



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

<b>1</b>	<b>工程使用说明</b>	<b>7</b>
<b>2</b>	<b>GPIO</b>	<b>8</b>
2.1	GPIO 寄存器结构 :	8
2.2	GPIO 函数库 :	10
2.2.1	<i>GPIO_init</i>	10
2.2.2	<i>GPIO_Read_Input_Data_Bit</i>	11
2.2.3	<i>GPIO_Write_Output_Data_Bit</i>	12
2.2.4	<i>GPIO_Write_Toggle_Data_Bit</i>	12
2.2.5	<i>GPIO_Read_Output_Data_Bit</i>	12
2.2.6	<i>GPIO_Read_Input_Data</i>	12
2.2.7	<i>GPIO_Write_Output_Data</i>	13
2.2.8	<i>GPIO_Write_Toggle_Data</i>	13
2.2.9	<i>GPIO_Read_Output_Data</i>	13
2.2.10	<i>GPIO_Fout0</i>	13
2.2.11	<i>GPIO_Fout1</i>	14
<b>3</b>	<b>CLK</b>	<b>15</b>
3.1	CLK 寄存器结构	15
3.2	CLK 库函数	15
3.2.1	<i>CLK_init</i>	15
3.2.2	<i>CLK_Set_Peripheral_Clk</i>	16
3.2.3	<i>CLK_Clr_Peripheral_Clk</i>	17
<b>4</b>	<b>CRC</b>	<b>19</b>
4.1	CRC 寄存器结构	19
4.2	CRC 函数库	19
4.2.1	<i>CalCRC16_CCITT</i>	19
4.2.2	<i>BitChange</i>	19
<b>5</b>	<b>EXIT</b>	<b>21</b>
5.1	EXIT 寄存器结构	21
5.2	EXIT 函数库	21
5.2.1	<i>EXIT_init</i>	21
5.2.2	<i>EXIT_Close</i>	22
5.2.3	<i>EXIT_Select_Edge</i>	23
<b>6</b>	<b>I2C</b>	<b>25</b>



6.1	I2C 寄存器结构 .....	25
6.2	I2C 函数库 .....	25
6.2.1	<i>Init_I2c</i> .....	25
6.2.2	<i>I2C_Write_Bottom</i> .....	25
6.2.3	<i>I2C_Read_Bottom</i> .....	26
<b>7</b>	<b>SPI .....</b>	<b>27</b>
7.1	SPI 寄存器结构 .....	27
7.2	SPI 函数库 .....	27
7.2.1	<i>SPI_user_init</i> .....	27
7.2.2	<i>SPI_init</i> .....	27
7.2.3	<i>SPI_rw_byte</i> .....	28
7.2.4	<i>SPI_send_buf</i> .....	28
7.2.5	<i>SPI_recv_buf</i> .....	28
<b>8</b>	<b>ADC .....</b>	<b>29</b>
8.1	ADC 寄存器结构 .....	29
8.2	ADC 函数库 .....	29
8.2.1	<i>ADC_Init</i> .....	29
8.2.2	<i>ADC_Enable</i> .....	30
8.2.3	<i>ADC_Disable</i> .....	30
8.2.4	<i>ADC_Get_Done_Flag</i> .....	30
8.2.5	<i>ADC_Get_Value</i> .....	30
8.2.6	<i>ADC_Clear_Done_Flag</i> .....	31
8.2.7	<i>ADC_Result_Voltage_Convert</i> .....	31
8.2.8	<i>ADC_Result_Temperature_Convert</i> .....	31
<b>9</b>	<b>FLASH 擦写/FLASH 模拟 EEPROM .....</b>	<b>32</b>
9.1	FLASH 寄存器结构 .....	32
9.2	FLASH 擦写/FLASH 模拟 EEPROM 库函数 .....	32
9.2.1	<i>Flash_Erase_SECTOR</i> .....	32
9.2.2	函数 <i>Flash_Write_BYTE</i> .....	33
9.2.3	函数 <i>Flash_Write_SHORT</i> .....	33
9.2.4	函数 <i>Flash_Write_STRING</i> .....	33
9.2.5	函数 <i>Flash_Write_SECTOR</i> .....	33
9.2.6	函数 <i>Flash_Read_BYTE</i> .....	34
9.2.7	函数 <i>Flash_Read_SHORT</i> .....	34
9.2.8	函数 <i>Flash_Read_STRING</i> .....	34
9.2.9	函数 <i>Flash_Int_Cfg</i> .....	34
<b>10</b>	<b>LPTIMER .....</b>	<b>36</b>







10.1	LPTIMER 寄存器结构.....	36
10.2	LPTIMER 库函数 .....	36
10.2.1	函数 LPTIM_Init.....	36
10.2.2	函数 LPTIM_PWM.....	37
10.2.3	函数 LPTIM_Init_Ms_Interrupt.....	37
10.2.4	函数 LPTIM_Init_Count_Interrupt.....	37
10.2.5	函数 LPTIM_Init_Trigger_Interrupt.....	38
10.2.6	函数 LPTIM_Enable .....	38
10.2.7	函数 LPTIM_Disable.....	38
11	定时器中断 .....	39
11.1	定时器寄存器结构.....	39
11.2	定时器中断库函数.....	41
11.2.1	函数 TIME_T0_Init_time_Interrupt.....	41
11.2.2	函数 TIME_T1_Init_time_Interrupt.....	41
11.2.3	函数 TIME_T2_Init_time_Interrupt.....	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.....	45
12.2.1	函数 PCA_PWM_Init.....	45
12.2.2	函数 PCA_Capture_Init.....	46
12.2.3	函数 PCA_Capture_Interrupt_Enable.....	46
12.2.4	函数 PCA_Capture_Interrupt_Disable.....	46
12.2.5	函数 PCA_Compare_Init.....	46
12.2.6	函数 PCA_Compare_Interrupt_Enable.....	47
12.2.7	函数 PCA_Compare_Interrupt_Disable .....	47
12.2.8	函数 PCA_Compare_Toggle_Out_Enable.....	47
12.2.9	函数 PCA_Cap_Cmp_Interrupt_Flag_Clear .....	47
12.2.10	函数 PCA_Overflow_Init.....	48
12.2.11	函数 PCA_Overflow_Interrupt_Enable.....	48
12.2.12	函数 PCA_Overflow_Interrupt_Disable.....	48
12.2.13	函数 PCA_Overflow_Interrupt_Flag_Clear.....	48
12.2.14	函数 PCA_Enable .....	49
12.2.15	函数 PCA_Disable.....	49
13	SVD .....	50
13.1	SVD 寄存器结构 .....	50
13.2	SVD 库函数 .....	50
13.2.1	函数 SVD_Init.....	50
13.2.2	函数 SVD_Open .....	50



13.2.3	函数 <i>SVD_Close</i> .....	51
13.2.4	函数 <i>SSVD_Software_Debounce_Judgement</i> .....	51
13.2.5	函数 <i>SVD_Software_Debounce_Judgement</i> .....	51
14	版本说明 .....	52

# 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
 Drives	2017/9/22 9:30	文件夹
 Project	2017/9/22 10:23	文件夹
 Sys	2017/9/22 10:40	文件夹
 User	2017/9/22 9:30	文件夹

- Drives 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，包含程序.C 文件和.H 头文件；
- Project 文件夹下存放了 Keil 的工程配置文件，工程输出文件等；
- Sys 文件夹下存放了工程仿真用的插件等；
- User 文件夹下存放了主程序及用户项目程序，包含程序.C 文件和.H 头文件，用户可以在这里添加自己的程序文件；

## 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	
PHOS	

## 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	mode 端口功能：

	<div>输入浮空： GPIO_MODE_IN_FLOATING</div> <div>输入上拉： GPIO_MODE_IN_PULL_UP</div> <div>开漏输出（仅支持 PORTB 和 PORTE）： GPIO_MODE_OUT_OPEN_DRAIN</div> <div>开漏输出内部上拉（仅支持 PORTB 与 PORTE）： GPIO_MODE_OUT_OPEN_DRAIN_PULL_UP</div> <div>推挽输出： GPIO_MODE_OUT_PUSH_PULL</div> <div>管脚不使用： GPIO_MODE_IO_OFF</div> <div>模拟： GPIO_MODE_ANALOG</div> <div>特殊功能选择： GPIO_MODE_ALTERNATE_FUNCTION</div> <div>特殊功能选择内部上拉： GPIO_MODE_ALTERNATE_FUNCTION_PULL_UP</div>
输入参数 4	<div>select 功能 (mode 为特殊功能选择时有效)：</div> <div><div>特殊功能选择 0： MODE_ALTERNATE_SELECT_0</div><div>特殊功能选择 1： MODE_ALTERNATE_SELECT_1</div><div>特殊功能不选择（保持原样）： MODE_ALTERNATE_SELECT_NULL</div></div>
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_Set_Peripheral_Clk	CLK 设置外置时钟
3	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\_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 <div>             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           </div>



	PERIPHERAL_ALL_CLK
输出参数	无
返回值	无

### 3.2.3 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  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	

## 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 : {0x11,0x22,0x33,0x44}
输入参数 3	Len 数据长度 : 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

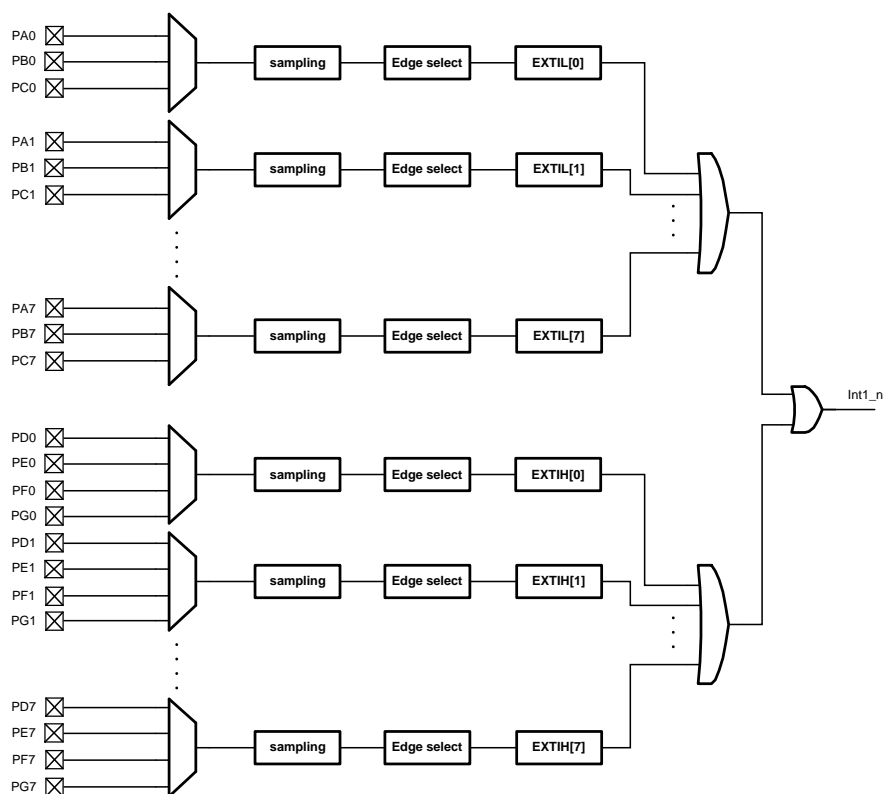
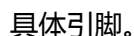
寄存器	描述
GPIO_EXTI_SEL0	GPIOx 中断选择寄存器
GPIO_EXTI_SEL1	
GPIO_EXTI_SEL2	
GPIO_EXTI_SEL3	
GPIO_EXTIL_ES0	GPIO 中断边沿选择寄存器
GPIO_EXTIL_ES1	
GPIO_EXTIH_ES0	
GPIO_EXTIH_ES1	
EXTILIF	GPIO 中断标志寄存器
EXTIHIF	

### 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



输入参数 3	Edge 触发沿： EXTI_TRIGGER_RISING EXTI_TRIGGER_FALLING EXTI_TRIGGER_RISING_FALLING EXTI_TRIGGER_DISABLE
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"/> PB0 <input type="checkbox"/> PC0 <input type="checkbox"/></p> <p>PA1 <input type="checkbox"/> PB1 <input type="checkbox"/> PC1 <input type="checkbox"/></p> <p>...</p> <p>PA7 <input type="checkbox"/> PB7 <input type="checkbox"/> PC7 <input type="checkbox"/></p> <p>PD0 <input type="checkbox"/> PE0 <input type="checkbox"/> PF0 <input type="checkbox"/> PG0 <input type="checkbox"/></p> <p>PD1 <input type="checkbox"/> PE1 <input type="checkbox"/> PF1 <input type="checkbox"/> PG1 <input type="checkbox"/></p> <p>...</p> <p>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]</p> <p>sampling Edge select 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。



	<p>PA0 <input type="checkbox"/> PB0 <input type="checkbox"/> PC0 <input type="checkbox"/></p> <p>PA1 <input type="checkbox"/> PB1 <input type="checkbox"/> PC1 <input type="checkbox"/></p> <p>...</p> <p>PA7 <input type="checkbox"/> PB7 <input type="checkbox"/> PC7 <input type="checkbox"/></p> <p>PD0 <input type="checkbox"/> PE0 <input type="checkbox"/> PF0 <input type="checkbox"/> PG0 <input type="checkbox"/></p> <p>PD1 <input type="checkbox"/> PE1 <input type="checkbox"/> PF1 <input type="checkbox"/> PG1 <input type="checkbox"/></p> <p>...</p> <p>PD7 <input type="checkbox"/> PE7 <input type="checkbox"/> PF7 <input type="checkbox"/> PG7 <input type="checkbox"/></p> <p>EXTIL[0] EXTIL[1] ... EXTIL[7]</p> <p>EXTIH[0] EXTH[1] ... EXTH[7]</p> <p>Int1_n</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



## 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_user_init	SPI 寄存器用户初始化
2	SPI_init	SPI 寄存器初始化
3	SPI_rw_byte	SPI 收发一个字节
4	SPI_send_buf	SPI 发送一串字节
5	SPI_recv_buf	SPI 接受一串字节

#### 7.2.1 SPI\_user\_init

函数名	SPI_user_init
函数原型	void SPI_user_init(void)
功能描述	SPI 寄存器，用户初始化，包含端口与 SPI 寄存器
输入参数	无
输出参数	无
返回值	无

#### 7.2.2 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	
输出参数	无
返回值	无

### 7.2.3 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

### 7.2.4 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

### 7.2.5 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℃ )
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
ET1CTRL1	ET1 控制寄存器 1
ET1CTRL2	ET1 控制寄存器 2
ET1CFG1	ET1 配置寄存器 1
ET1CFG2	ET1 配置寄存器 2
ET1PRESCALE	ET1 预分频寄存器
ET1LOADCTRL	ET1 加载控制寄存器
ET1CNTL	ET1 计数值低位
ET1CNTH	ET1 计数值高位
ET1PRESETL	ET1 预置数寄存器低位
ET1PRESETH	ET1 预置数寄存器高位
ET1LOADL	ET1 计数加载寄存器低位
ET1LOADH	ET1 计数加载寄存器高位
ET1CMPL	ET1 比较值寄存器低位
ET1CMPH	ET1 比较值寄存器高位
ET1OUTCNTL	ET1 输出脉冲宽度寄存器低位
ET1OUTCNTH	ET1 输出脉冲宽度寄存器高位
ET1OUTCTRL	ET1 输出控制寄存器
ET1IE	ET1 中断使能寄存器
ET1IF	ET1 中断标志寄存器
ET2CTRL1	ET2 控制寄存器 1
ET2CTRL2	ET2 控制寄存器 2
ET2CFG1	ET2 配置寄存器 1
ET2CFG2	ET2 配置寄存器 2
ET2PRESCALE	ET2 预分频寄存器
ET2LOADCTRL	ET2 加载控制寄存器
ET2CNTL	ET2 计数值低位
ET2CNTH	ET2 计数值高位
ET2PRESETL	ET2 预置数寄存器低位
ET2PRESETH	ET2 预置数寄存器高位
ET2LOADL	ET2 计数加载寄存器低位
ET2LOADH	ET2 计数加载寄存器高位



ET2CMPL	ET2 比较值寄存器低位
ET2CMPH	ET2 比较值寄存器高位
ET2OUTCNTL	ET2 输出脉冲宽度寄存器低位
ET2OUTCNTH	ET2 输出脉冲宽度寄存器高位
ET2OUTCTRL	ET2 输出控制寄存器
ET2IE	ET2 中断使能寄存器
ET2IF	ET2 中断标志寄存器
ET3CTRL	ET3 控制寄存器
ET3INSEL	ET3 输入选择寄存器
ET3PRESCALE1	ET3 预分频寄存器 1
ET3PRESCALE2	ET3 预分频寄存器 2
ET3STARTL	ET3 低位初值寄存器
ET3STARTH	ET3 高位初值寄存器
ET3IE	ET3 中断使能寄存器
ET3IF	ET3 中断标志寄存器
ET4CTRL	ET4 控制寄存器
ET4INSEL	ET4 输入选择寄存器
ET4PRESCALE1	ET4 预分频寄存器 1
ET4PRESCALE2	ET4 预分频寄存器 2
ET4STARTL	ET4 低位初值寄存器
ET4STARTH	ET4 高位初值寄存器
ET4IE	ET4 中断使能寄存器
ET4IF	ET4 中断标志寄存器
CCON	PCA 控制寄存前
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

## 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
ET1CTRL1	ET1 控制寄存器 1
ET1CTRL2	ET1 控制寄存器 2
ET1CFG1	ET1 配置寄存器 1
ET1CFG2	ET1 配置寄存器 2
ET1PRESCALE	ET1 预分频寄存器
ET1LOADCTRL	ET1 加载控制寄存器
ET1CNTL	ET1 计数值低位
ET1CNTH	ET1 计数值高位
ET1PRESETL	ET1 预置数寄存器低位
ET1PRESETH	ET1 预置数寄存器高位
ET1LOADL	ET1 计数加载寄存器低位
ET1LOADH	ET1 计数加载寄存器高位
ET1CMPL	ET1 比较值寄存器低位
ET1CMPH	ET1 比较值寄存器高位
ET1OUTCNTL	ET1 输出脉冲宽度寄存器低位
ET1OUTCNTH	ET1 输出脉冲宽度寄存器高位
ET1OUTCTRL	ET1 输出控制寄存器
ET1IE	ET1 中断使能寄存器
ET1IF	ET1 中断标志寄存器
ET2CTRL1	ET2 控制寄存器 1
ET2CTRL2	ET2 控制寄存器 2
ET2CFG1	ET2 配置寄存器 1
ET2CFG2	ET2 配置寄存器 2
ET2PRESCALE	ET2 预分频寄存器
ET2LOADCTRL	ET2 加载控制寄存器
ET2CNTL	ET2 计数值低位
ET2CNTH	ET2 计数值高位
ET2PRESETL	ET2 预置数寄存器低位
ET2PRESETH	ET2 预置数寄存器高位
ET2LOADL	ET2 计数加载寄存器低位
ET2LOADH	ET2 计数加载寄存器高位



ET2CMPL	ET2 比较值寄存器低位
ET2CMPH	ET2 比较值寄存器高位
ET2OUTCNTL	ET2 输出脉冲宽度寄存器低位
ET2OUTCNTH	ET2 输出脉冲宽度寄存器高位
ET2OUTCTRL	ET2 输出控制寄存器
ET2IE	ET2 中断使能寄存器
ET2IF	ET2 中断标志寄存器
ET3CTRL	ET3 控制寄存器
ET3INSEL	ET3 输入选择寄存器
ET3PRESCALE1	ET3 预分频寄存器 1
ET3PRESCALE2	ET3 预分频寄存器 2
ET3STARTL	ET3 低位初值寄存器
ET3STARTH	ET3 高位初值寄存器
ET3IE	ET3 中断使能寄存器
ET3IF	ET3 中断标志寄存器
ET4CTRL	ET4 控制寄存器
ET4INSEL	ET4 输入选择寄存器
ET4PRESCALE1	ET4 预分频寄存器 1
ET4PRESCALE2	ET4 预分频寄存器 2
ET4STARTL	ET4 低位初值寄存器
ET4STARTH	ET4 高位初值寄存器
ET4IE	ET4 中断使能寄存器
ET4IF	ET4 中断标志寄存器
CCON	PCA 控制寄存前
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 控制寄存前
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 版本说明

版本	日志
V1.0	首次发布