF	收发缓冲寄存器
SSPIR	中断寄存器

6.2 I2C 函数库

序号	函数名	描述
1	Init_I2C	初始化 I2C 配置
2	I2C_Write_Bottom	I2C 底层写
3	I2C_Read_Bottom	I2C 底层读

6.2.1 Init_I2c

函数名	Init_I2c
函数原型	extern void Init_I2C(unsigned char Ch, unsigned int BaudRate, unsigned char Fmclk)
功能描述	初始化 I2C 配置，目前不支持中断，不支持从模式
输入参数 1	通道： Ch0: 通道 PB6&7 Ch1: 通道 PE5&6
输入参数 2	通信波特率: xx_KHz
输入参数 3	系统时钟频率: xx_MHZ
输出参数	无
返回值	无

6.2.2 I2C_Write_Bottom

函数名	I2C_Write_Bottom
函数原型	extern unsigned char I2C_Write_Bottom(unsigned char DeviceAddr,



	unsigned int OpAddr, unsigned char OpAddrMode, unsigned char *Buf, unsigned char h)
功能描述	I2C 底层写
输入参数 1	DeviceAddr: 器件地址
输入参数 2	OpAddr: 操作地址
输入参数 3	OpAddrMode 操作地址长度 0:1 字节, 其他: 2 字节
输入参数 4	待写数据缓冲区地址
输入参数 5	写入长度
输出参数	无
返回值	0: 成功 1: 失败

6.2.3 I2C_Read_Bottom

函数名	I2C_Read_Bottom
函数原型	extern unsigned char I2C_Read_Bottom(unsigned char DeviceAddr, unsigned int OpAddr, unsigned char OpAddrMode, unsigned char *Buf, unsigned char h)
功能描述	I2C 底层读
输入参数 1	DeviceAddr: 器件地址
输入参数 2	OpAddr: 操作地址
输入参数 3	OpAddrMode 操作地址长度 0:1 字节, 其他: 2 字节
输入参数 4	待写数据缓冲区地址
输入参数 5	写入长度
输出参数	无
返回值	0: 成功 1: 失败

7 SPI

7.1 SPI 寄存器结构

寄存器	描述
SPCR1	控制寄存器 1
SPCR2	控制寄存器 2
SPCR3	控制寄存器 3
SPCR4	控制寄存器 4
SPIIE	中断使能寄存器
SPSR	状态寄存器
TXBUF	写数据 buffer
RXBUF	接受数据 buffer

7.2 SPI 函数库

序号	函数名	描述
1	SPI_init	SPI 寄存器初始化
2	SPI_rw_byte	SPI 收发一个字节
3	SPI_send_buf	SPI 发送一串字节
4	SPI_recv_buf	SPI 接受一串字节

7.2.1 SPI_init

函数名	SPI_init
函数原型	extern void SPI_init(SPI_INIT_STRU *init)
功能描述	SPI 寄存器初始化
输入参数 1	<pre>typedef struct { unsigned char SPI_BaudRatePrescaler;//波特率 unsigned char SPI_Mode;//主从 unsigned char SPI_XSBMode;//MSB or LSB 大小端 unsigned char SPI_CPOL;//时钟极性 unsigned char SPI_CPHA; //时钟相位 unsigned char SPI_SSN; //SSN 软件控制还是硬件控制</pre>



	<pre> }SPI_INIT_STRU; BaudRatePrescaler: SPI_BAUDRATE_2 //fpclk/2 SPI_BAUDRATE_4 //fpclk/4 SPI_BAUDRATE_8 //fpclk/8 SPI_BAUDRATE_16 //fpclk/16 SPI_BAUDRATE_32 //fpclk/32 SPI_BAUDRATE_64 //fpclk/64 SPI_BAUDRATE_128 //fpclk/128 SPI_BAUDRATE_256 //fpclk/256 SPI_Mode: SPI_SLAVE_MODE //slave 模式 SPI_MASTER_MODE //master 模式 SPI_XSBMode: SPI_MSB_MODE //先发送 MSB SPI_LSB_MODE //先发送 LSB SPI_CPOL: SPI_CPOL_HIGH //stop 在高电平 SPI_CPOL_LOW //stop 在低电平 SPI_CPHA: SPI_CPHA_1Edge //第一个时钟边沿是第一个捕捉边沿 SPI_CPHA_2Edge //第二个时钟边沿是第一个捕捉边沿 SPI_SSN: SPI_SSN_MCU_HIGH //SSN 被软件写成 1 SPI_SSN_MCU_LOW //SSN 被软件写成 0 </pre>
输出参数	无
返回值	无

7.2.2 SPI_rw_byte

函数名	SPI_rw_byte
函数原型	extern unsigned char SPI_rw_byte(unsigned char dat)
功能描述	SPI 收发一个字节
输入参数 1	dat 发送字节
输出参数	无
返回值	无



7.2.3 SPI_send_buf

函数名	SPI_send_buf
函数原型	extern void SPI_send_buf (unsigned char* sendBuf,unsigned int length)
功能描述	SPI 发送一串字节
输入参数 1	SendBuf 发送 buf 指针
输入参数 2	Length 发送长度
输出参数	无
返回值	无

7.2.4 SPI_recv_buf

函数名	SPI_recv_buf
函数原型	extern void SPI_recv_buf(unsigned char *recvBuf,unsigned int length)
功能描述	SPI 发送一串字节
输入参数 1	SendBuf 发送 buf 指针
输入参数 2	Length 发送长度
输出参数	RecvBuf 收到字节放入 buf
返回值	无

8 ADC

8.1 ADC 寄存器结构

寄存器	描述
ADCCTL	ADC 控制寄存器
ADCTRIMH	ADC 调校寄存器高位
ADCTRIML	ADC 调校寄存器低位
ADCDATAH	ADC 输出寄存器高位
ADCDATAH	ADC 输出寄存器低位
ADCIF	ADC 中断标志寄存器
ANATESTSEL	ADC 通道控制寄存器

8.2 ADC 函数库

序号	函数名	描述
1	ADC_Init	初始化 ADC
2	ADC_Enable	打开 ADC
3	ADC_Disable	关闭 ADC
4	ADC_Get_Done_Flag	ADC 转换完成标志
5	ADC_Get_Value	获取转换好的 ADC 值
6	ADC_Clear_Done_Flag	清除转换完成标志
7	ADC Result Voltage Convert	ADC 结果转换为电压值
8	ADC Result Temperature Convert	ADC 结果转换为温度值

8.2.1 ADC_Init

函数名	ADC_Init
函数原型	void ADC_Init(unsigned char type , unsigned char irp_mode)
功能描述	初始化 ADC 模块
输入参数 1	Type: ADC_TEST_TEMPERATURE ADC_TEST_VDD15 ADC_TEST_AN0 ADC_TEST_AN1 ADC_TEST_AN2



	ADC_TEST_VDD
输入参数 2	Mode 是否打开中断: ADC_IE_EN ADC_IE_DIS
输出参数	无
返回值	无

8.2.2 ADC_Enable

函数名	ADC_Enable
函数原型	void ADC_Enable(void)
功能描述	使能 ADC 模块
输入参数	无
输出参数	无
返回值	无

8.2.3 ADC_Disable

函数名	ADC_Disable
函数原型	void ADC_Disable(void)
功能描述	关闭 ADC
输入参数	无
输出参数	无
返回值	无

8.2.4 ADC_Get_Done_Flag

函数名	ADC_Get_Done_Flag
函数原型	unsigned char ADC_Get_Done_Flag(void)
功能描述	ADC 转换完成标志位
输入参数	无
输出参数	无
返回值	0 未完成 1 已完成

8.2.5 ADC_Get_Value

函数名	ADC_Get_Value
-----	---------------



函数原型	unsigned int ADC_Get_Value(void)
功能描述	获取转换好的 ADC 值
输入参数	无
输出参数	无
返回值	ADC 值

8.2.6 ADC_Clear_Done_Flag

函数名	ADC_Clear_Done_Flag
函数原型	void ADC_Clear_Done_Flag(void)
功能描述	清楚转换完成标志
输入参数	无
输出参数	无
返回值	ADC 值

8.2.7 ADC_Result_Voltage_Convert

函数名	ADC Result Voltage Convert
函数原型	unsigned int ADC_Result_Voltage_Convert(unsigned int value)
功能描述	将 ADC 值转换成电压值 (单位: 0.1V)
输入参数	无
输出参数	无
返回值	电压值

8.2.8 ADC_Result_Temperature_Convert

函数名	ADC Result Temperature Convert
函数原型	unsigned int ADC_Result_Temperature_Convert(unsigned int value)
功能描述	将 ADC 值转换为温度值 (单位: 0.1°C)
输入参数	无
输出参数	无
返回值	温度值

9 FLASH 擦写/FLASH 模拟 EEPROM

9.1 FLASH 寄存器结构

寄存器	描述
ERCSR	Flash 擦除控制寄存器
PRCSR	Flash 编程控制寄存器
FLSKEY	FlashKey 寄存器
FLSCIE	Flash 中断使能寄存器
EPFLAG	Flash 擦写标志寄存器
RPARERR	RAM 校验错误标志寄存器
RPARIE	RAM 校验错误中断使能寄存器

9.2 FLASH 擦写/FLASH 模拟 EEPROM 库函数

函数名	描述
Flash_Erase_SECTOR	FLASH 扇区擦除
Flash_Write_BYTE	FLASH 单字节写入
Flash_Write_SHORT	FLASH 双字节写入
Flash_Write_STRING	FLASH 多字节写入
Flash_Write_SECTOR	FLASH 扇区写入
Flash_Read_BYTE	FLASH 单字节读取
Flash_Read_SHORT	FLASH 双字节读取
Flash_Read_STRING	FLASH 多字节读取
Flash_Int_Cfg	FLASH 擦写中断配置

9.2.1 Flash_Erase_SECTOR

函数名	Flash_Erase_SECTOR
函数原型	unsigned char Flash_Erase_SECTOR(unsigned long Addr)
功能描述	FLASH 扇区擦除
输入参数 1	Addr: 被擦扇区任意字节地址
输出参数 1	无
返回值	0: 成功; 其他: 发生错误



9.2.2 函数 Flash_Write_BYTE

函数名	Flash_Write_BYTE
函数原型	unsigned char Flash_Write_BYTE(unsigned long Addr, unsigned char WrByte)
功能描述	FLASH 单字节写入
输入参数 1	Addr: 编程地址
输入参数 2	WrByte: 编程数据
输出参数 1	无
返回值	0: 成功; 其他: 发生错误

9.2.3 函数 Flash_Write_SHORT

函数名	Flash_Write_SHORT
函数原型	unsigned char Flash_Write_SHORT(unsigned long Addr, unsigned short WrShrot)
功能描述	FLASH 双字节写入
输入参数 1	Addr: 编程地址
输入参数 2	WrShrot: 编程数据
输出参数 1	无
返回值	0: 成功; 其他: 发生错误

9.2.4 函数 Flash_Write_STRING

函数名	Flash_Write_STRING
函数原型	unsigned char Flash_Write_STRING(unsigned long Addr, unsigned char *Buf, unsigned int Len)
功能描述	FLASH 多字节写入
输入参数 1	Addr: 编程地址
输入参数 2	*Buf: 编程数据数组地址
输入参数 3	Len: 编程字节数
输出参数 1	无
返回值	0: 成功; 其他: 发生错误

9.2.5 函数 Flash_Write_SECTOR

函数名	Flash_Write_SECTOR
函数原型	unsigned char Flash_Write_SECTOR(unsigned long Addr)
功能描述	FLASH 扇区写入
输入参数 1	Addr: 编程地址



输出参数 1	无
返回值	0: 成功; 其他: 发生错误

9.2.6 函数 Flash_Read_BYTE

函数名	Flash_Read_BYTE
函数原型	unsigned char Flash_Read_BYTE(unsigned long Addr)
功能描述	FLASH 单字节读取
输入参数 1	Addr: 读取地址
输出参数 1	无
返回值	对应地址的 FLASH 内容

9.2.7 函数 Flash_Read_SHORT

函数名	Flash_Read_SHORT
函数原型	unsigned short Flash_Read_SHORT(unsigned long Addr)
功能描述	FLASH 双字节读取
输入参数 1	Addr: 读取地址
输出参数 1	无
返回值	对应地址的 FLASH 内容

9.2.8 函数 Flash_Read_STRING

函数名	Flash_Read_STRING
函数原型	unsigned char Flash_Read_STRING(unsigned long Addr, unsigned char *Buf, unsigned int Len)
功能描述	FLASH 多字节读取
输入参数 1	Addr: 读取地址
输入参数 2	*Buf: 读取缓冲区地址
输入参数 3	Len: 读取长度
输出参数 1	对应地址的 FLASH 内容
返回值	无

9.2.9 函数 Flash_Int_Cfg

函数名	Flash_Int_Cfg
函数原型	void Flash_Int_Cfg(unsigned char Mode)
功能描述	FLASH 擦写中断配置



输入参数 1	Mode: 0=关闭中断; 1=开启擦写完成中断; 2=开启擦写完成中断+错误中断
输出参数 1	无
返回值	无

10 LPTIMER

10.1 LPTIMER 寄存器结构

寄存器	描述
LPTCFG0	LPTIM 配置寄存器 0
LPTCFG1	LPTIM 配置寄存器 1
LPTCNTL	LPTIM 计数寄存器低位
LPTCNTH	LPTIM 计数寄存器高位
LPTCMPL	LPTIM 比较寄存器低位
LPTCMPH	LPTIM 比较寄存器高位
TARGETL	LPTIM 目标值寄存器低位
TARGETH	LPTIM 目标值寄存器高位
LPTIMIE	LPTIM 中断使能寄存器
LPTIMIF	LPTIM 中断标志寄存器
LPTIMCTRL	LPTIM 控制寄存器

10.2 LPTIMER 库函数

函数名	描述
LPTIM_Init	初始化 LPTIM (所有寄存器自行配置, 功能自行配置)
LPTIM_PWM	LLPTIM PWM 输出 (PB5 输出) 输出频率自行修改 示例使用 2KHz
LPTIM_Init_Ms_Interrupt	初始化 LPTIM 产生毫秒定时中断 (使用外部 32768 晶振)
LPTIM_Init_Count_Interrupt	初始化 LPTIM 作为外部异步脉冲计数比较中断(PB4 输入, 下降沿计数, 默认 5 个脉冲中断, 可自行修改)
LPTIM_Init_Trigger_Interrupt	初始化 LPTIM 作为 Trigger 脉冲触发中断(PB6 输入, 下降沿触发)
LPTIM_Enable	打开 LPTIM
LPTIM_Disable	关闭 LPTIM

10.2.1 函数 LPTIM_Init

函数名	LPTIM_Init
-----	------------



函数原型	void LPTIM_Init(void)
功能描述	初始化 LPTIM
输入参数 1	无
输出参数 1	无
返回值	无

10.2.2 函数 LPTIM_PWM

函数名	LPTIM_PWM
函数原型	void LPTIM_PWM(void)
功能描述	LLPTIM PWM 输出 (PB5 输出) 输出频率自行修改 示例使用 2KHz
输入参数 1	无
输出参数 1	无
返回值	无

10.2.3 函数 LPTIM_Init_Ms_Interrupt

函数名	LPTIM_Init_Ms_Interrupt
函数原型	unsigned char LPTIM_Init_Ms_Interrupt(unsigned int ms)
功能描述	初始化 LPTIM 产生毫秒定时中断 (使用外部 32768 晶振)
输入参数 1	ms: 毫秒, 数值范围 1-1000
输出参数 1	无
返回值	0: 正确; 1: 参数错误

10.2.4 函数 LPTIM_Init_Count_Interrupt

函数名	LPTIM_Init_Count_Interrupt
函数原型	void LPTIM_Init_Count_Interrupt(void)
功能描述	初始化 LPTIM 作为外部异步脉冲计数比较中断(PB4 输入, 下降沿计数, 默认 5 个脉冲中断, 可自行修改)
输入参数 1	无
输出参数 1	无
返回值	无



10.2.5 函数 LPTIM_Init_Trigger_Interrupt

函数名	LPTIM_Init_Trigger_Interrupt
函数原型	void LPTIM_Init_Trigger_Interrupt(void)
功能描述	初始化 LPTIM 作为 Trigger 脉冲触发中断(PB6 输入, 下降沿触发)
输入参数 1	无
输出参数 1	无
返回值	无

10.2.6 函数 LPTIM_Enable

函数名	LPTIM_Enable
函数原型	void LPTIM_Enable(void)
功能描述	打开 LPTIM
输入参数 1	无
输出参数 1	无
返回值	无

10.2.7 函数 LPTIM_Disable

函数名	LPTIM_Disable
函数原型	void LPTIM_Disable(void)
功能描述	关闭 LPTIM
输入参数 1	无
输出参数 1	无
返回值	无

11 定时器中断

11.1 定时器寄存器结构

寄存器	描述
TCON	Timer/Count 0 及 Timer/Count 1 控制
TMOD	Timer/Count 0 及 Timer/Count 1 模式控制
TL0	Timer/Count 0 低字节
TL1	Timer/Count 1 低字节
TH0	Timer/Count 0 高字节
TH1	Timer/Count 1 高字节
T2CON	Timer/Count 2 控制
T2MOD	Timer/Count 2 模式控制
RCAP2L	Timer2 重载/捕捉低字节
RCAP2H	Timer2 重载/捕捉高字节
TL2	Timer/Count 2 低字节
TH2	Timer/Count 2 高字节

11.2 定时器中断库函数

函数名	描述
TIME_T0_Init_time_Interrupt	初始化 T0 产生定时中断
TIME_T1_Init_time_Interrupt	初始化 T1 产生定时中断
TIME_T2_Init_time_Interrupt	初始化 T2 产生定时中断

11.2.1 函数 TIME_T0_Init_time_Interrupt

函数名	TIME_T0_Init_time_Interrupt
函数原型	void TIME_T0_Init_time_Interrupt(void)
功能描述	初始化 T0 产生定时中断
输入参数 1	
输出参数 1	无
返回值	无



11.2.2 函数 TIME_T1_Init_time_Interrupt

函数名	TIME_T1_Init_time_Interrupt
函数原型	void TIME_T1_Init_time_Interrupt(void)
功能描述	初始化 T1 产生定时中断
输入参数 1	
输出参数 1	无
返回值	无

11.2.3 函数 TIME_T2_Init_time_Interrupt

函数名	TIME_T2_Init_time_Interrupt
函数原型	void TIME_T2_Init_time_Interrupt(void)
功能描述	初始化 T2 产生定时中断
输入参数 1	
输出参数 1	无
返回值	无

12 PCA 捕获/比较

12.1 PCA 寄存器结构

寄存器	描述
CCON	PCA 控制寄存器
CMON	PCA 模式寄存器
CH	PCA 计数器高位寄存器
CL	PCA 计数器低位寄存器
CCAP0H	PCA 比较/捕捉模块 0 高位寄存器
CCAP1H	PCA 比较/捕捉模块 1 高位寄存器
CCAP2H	PCA 比较/捕捉模块 2 高位寄存器
CCAP3H	PCA 比较/捕捉模块 3 高位寄存器
CCAP4H	PCA 比较/捕捉模块 4 高位寄存器
CCAP0L	PCA 比较/捕捉模块 0 低位寄存器
CCAP1L	PCA 比较/捕捉模块 1 低位寄存器
CCAP2L	PCA 比较/捕捉模块 2 低位寄存器
CCAP3L	PCA 比较/捕捉模块 3 低位寄存器
CCAP4L	PCA 比较/捕捉模块 4 低位寄存器
CCAPM0	PCA 比较/捕捉模块 0 模式寄存器
CCAPM1	PCA 比较/捕捉模块 1 模式寄存器
CCAPM2	PCA 比较/捕捉模块 2 模式寄存器
CCAPM3	PCA 比较/捕捉模块 3 模式寄存器
CCAPM4	PCA 比较/捕捉模块 4 模式寄存器
CCAPO	PCA 输出寄存器
PWMPREL	PCA 脉冲宽度调制初值寄存器
ECISAMPLE	PCA 外部输入采样选择寄存器

12.2 PCA 捕获/比较库函数

函数名	描述
PCA_PWM_Init	PCA PWM 输出
PCA_Capture_Init	PCA capture 捕捉
PCA_Capture_Interrupt_Enable	打开 Capture 中断
PCA_Capture_Interrupt_Disable	关闭 Capture 中断

PCA_Compare_Init	PCA Compare 比较
PCA_Compare_Interrupt_Enable	打开 Compare 中断
PCA_Compare_Interrupt_Disable	关闭 Compare 中断
PCA_Compare_Toggle_Out_Enable	比较反转输出
PCA_Cap_Cmp_Interrupt_Flag_Clear	清除捕捉 比较 中断标志
PCA_Overflow_Init	PCA 溢出
PCA_Overflow_Interrupt_Enable	打开溢出 中断
PCA_Overflow_Interrupt_Disable	关闭溢出 中断
PCA_Overflow_Interrupt_Flag_Clear	清除捕捉 溢出 中断标志
PCA_Enable	打开 PCA
PCA_Disable	关闭 PCA

12.2.1 函数 PCA_PWM_Init

函数名	PCA_PWM_Init
函数原型	void PCA_PWM_Init(unsigned char PCAx)
功能描述	PCA PWM 输出
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.2 函数 PCA_Capture_Init

函数名	PCA_Capture_Init
函数原型	void PCA_Capture_Init(unsigned char PCAx)
功能描述	PCA capture 捕捉
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.3 函数 PCA_Capture_Interrupt_Enable

函数名	PCA_Capture_Interrupt_Enable
函数原型	void PCA_Capture_Interrupt_Enable(unsigned char PCAx)
功能描述	打开 Capture 中断
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无



12.2.4 函数 PCA_Capture_Interrupt_Disable

函数名	PCA_Capture_Interrupt_Disable
函数原型	void PCA_Capture_Interrupt_Disable(unsigned char PCAx)
功能描述	关闭 Capture 中断
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.5 函数 PCA_Compare_Init

函数名	PCA_Compare_Init
函数原型	void PCA_Compare_Init(unsigned char PCAx)
功能描述	PCA Compare 比较
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.6 函数 PCA_Compare_Interrupt_Enable

函数名	PCA_Compare_Interrupt_Enable
函数原型	void PCA_Compare_Interrupt_Enable(unsigned char PCAx)
功能描述	打开 Compare 中断
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.7 函数 PCA_Compare_Interrupt_Disable

函数名	PCA_Compare_Interrupt_Disable
函数原型	void PCA_Compare_Interrupt_Disable(unsigned char PCAx)
功能描述	关闭 Compare 中断
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无



12.2.8 函数 PCA_Compare_Toggle_Out_Enable

函数名	PCA_Compare_Toggle_Out_Enable
函数原型	void PCA_Compare_Toggle_Out_Enable(unsigned char PCAx)
功能描述	比较反转输出
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.9 函数 PCA_Cap_Cmp_Interrupt_Flag_Clear

函数名	PCA_Cap_Cmp_Interrupt_Flag_Clear
函数原型	void PCA_Cap_Cmp_Interrupt_Flag_Clear(unsigned char PCAx)
功能描述	清除捕捉 比较 中断标志
输入参数 1	PCAx: 可以是 PCA_0、PCA_1、PCA_2、PCA_3、PCA_4 的 PCA 序号
输出参数 1	无
返回值	无

12.2.10 函数 PCA_Overflow_Init

函数名	PCA_Overflow_Init
函数原型	void PCA_Overflow_Init(void)
功能描述	PCA 溢出
输入参数 1	无
输出参数 1	无
返回值	无

12.2.11 函数 PCA_Overflow_Interrupt_Enable

函数名	PCA_Overflow_Interrupt_Enable
函数原型	void PCA_Overflow_Interrupt_Enable(void)
功能描述	打开溢出 中断
输入参数 1	无
输出参数 1	无
返回值	无



12.2.12 函数 PCA_Overflow_Interrupt_Disable

函数名	PCA_Overflow_Interrupt_Disable
函数原型	void PCA_Overflow_Interrupt_Disable(void)
功能描述	关闭溢出 中断
输入参数 1	无
输出参数 1	无
返回值	无

12.2.13 函数 PCA_Overflow_Interrupt_Flag_Clear

函数名	PCA_Overflow_Interrupt_Flag_Clear
函数原型	void PCA_Overflow_Interrupt_Flag_Clear(void)
功能描述	清除捕捉 溢出 中断标志
输入参数 1	无
输出参数 1	无
返回值	无

12.2.14 函数 PCA_Enable

函数名	PCA_Enable
函数原型	void PCA_Enable(void)
功能描述	打开 PCA
输入参数 1	无
输出参数 1	无
返回值	无

12.2.15 函数 PCA_Disable

函数名	PCA_Disable
函数原型	void PCA_Disable(void)
功能描述	关闭 PCA
输入参数 1	无
输出参数 1	无
返回值	无

13 SVD

13.1 SVD 寄存器结构

寄存器	描述
SVDCTRL	SVD 控制寄存器
SVDSTAT	SVD 状态寄存器
SVDLPC	SVD 低功耗寄存器

13.2 SVD 库函数

函数名	描述
SVD_Init	SVD 初始化程序(初始化并开启)
SVD_Open	SVD 打开
SVD_Close	SVD 关闭
SVD_Software_Debounce_Judgement	SVD 软件去抖动判定低于判定电压(一般用于非中断的判定电压判断)
SVD_Software_Debounce_Judgement	SVD 软件去抖动判定高于判定电压(一般用于非中断的判定电压判断)

13.2.1 函数 SVD_Init

函数名	SVD_Init
函数原型	void SVD_Init(unsigned char voltage, unsigned char interval, unsigned char irp)
功能描述	SVD 初始化程序(初始化并开启)
输入参数 1	voltage: SVD_V180、SVD_V203、SVD_V226、SVD_V249、SVD_V272、SVD_V295、SVD_V319、SVD_V342、SVD_V365、SVD_V388、SVD_V411、SVD_V434、SVD_V457、SVD_V480、SVD_SVS0、SVD_SVS1, 用于判定电压
输入参数 2	interval: SVD_T4000、SVD_T2000、SVD_T1000、SVD_T0512、SVD_T0256、SVD_T0128、SVD_T0064、SVD_T0032, 间歇模式启动时间间隔
输入参数 3	irp: SVD_BOTH_IRP、SVD_LV_RE_IRP、SVD_LV_IRP、SVD_NONE_IRP, 中断模式
输出参数 1	无
返回值	无

13.2.2 函数 SVD_Open

函数名	SVD_Open
-----	----------



函数原型	void SVD_Open(unsigned char mode)
功能描述	SVD 打开
输入参数 1	mode: SVD_NORMAL_MODE、SVD_SLEEP_MODE, 工作模式
输出参数 1	无
返回值	无

13.2.3 函数 SVD_Close

函数名	SVD_Close
函数原型	void SVD_Close(void)
功能描述	SVD 打开
输入参数 1	无
输出参数 1	无
返回值	无

13.2.4 函数 SSVD_Software_Debounce_Judgement

函数名	SVD_Software_Debounce_Judgement
函数原型	unsigned char SVD_Software_Debounce_Judgement_Low(void)
功能描述	SVD 软件去抖动判定低于判定电压(一般用于非中断的判定电压判断)
输入参数 1	无
输出参数 1	无
返回值	0 : 一直低于判定电压, 1 : 有高于判定电压

13.2.5 函数 SVD_Software_Debounce_Judgement

函数名	SVD_Software_Debounce_Judgement
函数原型	unsigned char SVD_Software_Debounce_Judgement_High(void)
功能描述	SVD 软件去抖动判定高于判定电压(一般用于非中断的判定电压判断)
输入参数 1	无
输出参数 1	无
返回值	0: 一直高于判定电压, 1: 有低于判定电压

14 LCD

14.1 LCD 寄存器结构

寄存器	描述
DISPCTRL	显示控制寄存器
LCDTEST	显示测试控制寄存器
LCDDF	显示频率控制寄存器
TON	闪烁点亮时间寄存器
TOFF	闪烁熄灭时间寄存器
DISPIE	显示中断使能寄存器
DISPIF	显示中断标志寄存器
LCDSET	显示设置寄存器
ENMODE	显示驱动模式寄存器
DISPDATA0	显示数据缓存寄存器
DISPDATA1	
DISPDATA2	
DISPDATA3	
DISPDATA4	
DISPDATA5	
DISPDATA6	
DISPDATA7	
DISPDATA8	
DISPDATA9	
DISPDATA10	
DISPDATA11	
DISPDATA12	
DISPDATA13	
DISPDATA14	
DISPDATA15	
DISPDATA16	
DISPDATA17	
LCDBIAS	显示灰度设置寄存器
COMSEG_EN1	显示输出使能寄存器
COMSEG_EN2	
COMSEG_EN3	



COMSEG_EN4

14.2 LCD 库函数

函数名	描述
LCD_Register_Init	初始化 LCD 寄存器
LCD_IO_Init	IO 配置成 LCD 引脚
LCD_Flicker_Enable	LCD 闪烁
LCD_Flicker_Disable	LCD 闪烁关闭
LCD_Enable	LCD 打开
LCD_Disable	LCD 关闭
Lcd_Display_FullScreen	全屏显示
Lcd_Display_ClrScreen	清显示

14.2.1 函数 LCD_Register_Init

函数名	LCD_Register_Init
函数原型	void LCD_Register_Init(unsigned char type, unsigned char com)
功能描述	初始化 LCD 寄存器
输入参数 1	type: LCD_MODE_COUT LCD_MODE_RIN
输入参数 2	com: LCD_6COM LCD_4COM
输入参数 3	无
输出参数 1	无
返回值	无

14.2.2 函数 LCD_IO_Init

函数名	LCD_IO_Init
函数原型	LCD_IO_Init(unsigned char io)
功能描述	IO 配置成 LCD 引脚
输入参数 1	io: LCD_COM0 LCD_COM1 LCD_COM2



	LCD_COM3 LCD_COM4 LCD_COM5 LCD_VGIN1 LCD_VGIN2 LCD_VDISP LCD_VDISP LCD_VDISP LCD_SEG0 LCD_SEG1 LCD_SEG2 LCD_SEG3 LCD_SEG4 LCD_SEG5 LCD_SEG6 LCD_SEG7 LCD_SEG8 LCD_SEG9 LCD_SEG10 LCD_SEG11 LCD_SEG12 LCD_SEG13 LCD_SEG14 LCD_SEG15 LCD_SEG16 LCD_SEG17 LCD_SEG18 LCD_SEG19 LCD_SEG20 LCD_SEG21 LCD_SEG22 LCD_SEG23 LCD_SEG24 LCD_SEG25
输出参数 1	无
返回值	无



14.2.3 函数 LCD_Flicker_Enable

函数名	LCD_Flicker_Enable
函数原型	void LCD_Flicker_Enable(unsigned char ton_time,unsigned char toff_time)
功能描述	LCD 闪烁
输入参数 1	ton_time 闪烁亮时间 ton_time*0.25S
输出参数 1	toff_time 闪烁暗时间 toff_time*0.25S
返回值	无

14.2.4 函数 LCD_Flicker_Disable

函数名	LCD_Flicker_Disable
函数原型	void LCD_Flicker_Disable(void)
功能描述	LCD 闪烁关闭
输入参数 1	无
输出参数 1	无
返回值	无

14.2.5 函数 LCD_Enable

函数名	LCD_Enable
函数原型	void LCD_Enable(void)
功能描述	LCD 关闭
输入参数 1	无
输出参数 1	无
返回值	无

14.2.6 函数 LCD_Disable

函数名	LCD_Disable
函数原型	void LCD_Disable(void)
功能描述	LCD 关闭
输入参数 1	无
输出参数 1	无
返回值	无



14.2.7 函数 Lcd_Display_FullScreen

函数名	Lcd_Display_FullScreen
函数原型	void Lcd_Display_FullScreen(void)
功能描述	全屏显示
输入参数 1	无
输出参数 1	无
返回值	无

14.2.8 函数 Lcd_Display_ClrScreen

函数名	Lcd_Display_ClrScreen
函数原型	void Lcd_Display_ClrScreen(void)
功能描述	清显示
输入参数 1	无
输出参数 1	无
返回值	无

15 UART

15.1 UART 寄存器结构

寄存器	描述
UARTIE	UART 中断使能寄存器
UARTIF	UART 中断标志寄存器
TZBRGH	红外调制配置高位寄存器
TZBRGL	红外调制配置低位寄存器
RXSTA0~3	UART 接收状态寄存器
TXSTA0~3	UART 发送状态寄存器
RXREG0~3	UART 接收缓冲寄存器
TXREG0~3	UART 发送缓冲寄存器
SPBRGH0~3	UART 波特率发生器高位
SPBRGL0~3	UART 波特率发生器低位
TXFIFOSTA0~3	UART 发送缓冲区状态寄存器
RXFIFOSTA0~3	UART 接收缓冲区状态寄存器
RTXCON0~3	收发取反控制器

15.2 UART 库函数

函数名	描述
UART_Init	UART 初始化
UART_TX_Interrupt_Enable	UART 发送中断允许
UART_RX_Interrupt_Enable	UART 接收中断允许
UART_TX_Interrupt_Disable	UART 发送中断关闭
UART_RX_Interrupt_Disable	UART 接收中断关闭
UART_TX_Interrupt_Clear_Flag	UART 发送中断标志清除
UART_RX_Interrupt_Clear_Flag	UART 接收中断标志清除
UART_RX_Enable	UART 接收模块打开
UART_RX_Disable	UART 接收模块关闭
UART_TX_Enable	UART 接收模块关闭
UART_ReceiveData	接收数据
UART_SendData	发送数据
UART_Baudrate_Set	波特率设置



UART_Ir_Init	红外调制初始化
UART_Ir_Enable	红外调制打开
UART_Ir_Disbale	红外调制关闭

15.2.1 函数 UART_Init

函数名	UART_Init
函数原型	unsigned char UART_Init(unsigned char UARTx, UART_INIT_STRU *init)
功能描述	UART 初始化
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输入参数 2	*init: typedef struct { unsigned long UART_BaudRate; //波特率 最高 115200bps unsigned int UART_DataLength; //数据位长度 7, 8 或 9 (7 位与 9 位数据无校验位) unsigned int UART_StopBits; //停止位 1 或 2 unsigned int UART_Parity; //校验位 0 无校验, 1 奇校验, 2 偶校验 unsigned int UART_RxTxDfl; //电平取反(RS485 不同型号使用) 0 不取反, 1 发送取反, 2 接收取反, 3 接收发送都取反 } UART_INIT_STRU;
输出参数 1	无
返回值	无

15.2.2 函数 UART_TX_Interrupt_Enable

函数名	UART_TX_Interrupt_Enable
函数原型	void UART_TX_Interrupt_Enable(unsigned char UARTx)
功能描述	UART 发送中断允许
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.3 函数 UART_RX_Interrupt_Enable

函数名	UART_RX_Interrupt_Enable
函数原型	void UART_RX_Interrupt_Enable(unsigned char UARTx)
功能描述	UART 接收中断允许
输入参数 1	UARTx: UART0、UART1、UART2、UART3



输出参数 1	无
返回值	无

15.2.4 函数 UART_TX_Interrupt_Disable

函数名	UART_TX_Interrupt_Disable
函数原型	void UART_TX_Interrupt_Disable(unsigned char UARTx)
功能描述	UART 发送中断关闭
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.5 函数 UART_RX_Interrupt_Disable

函数名	UART_RX_Interrupt_Disable
函数原型	void UART_RX_Interrupt_Disable(unsigned char UARTx)
功能描述	UART 接收中断关闭
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.6 函数 UART_TX_Interrupt_Clear_Flag

函数名	UART_TX_Interrupt_Clear_Flag
函数原型	void UART_TX_Interrupt_Clear_Flag(unsigned char UARTx)
功能描述	UART 发送中断标志清除
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.7 函数 UART_RX_Interrupt_Clear_Flag

函数名	UART_RX_Interrupt_Clear_Flag
函数原型	void UART_RX_Interrupt_Clear_Flag(unsigned char UARTx)
功能描述	UART 接收中断标志清除
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无



15.2.8 函数 UART_RX_Enable

函数名	UART_RX_Enable
函数原型	void UART_RX_Enable(unsigned char UARTx)
功能描述	UART 接收模块打开
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.9 函数 UART_RX_Disable

函数名	UART_RX_Disable
函数原型	void UART_RX_Disable(unsigned char UARTx)
功能描述	UART 接收模块关闭
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.10 函数 UART_TX_Enable

函数名	UART_TX_Enable
函数原型	void UART_TX_Enable(unsigned char UARTx)
功能描述	UART 接收模块打开
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.11 函数 UART_TX_Disable

函数名	UART_TX_Disable
函数原型	void UART_TX_Disable(unsigned char UARTx)
功能描述	UART 接收模块关闭
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无



15.2.12 函数 UART_ReceiveData

函数名	UART_ReceiveData
函数原型	unsigned int UART_ReceiveData(unsigned char UARTx)
功能描述	接收数据
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	接收到的数据

15.2.13 函数 UART_SendData

函数名	UART_SendData
函数原型	void UART_SendData(unsigned char UARTx, unsigned int Data)
功能描述	发送数据
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输入参数 2	Data: 发送的数据
输出参数 1	无
返回值	无

15.2.14 函数 UART_Baudrate_Set

函数名	UART_Baudrate_Set
函数原型	unsigned char UART_Baudrate_Set(unsigned char UARTx, unsigned long Baudrate)
功能描述	波特率设置
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输入参数 2	Baudrate: 波特率(<=115200bps)
输出参数 1	无
返回值	0 正确。 1 错误

15.2.15 函数 UART_Ir_Init

函数名	UART_Ir_Init
函数原型	void UART_Ir_Init(unsigned char Irflag,unsigned char Th,unsigned int Tzbrg)
功能描述	红外调制初始化
输入参数 1	Irflag: 调制极性
输入参数 2	th: 占空比调制参数



输入参数 3	Tzbrg : 调制频率参数
输出参数 1	无
返回值	无

15.2.16 函数 UART_Ir_Enable

函数名	UART_Ir_Enable
函数原型	void UART_Ir_Enable(unsigned char UARTx)
功能描述	红外调制打开
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

15.2.17 函数 UART_Ir_Disbale

函数名	UART_Ir_Disbale
函数原型	v void UART_Ir_Disbale(unsigned char UARTx)
功能描述	红外调制关闭
输入参数 1	UARTx: UART0、UART1、UART2、UART3
输出参数 1	无
返回值	无

16 RTC

16.1 RTC 寄存器结构

寄存器	描述
RTCWE	RTC 写使能寄存器
RTCIE1	RTC 中断使能寄存器 1
RTCIF1	RTC 中断标志寄存器 1
RTCIE2	RTC 中断使能寄存器 2
RTCIF2	RTC 中断标志寄存器 2
BCDSEC	BCD 时间寄存器
BCDMIN	
BCD HOUR	
BCDDATE	
BCDWEEK	
BCDMONTH	
BCDYEAR	
ALARMSEC	闹钟设置寄存器
ALARMMIN	
ALARMHOUR	
FSEL	FOUT 输出选择寄存器
ADJUST	LTBC 计时调校寄存器
ADJUST1	
ADSIGN	LTBC 调校符号寄存器
PRLSEN	精确秒时标虚拟调校使能
SECCNT	RTC 秒内计数值

16.2 RTC 库函数

函数名	描述
RTC_Interrupt_Enable	RTC 中断使能
RTC_Interrupt_Disable	RTC 中断关闭
RTC_Interrupt_Flag_Clear	RTC 中断标志清
RTC_Write_Enable	RTC 写入初值时间使能
RTC_Write_Disable	RTC 写入初值时间关闭



RTC_Byte_Is_BCD	字节是否是 BCD
RTC_Judge_Clock	判断时间是否合法, 秒 分 时 周 日 月 年
RTC_Read_Rtc	读取时间
RTC_Set_Rtc	设置时间
RTC_Syn_Clock	同步主和备份时钟
RTC_Read_Sysclock	带时钟错误用备份时钟修正 的读时钟
RTC_Fout_Select	RTC Fout 选择 (输出还需 FOUT0 引脚配置正确)
RTC_Set_Alarm	设置闹钟时间
RTC_Adjust_Ltbc	RTC 校准值写入

16.2.1 函数 RTC_Interrupt_Enable

函数名	RTC_Interrupt_Enable
函数原型	void RTC_Interrupt_Enable(unsigned short rtc_sel)
功能描述	RTC 中断使能
输入参数 1	rtc_sel : 中断时间选择 RTC_T_DAY //日中断 RTC_T_HOUR //小时中断 RTC_T_MINUTE //分中断 RTC_T_SECOND //秒中断 RTC_T_2HZ //2Hz 中断 RTC_T_4HZ //4Hz 中断 RTC_T_8HZ //8Hz 中断 RTC_T_16HZ //16Hz 中断 RTC_T_ADJ128 //数字调校周期中断 RTC_T_ARLARM //ARLARM 中断 RTC_T_1KHZ //1KHz 中断 RTC_T_256HZ //256Hz 中断 RTC_T_64HZ //64Hz 中断
输出参数 1	无
返回值	无

16.2.2 函数 RTC_Interrupt_Disable

函数名	RTC_Interrupt_Disable
函数原型	void RTC_Interrupt_Disable(unsigned int rtc_sel)
功能描述	RTC 中断关闭
输入参数 1	rtc_sel : 中断时间选择



	RTC_T_DAY //日中断 RTC_T_HOUR //小时中断 RTC_T_MINUTE //分中断 RTC_T_SECOND //秒中断 RTC_T_2HZ //2Hz 中断 RTC_T_4HZ //4Hz 中断 RTC_T_8HZ //8Hz 中断 RTC_T_16HZ //16Hz 中断 RTC_T_ADJ128 //数字调校周期中断 RTC_T_ARLARM //ARLARM 中断 RTC_T_1KHZ //1KHz 中断 RTC_T_256HZ //256Hz 中断 RTC_T_64HZ //64Hz 中断
输出参数 1	无
返回值	无

16.2.3 函数 RTC_Interrupt_Flag_Clear

函数名	RTC_Interrupt_Flag_Clear
函数原型	void RTC_Interrupt_Flag_Clear(unsigned int rtc_sel)
功能描述	RTC 中断标志清
输入参数 1	rtc_sel : 中断时间选择 RTC_T_DAY //日中断 RTC_T_HOUR //小时中断 RTC_T_MINUTE //分中断 RTC_T_SECOND //秒中断 RTC_T_2HZ //2Hz 中断 RTC_T_4HZ //4Hz 中断 RTC_T_8HZ //8Hz 中断 RTC_T_16HZ //16Hz 中断 RTC_T_ADJ128 //数字调校周期中断 RTC_T_ARLARM //ARLARM 中断 RTC_T_1KHZ //1KHz 中断 RTC_T_256HZ //256Hz 中断 RTC_T_64HZ //64Hz 中断
输出参数 1	无
返回值	无



16.2.4 函数 RTC_Write_Enable

函数名	RTC_Write_Enable
函数原型	void RTC_Write_Enable(void)
功能描述	RTC 写入初值时间使能
输入参数 1	无
输出参数 1	无
返回值	无

16.2.5 函数 RTC_Write_Disable

函数名	RTC_Write_Disable
函数原型	void RTC_Write_Disable(void)
功能描述	RTC 写入初值时间使能关闭
输入参数 1	无
输出参数 1	无
返回值	无

16.2.6 函数 RTC_Byte_Is_BCD

函数名	RTC_Byte_Is_BCD
函数原型	unsigned char RTC_Byte_Is_BCD(unsigned char Value)
功能描述	字节是否是 BCD
输入参数 1	Value: 判断的值
输出参数 1	无
返回值	0 是 BCD 码 , 1 非 BCD

16.2.7 函数 RTC_Judge_Clock

函数名	RTC_Judge_Clock
函数原型	unsigned char RTC_Judge_Clock(unsigned char Start, unsigned char *Ck, unsigned char Len)
功能描述	判断时间是否合法, 秒 分 时 周 日 月 年
输入参数 1	Start: 起始
输入参数 2	Ck: 时间 buf
输入参数 3	Len: 长度
输出参数 1	无
返回值	0 正确 , 1 错误



16.2.8 函数 RTC_Read_Rtc

函数名	RTC_Read_Rtc
函数原型	unsigned char RTC_Read_Rtc(unsigned char *RTC_Buf)
功能描述	读取时间
输入参数 1	RTC_Buf: 时间 buf 指针
输出参数 1	RTC_Buf: 时间 buf 数据
返回值	0 正确 , 1 错误

16.2.9 函数 RTC_Set_Rtc

函数名	RTC_Set_Rtc
函数原型	unsigned char RTC_Set_Rtc(unsigned char Start, unsigned char *Ck, unsigned char Len)
功能描述	设置时间
输入参数 1	Start: 起始
输入参数 2	Ck: 时间 buf
输入参数 3	Len: 长度
输出参数 1	无
返回值	0 正确 , 1 错误

16.2.10 函数 RTC_Syn_Clock

函数名	RTC_Syn_Clock
函数原型	void RTC_Syn_Clock (void)
功能描述	同步主和备份时钟
输入参数 1	无
输出参数 1	无
返回值	无

16.2.11 函数 RTC_Read_Sysclock

函数名	RTC_Read_Sysclock
函数原型	void RTC_Read_Sysclock(void)
功能描述	带时钟错误用备份时钟修正 的读时钟
输入参数 1	无
输出参数 1	无



返回值	无
-----	---

16.2.12 函数 RTC_Fout_Select

函数名	RTC_Fout_Select
函数原型	void RTC_Fout_Select(unsigned int fsel)
功能描述	RTC Fout 选择 (输出还需 FOUT0 引脚配置正确)
输入参数 1	fsel: 0:高电平精确秒方波 1: 高电平 80ms 秒时标 2:高电平秒进位信号 3: 高电平分进位信号 4:高电平小时进位信号 5:高电平日进位信号 6: 高电平闹钟匹配信号 7: 128 秒方波信号 8:低电平 80ms 秒时标 9:低电平秒进位信号 10: 低电平分进位信号 11: 低电平小时进位信号 12:低电平日进位信号 13: 低电平闹钟匹配信号 14:低电平精确秒方波 15: RTC 内部秒时标方波
输出参数 1	无
返回值	无

16.2.13 函数 RTC_Set_Alarm

函数名	RTC_Set_Alarm
函数原型	unsigned char RTC_Set_Alarm(unsigned char *Ck)
功能描述	设置闹钟时间
输入参数 1	Ck: 时间 buf 秒 分 时 3 字节 BCD
输出参数 1	无
返回值	0 正确 , 1 错误

16.2.14 函数 RTC_Adjust_Ltbc

函数名	RTC_Adjust_Ltbc
函数原型	void RTC_Adjust_Ltbc(unsigned char adsign,unsigned int adjust,unsigned char visual)
功能描述	设置闹钟时间
输入参数 1	adsign 调整符号 0 增加 1 减少
输入参数 2	adjust 调整数值
输入参数 3	visual 虚拟调校 0 使能, 1 禁止
输出参数 1	无
返回值	无



17 版本说明

版本	日志
V1.0	首次发布
V1.1	增加 lcd 、 rtc、 uart 驱动函数, 修正一些错误