



# SDI5212/SDI516系列

V3.4

深圳芯易德(Solidic)科技有限公司

电话： 86-0755-82981311

传真： 86-0755-82844170

地址： 深圳宝安区西乡宝源路名优产品中心 A412

公司网站： [www.solidic.net](http://www.solidic.net)



## 版本更新记录

版本	修改时间	说明
V2.X		Flash 操作说明（操作前，最好清除安全状态，请注意最新的样例代码）
		在配置PCON进入休眠前，最好加一条空指令“NOP”
		在采用“STOP1/2进入指令”的程序中，最好将“非唤醒中断”优先级设置为低（复位默认），用作唤醒的中断设置为高优先级！！（10.4 MCU工作模式）
V3.0	2014.07	修改SDI5212的管脚图
V3.2	2015.08	
V3.4	2015.08	增加了定时器0/1计数模式时的外部输入管脚T0/T1描述 修PWM的描述



<b>1-概述</b> .....	<b>5</b>
1.1 主要特征: .....	5
1.2 其他: .....	5
1.3 型号及封装.....	5
<b>2-引脚</b> .....	<b>6</b>
<b>3-特殊功能寄存器</b> .....	<b>8</b>
<b>4-存储器</b> .....	<b>9</b>
4.1 RAM.....	9
4.2 FLASH.....	9
<b>5-I/O 口</b> .....	<b>10</b>
5.1 相关特殊寄存器.....	10
5.2 I/O 口模式配置.....	11
5.3 IO 口驱动电流.....	12
<b>6-定时/计数器</b> .....	<b>13</b>
6.1 相关特殊寄存器.....	13
6.2 定时/计数器的四种模式.....	14
<b>7-中断</b> .....	<b>16</b>
7.1 相关特殊寄存器.....	16
<b>8-串口(UART)</b> .....	<b>18</b>
8.1 相关特殊寄存器.....	18
<b>9- (SRA) 8 位 ADC</b> .....	<b>19</b>
9.1 相关特殊寄存器.....	19
9.2 (SAR) 8 位 ADC 参考程序.....	19
<b>10-时钟系统、电源管理</b> .....	<b>20</b>
10.1 时钟系统概述.....	20
10.2 外部震荡.....	21
10.3 MCU 的时钟特性.....	21
10.4 MCU 工作模式.....	21
10.5 电压监测.....	22
<b>11-看门狗与低频唤醒</b> .....	<b>22</b>
11.1 WDCON 的“访问窗口” .....	23
11.2 “看门狗” .....	23
<b>12-脉宽调制模块(PWM)</b> .....	<b>24</b>
12.1 相关寄存器.....	24
12.2 工作说明.....	错误! 未定义书签。



13-内部 LDO.....	25
<b>14-(SIGMA-DELTA) 24 位 ADC.....</b>	<b>27</b>
14.1 概述.....	27
14.2 相关寄存器.....	27
14.3 噪声性能: .....	29
<b>15-温度传感器.....</b>	<b>29</b>
15.1 温度传感器概述.....	29
15.2 相关寄存器.....	30
15.3 温度传感器的转换数值.....	30
15.4 温度测量样例程序.....	30
<b>16-FLASH 操作说明.....</b>	<b>33</b>
16.1 FLASH 概述.....	33
16.2 FLASH 数据区的“读” .....	33
16.3 相关寄存器.....	34
16.4 “FLASH 数据区”的操作保护.....	34
16.5 “FLASH 数据区”的“擦除” .....	35
16.6 “FLASH 数据区”的“写入” .....	35
16.7 FLASH 的抗干扰程序样例.....	36
<b>17 在线 ISP 程序烧录.....</b>	<b>38</b>
<b>18-电器特性.....</b>	<b>39</b>
18.1 极限条件.....	39
18.2 直流特性.....	39
参考: “10.6 低功耗配置” .....	错误! 未定义书签。
参考: “5.3 IO 口驱动电流” .....	39
18.3 ADC 参数.....	40
参考 “14.3 噪声性能” .....	40



# 1-概述

## 1.1 主要特征:

- **内核:** 增强型80C51 (8051单片机兼容)
- **Flash:** 超过100,000 次的烧写寿命, 室温下数据可保存超过100年。  
30KB(SDI5212)/8K (SDI5215) /30K (SDI5216)FLASH 空间;  
内置ISP功能 (SDA、SCL两线烧录)
- **RAM:** 512Bytes  
--- 256 Bytes 内部RAM  
--- 256 Bytes 内嵌外部寻址RAM (XDATA)
- **时钟:** (主震荡, 看门狗均可配置采用外部晶振)  
9.83MHz 内部RC震荡  
32KHz 内部看门狗时钟 (经过4分频输入到看门狗)
- **电源/功耗:**  
工作电压: 2.1V - 5.5V  
MCU核全速工作 (9.83MHz), 功耗 < 1mA(关闭ADC等相关外设)
- **主要外设:**  
--- 4通道8位低精度ADC  
--- 温度通道24位高精度ADC  
--- 温度传感器  
--- 可配置基准源输出 (LDO输出1.5v、2.0v、2.5v)

## 1.2 其他:

- ◇ 两个16 位定时/计数器
- ◇ 10个中断源, 2级优先级
- ◇ 一组UART
- ◇ 15 位看门狗-8K时钟 (32k内部RC, 内部4分频)
- ◇ 2路8位脉宽调制 (PWM) 输出
- ◇ IO可配置4种工作模式  
4个大电流驱动IO口
- ◇ 4T 指令周期

## 1.3 型号及封装

	FLASH	RAM	PACKAGE	REMARK
SDI5212AS	30KB	512B	SOP16	
SDI5212AD	30KB	512B	DIP16	
SDI5216AD	30KB	512B	SOP20	
SDI5216AS	30KB	512B	DIP20	
SDI5216B	30KB	512B		



## 2-引脚

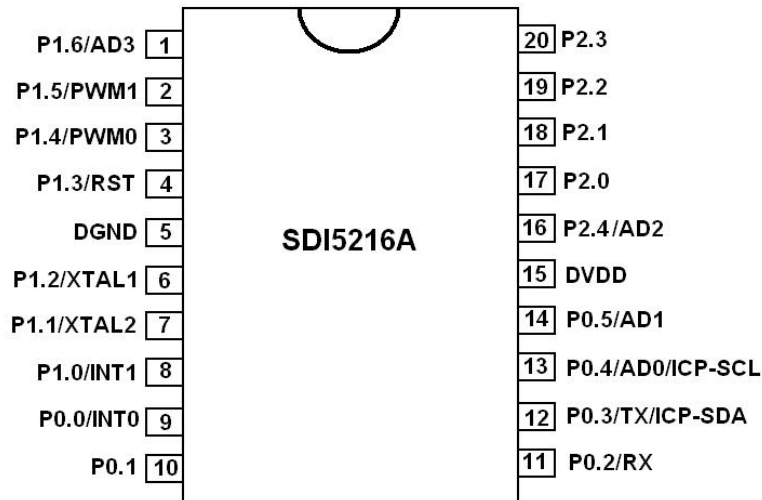


图 1: SDI5216A 管脚图

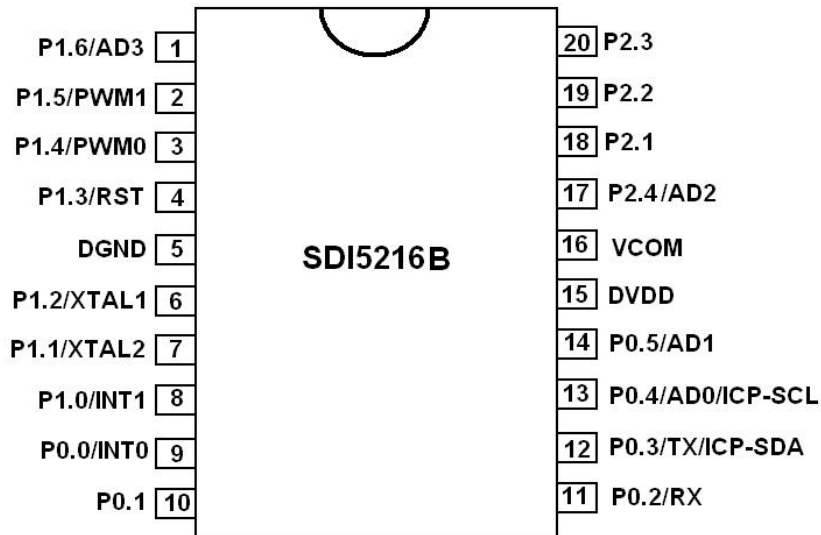


图 2: SDI5216B 管脚图

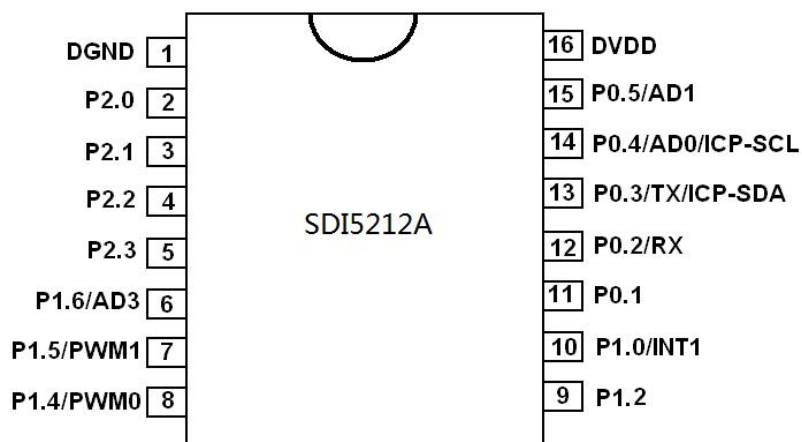


图 3: SDI5212A 管脚图



	类型	说明
DVDD	P	数字电源
DGND	P	数字地
AVDD	P	模拟电源
AGND	P	模拟地
VDDR	P	数字滤波管脚（接 0.1uF - 1uF 电容）
P1.0 - P1.6	I/O	普通 IO
P0.0 -- P0.5	I/O	普通 IO
P2.0 - P2.5	I/O	普通 IO
<b>LDO 输出</b>		
VCOM	0	内部 LDO 输出，可配置为： 1.5v\2.0v\2.5v\DVDD 输出
<b>其他复用 IO</b>		
PWMO\PWM1	I/O	脉宽调制模块输出（复用 P1.5、P1.4）
RST	I/O	外部复位管脚（复用 P1.3）
XTAL1\XTAL2		接外部晶振(复用 P1.1、P1.2)
INT0	I/O	外部中断 0（复用 P0.0）
INT1	I/O	外部中断 1（复用 P1.0）
RX	I/O	UART 的接受信号(复用 P0.2)
TX	I/O	UART 的发射信号（复用 P0.3）
ICP-SDA	I/O	在线烧录信号： 数据（复用 P0.3） （为了不影响烧录，请不要用小电阻将其拉到地）
ICP-SCL	I/O	在线烧录信号： 时钟（复用 P0.4） （为了不影响烧录，请不要用小电阻将其拉到地）
AD0\AD1\ AD2\AD3	I	8 位 ADC 的 4 路输入（复用:P0.4、P0.5、P2.4、P1.6）



### 3-特殊功能寄存器

	8	9	A	B	C	D	E	F
F8	EIP							
F0	B							
E8	EIE							
E0	ACC							
D8	EICON							
D0	PSW							
C8	***	***	***					
C0		FLASH_DATA	FLASH_ADDL	FLASH_ADDH	FLASH_ENA *	FLASH_ENB *	FLASH_ENC *	FLASH_CON
B8	IP	***	***	***	***	***	***	
B0	P3	SGADCON	SGADC3	SGADC2	SGADC1	***	***	***
A8	IE	WDCON	WD_TA *	SARCON	SARDATA			
A0	P2	SGADCON2	PD_CON					
98	SCON	SBUF	PWMF_H	PWMF_L	PWM0	PWM1	PWM_CON	
90	P1	EXIF	P0M1	P0M1	P1M1	P1M1	P2M1	P2M1
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
80	P0	SP	DPL	DPH	DPL (1)	DPH (1)	DSP (1)	PCON
	0	1	2	3	4	5	6	7

\*: 只写

\*\*\*: 内部保留





## 4-存储器

### 4.1 RAM

对于SDI5216系列，内建512 字节RAM。

- ◇ 用户可直接寻址开始的128 字节RAM，我们叫它直接RAM，它的地址空间是 00h~7Fh.
- ◇ 接下来的128 字节RAM，用户可以间接寻址到它。我们叫它间接RAM，它的空间地址是80h~FFh
- ◇ 其它的RAM被叫做扩展RAM，它占用的空间地址00h~FFh 用户可以通过寄存器Ri 或数据指针DPTR，使用**MOVX** 指令来访问它，如：[MOVX A,@R1](#) 或者 [MOVX A,@DPTR](#)。

SDI5216系列RAM空间：

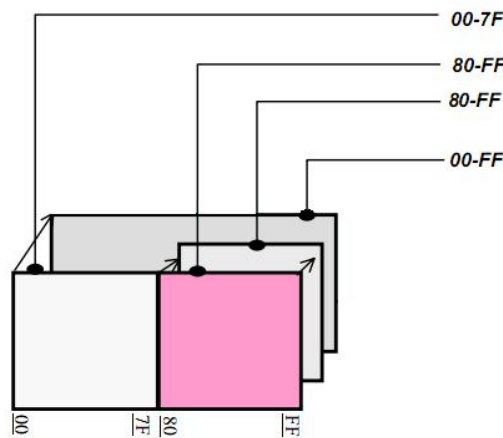


图4： RAM空间

### 4.2 FLASH

对于SDI5216系列，总共有30K的Flash空间。该Flash空间在烧录时可分配为：“程序空间”以及“数据空间”

- ◇ 程序空间：用户程序存储空间
- ◇ 数据空间：非易失性数据存储空间，程序可修改此空间数据

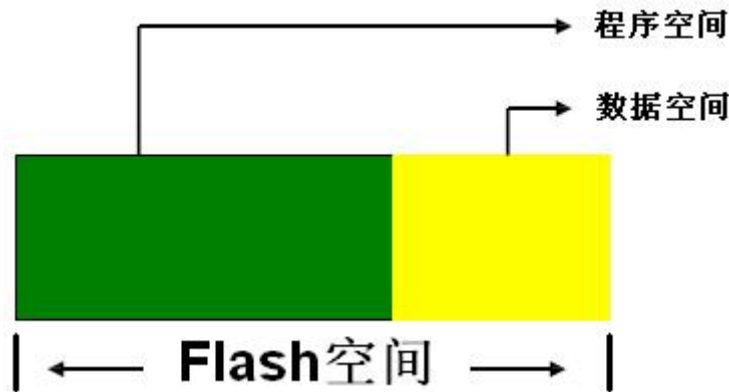


图 5： Flash 空间



## 5-I/O 口

SDI5216系列的I/O 口可分别设成4 种不同的模式：(X=0,1,2; n=0,1,2,3,4,5,6,7)

PxM1n	PxM0n	I/O 口模式
0	0	标准51输出模式(默认)
0	1	CMOS推拉输出
1	0	仅输入
1	1	集电极开路

注意：配置为“仅输入”“集电极开路”时，IO口的对应“数据寄存器”必须置1，否则IO口将被拉到地。

要注意IO口的配置锁定功能

### 5.1 相关特殊寄存器

P0口：默认配置“标准51输出模式”

■ **P0 (0x80): P0 口数据寄存器（默认值：0xFF）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		P05	P04	P03	P02	P01	P00

■ **P0M0(0x92): P0 口模式配置寄存器0（默认值：0x00）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		P0M05	P0M04	P0M03	P0M02	P0M01	P0M00

■ **P0M1(0x93): P0 口模式配置寄存器1（默认值：0x00）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		P0M15	P0M14	P0M13	P0M12	P0M11	P0M10

P1口：P1.1, P1.2默认配置“仅输入模式”，其它为“标准51输出模式”

■ **P1 (0x90): P1 口数据寄存器（默认值：0xFF）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	P16	P15	P14	P13	P12	P11	P10

■ **P1M0(0x94): P1 口模式配置寄存器0（默认值：0x00）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00

■ **P1M1(0x95): P1 口模式配置寄存器1（默认值：0x06）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10

P2口：默认配置“标准51输出模式”

■ **P2(0xA0): P2 口数据寄存器（默认值：0xFF）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			P24	P23	P22	P21	P20

■ **P2M0(0x96): P2 口模式配置寄存器0（默认值：0x00）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			P2M04	P2M03	P2M02	P2M01	P2M00



■ P2M1(0x97): P2 口模式配置寄存器1 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			P2M14	P2M13	P2M12	P2M11	P2M10

I/O 口配置锁定:

■ PD\_CON(0xA2): 休眠辅助寄存器 (默认值: 0x06)

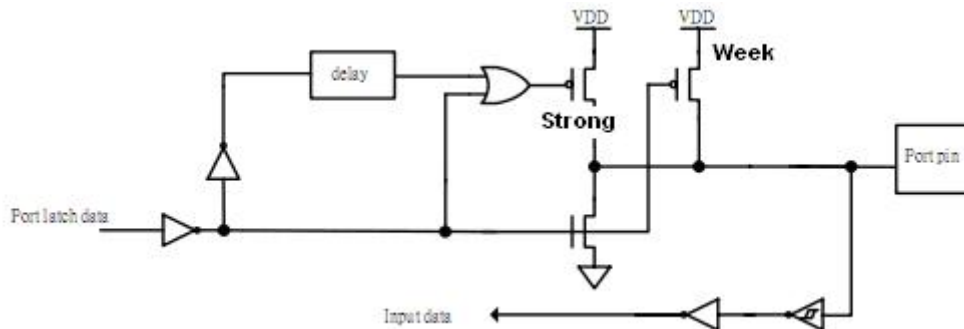
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
XIO_LCK	P2_LCK1	P2_LCK0	SG_LCK1	SG_LCK0	ALDO_EN	ALDO_SYN	TEMP_EN

- XIO\_LCK:** 外置振荡复用 IO 锁定
  - 1: 锁定 P1.1、p1.2 为仅输入模式
  - 0: 解锁(IO 模式由对应的模式控制位控制)
- P2\_LCK1 \ P2\_LCK0:** P2 口 IO 锁定
  - 11、10、01: 锁定 P2 为仅输入模式
  - 00: 解锁(IO 模式由对应的模式控制位控制)
- SG\_LCK1 \ SG\_LCK0:** 24 位 ADC 核心寄存器锁定
  - 11、10、01: 锁定寄存器: sgadcon, sgadcon2, 忽略写入功能
  - 00: 解锁写权限
- ALDO\_EN:** 内部LDO使能 “参见章节: 12-内部LDO”
  - 1: 开启内部LDO
  - 0: 关闭内部LDO
- ALDO\_SYN:** 内部LDO同步信号
  - 1: 内部LDO和24位ADC 同步休眠
  - 0: 不同步, 关掉24为ADC, 内部LDO正常工作
- TEMP\_EN:** 温度模块使能
  - 1: 开启温度模块
  - 0: 关闭温度模块

5.2 I/O 口模式配置

■ 标准51 输出模式

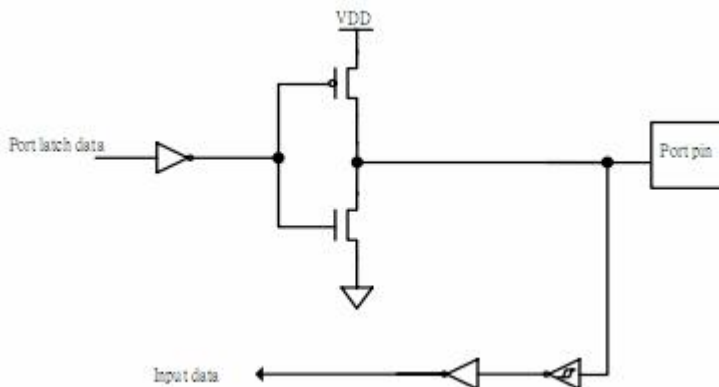
PxM1n = 0, PxM0n = 0



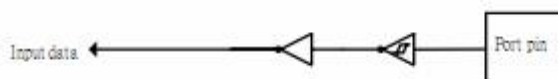
- ✓ 上升沿会启动100ns的强上拉, 然后强上拉关闭, 只剩下弱上拉
- ✓ 弱上拉相当50K的电阻。此配置下, IO口输出低电平, 将持续有电流消耗。低功耗是需要注意!!



- CMOS推拉输出  
 $PxM1n = 0; PxM0n = 1;$



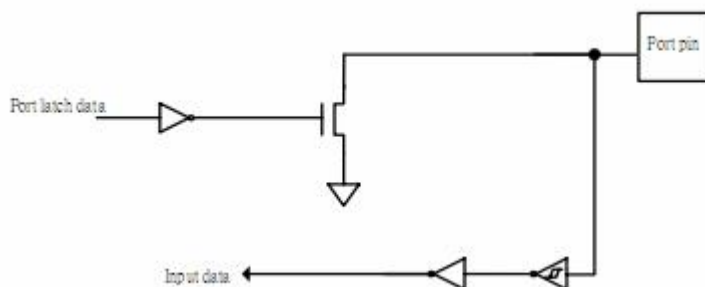
- 仅输入模式(高阻)  
 $PxM1n = 1; PxM0n = 0;$



注意:

- ◇ 该模式用于给MCU内部ADC等输入模拟信号，内部数字IO输入被锁定为高电平，也就是说如果配置成该模式的管脚，通过指令读取IO口电平恒为高电平。
- ◇ 该模式必须将IO口对应的寄存器位写1才能有效，如果对应的寄存器为0，则IO口将被拉到地。

- 开集输出模式  
 $PxM1n = 1; PxM0n = 1;$



### 5.3 IO口驱动电流

	外灌电流能力	吸入电流能力
P1.3/P1.4/P1.5/P1.6(大驱动IO)	25mA	30mA
其他IO	7mA	15mA



## 6-定时/计数器

SDI5216系列提供了两个16 位定时/计数器 T0,T1。

（相关中断，请参考第7章“7-中断”）

注意：使用内部震荡提供串口时钟时，时器基准时钟采用“CLK\_OSC /4”，这样能配置各种串口速率。

### 6.1 相关特殊寄存器

#### ■ TMOD(0x89): TIMER 模式控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

用于T1

用于T0

- GATE: 0: 只要TRx 置1, Timer x 即使能  
1: 必须TRx 置1, 且/INTx 为高, Timer x 才使能
- C/T: 0: 作为定时器  
1: 作为计数器
- M1,M0 模式选择  
0,0: 作为13 位定时/计数器  
0,1: 作为16 位定时/计数器  
1,0: 作为8 位自动重载定时/计数器, 重载值存于THx  
1,1: 对于T0, TL0 是一个8 位定时/计数器, TH0 是一个8 位定时器  
T1 被停止

#### ■ TCON(0x88)（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- TF1: T1 溢出标志位  
当T1 溢出时, 该位会自动置1. 当执行T1 溢出中断时, 该位自动清零.
- TR1: 0: 停止T1  
1: 开始T1
- TF0: T0 溢出标志位  
当T0 溢出时, 该位会自动置1. 当执行T0 溢出中断时, 该位自动清零.
- TR0: 0: 停止T0  
1: 开始T0
- IE1: 外部中断1 标志  
当外部中断1 产生时, 该位会自动置1. 当执行外部中断1 时, 该位自动清零.
- IT1: 0: 引脚EX1 低电平, 产生外部中断1  
1: 引脚EX1 下降沿, 产生外部中断1
- IE0: 外部中断0 标志  
当外部中断0 产生时, 该位会自动置1. 当执行外部中断0 时, 该位自动清零.
- IT0: 0: 引脚EX0 低电平, 产生外部中断0  
1: 引脚EX0 下降沿, 产生外部中断0



■ CKCON(0x8E): 时钟辅助寄存器 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				T1M	T0M	CKDIV1	CKDIV0

T0M: 定时器T0 时钟选择

0: CLK\_MCU /12

1: CLK\_MCU /4

T1M: 定时器T1 时钟选择

0: CLK\_MCU /12

1: CLK\_MCU /4

CKDIV1/CKDIV0: 用于MCU时钟分频

00: CLK\_MCU = CLK\_OSC

01: CLK\_MCU = CLK\_OSC / 2

10: CLK\_MCU = CLK\_OSC / 4

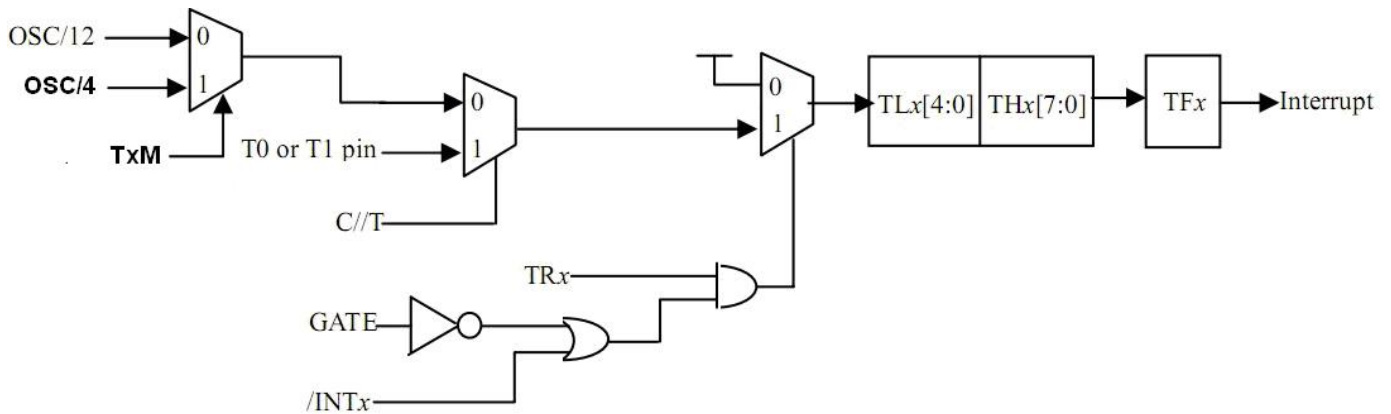
11: CLK\_MCU = CLK\_OSC / 8

6.2 定时/计数器的四种模式

下图中: T0(P0.1)、T1(P0.4)、INTx (INT0\INT1)

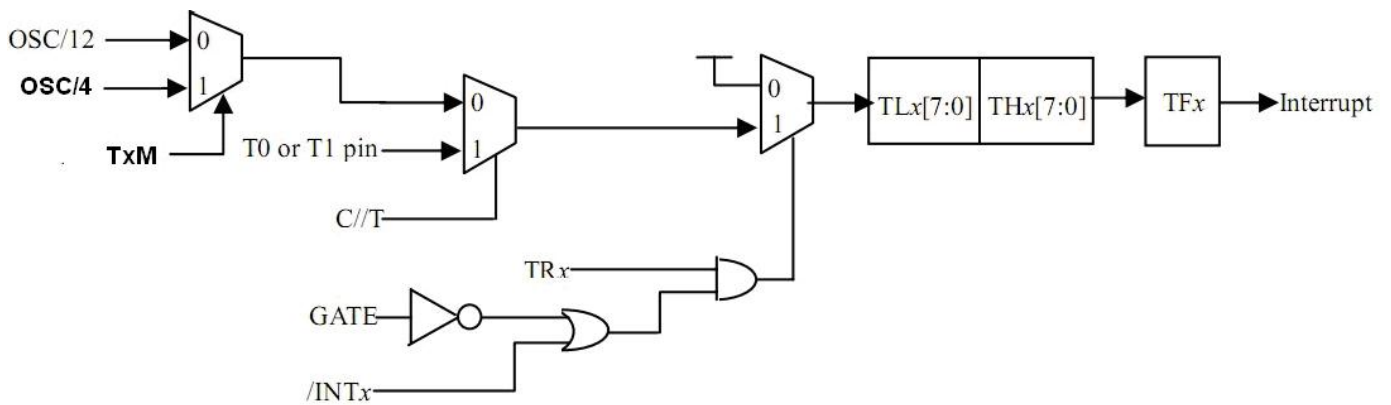
■ M1,M0 = 0,0: 模式0

13 位定时/计数器



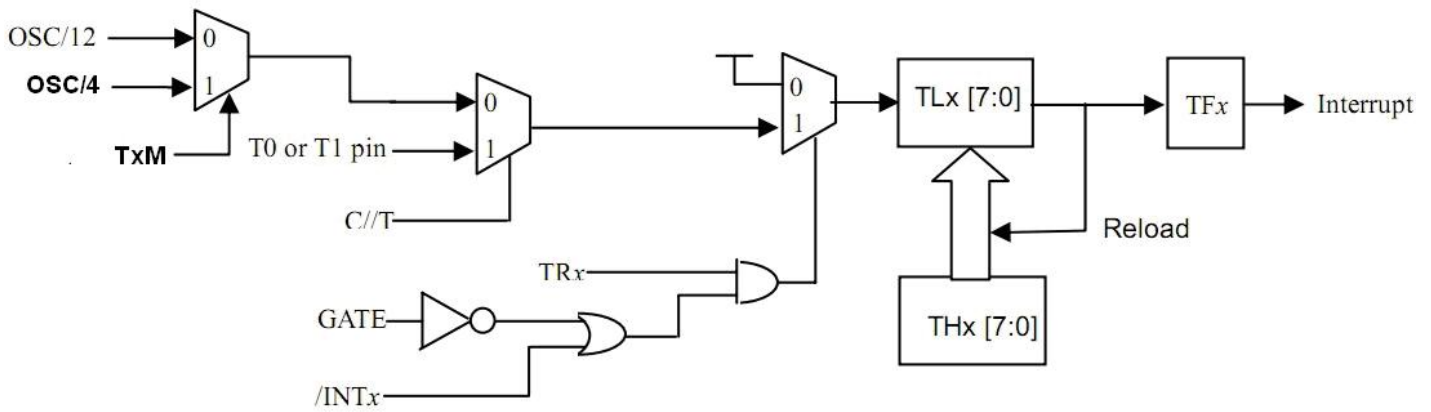
■ M1,M0 = 0,1: 模式1

16 位定时/计数器

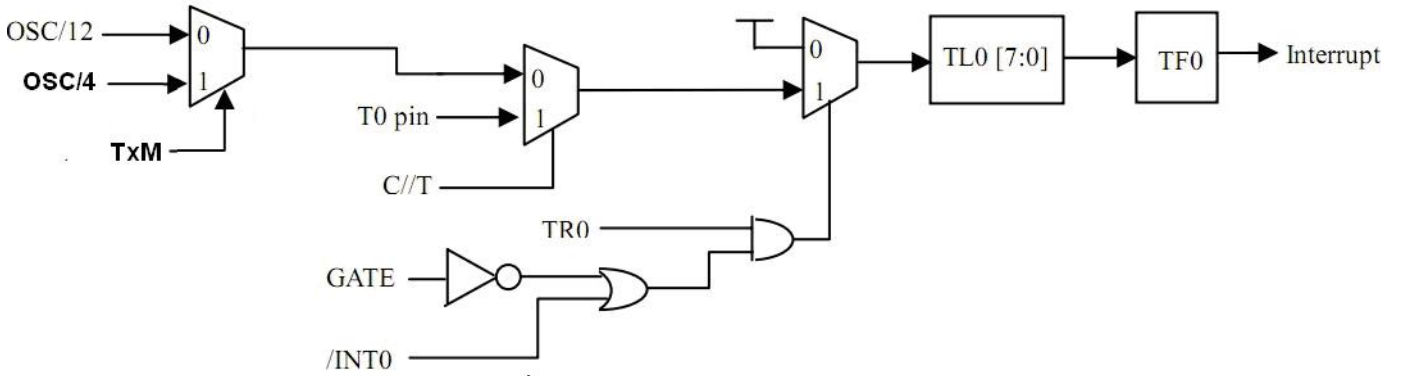




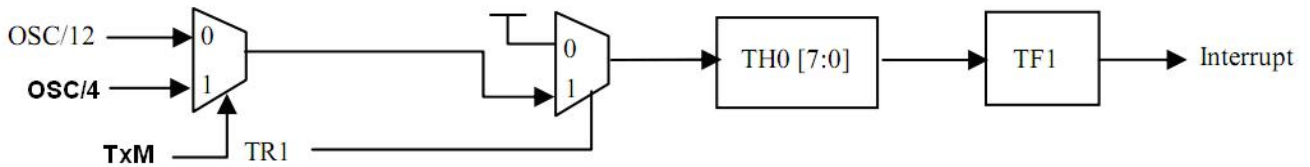
- **M1,M0 = 1,0:** 模式2  
8 位自动重载定时/计数器



- **M1,M0 = 1,1:** 模式3  
TL0 是一个8 位定时/计数器



TH0 是一8 位定时器，使用TR1 使能，溢出时置位TF1





## 7-中断

SDI5216系列，提供10个中断源，2级优先级。处理高优先级中断时，不会响应低优先级的中断请求。如果两个不同优先级的中断同时发出请求，高优先级的中断请求将会被响应。如果相同优先级的中断同时发出请求，则由内部优先级来决定哪个中断会被响应。下表说明了内部优先级和中断向量地址

中断源	中断向量地址	中断内部优先级
外部中断0	03H	1(最高)
定时器0	0BH	2
外部中断1	13H	3
定时器1	1BH	4
串口	23H	5
(保留)	2BH	6
低频时钟唤醒中断	33H	7
24位ADC中断	3BH	8
8位ADC中断	43H	9
看门狗中断	4BH	10

### 7.1 相关特殊寄存器

■ **IE(0xA8)** - 中断始能（5个普通中断源的中断始能）（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
EA			ES	ET1	EX1	ET0	EX0

EA: 0: 禁止所有中断

1: 使能中断

ES: 串口中断使能位

0: 禁止

1: 使能

ET1: 定时器1 中断使能位

0: 禁止

1: 使能

EX1: 外部中断使能位

0: 禁止

1: 使能

ET0: 定时器0 中断使能位

0: 禁止

1: 使能

EX0: 外部中断0 使能位

0: 禁止

1: 使能

■ **IP(0xB8)** 中断优先级（5个普通中断源的中断优先级）（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			PS	PT1	PX1	PT0	PX0

对应位 1 的优先级高于 0 的优先级





■ **EIE(0xE8)** - 扩展中断始能（4个扩展中断源的中断始能）（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			EWDI	EX5	EX4	EX3	

- EWDI: “看门狗中断” 使能位  
0: 禁止  
1: 使能
- EX5: “8位ADC中断”  
0: 禁止  
1: 使能
- EX4: “24位ADC中断使能位”  
0: 禁止  
1: 使能
- EX3: “低频时钟唤醒中断” 使能位  
0: 禁止  
1: 使能

■ **EIP(0xF8)** 扩展中断优先级（4个扩展中断源的中断优先级）（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			PWDI	PX5	PX4	PX3	

对应位 1 的优先级高于 0 的优先级

■ **EXIF (0x91)** 扩展中断的标志位（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
IE5	IE4	IE3					

- IE5: “8位ADC中断” 标志  
当“8位ADC中断”产生时，该位会自动置1，该位需要手动清零。
- IE4: “24位ADC中断使能位” 标志  
当“24位ADC中断使能位”产生时，该位会自动置1，该位需要手动清零。
- IE3: “低频时钟唤醒中断” 标志  
当“低频时钟唤醒中断”产生时，该位会自动置1，该位需要手动清零。

■ **EICON(0xD8)** 扩展中断控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				WDFI			

- WDFI: “看门狗中断” 标志  
当“看门狗中断”产生时，该位会自动置1，该位需要手动清零。



## 8-串口(UART)

兼容标准 C51 的串口，具体模式说明请参考标准 C51 的相关资料

(相关中断，请参考第 7 章“7-中断”)

### 8.1 相关特殊寄存器

■ **SBUF(0x99):** 串口发送，接收数据寄存器（默认值：0x00）

■ **SCON(0x98):** 串口控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM1 和 SM0 定义串行口操作模式 见下表

SM0 SM1 UART 模式

0 0	模式0,	同步移位寄存器
0 1	模式1,	8 位 UART
1 0	模式2,	9 位 UART
1 1	模式3,	9 位 UART

(对于工作模式的具体描述，请参照标准51的相关说明)

SM2 在模式 2 和3 中多处理机通信使能位 在模式 2 或3 中若SM2=1 且接收到的第9 位数据 RB8 是 0 则 RI 接收中断标志 不会被激活 在模式1 中若 SM2=1 且没有接收到有效的停止位 则 RI 不会被激活 在模式 0 中SM2 必须是 0

REN 允许接收位 由软件置位或清除 REN=1 时允许接收 REN=0 时 禁止接收

TB8 模式 2 和 3 中发送的第 9 位数据 可以按需要由软件置位或清除

RB8 模式 2 和 3 中已接收的第 9 位数据 在模式 1 中 或 sm2=0 RB8 是已接收的停止位 在模式 0 中 RB8 未用

TI 发送中断标志 模式 0 中 在发送完第 8 位数据时 由硬件置位 其它模式中 在发送停止位之初 由硬件置位 在任何模式中 都必须由软件来清除 TI

RI 接收中断标志 模式 0 中 接收第 8 位结束时由硬件置位 其它模式中在接收停止位的中间时刻 由硬件置位.在任何模式(SM2 所述情况除外)必须由软件清除

■ **PCON(0x87):** 电源控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SMOD0	SOFT_RST	OSC_PD	STOP2	GF1	GF2	STOP1	IDLE

SMOD = 1时，串行口的波特率加倍



## 9-（SRA）8 位 ADC

（相关中断，请参考第 7 章“7-中断”）

SDI5216系列提供了4通道的ADC，当相关I/O口作为ADC 通道使用时，该I/O 口必须设为仅输入模式(高阻)，且I/O口相关寄存器的对应位需要置1。

### 9.1 相关特殊寄存器

请注意相关保留位保持默认值

- **SARDATA(0xAC):** ADC 转换结果数据寄存器（默认值：0x00）

SAR ADC的参考电压为(DVDD-DGND)

$$SARDAT = 256 * ((V_{in} - DGND) / (DVDD - DGND))$$

- **SARCON (0xAB):** ADC 转换控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
保留	保留	SAR_FCON1	SAR_FCON0	SAREN	SAR_STA	SAR_SEL1	SAR_SELO

SAR\_FCON 1, SAR\_FCON 0 : ADC 转换速度选择寄存器

0, 0	(default), 转换时钟约560KHz, 40 个机器周期（完成转换）
0, 1	转换时钟约280KHz, 80 个机器周期（完成转换）
1, 0	转换时钟约140KHz, 160 个机器周期（完成转换）
1, 1	转换时钟约70KHz, 320 个机器周期（完成转换）

SAREN 0: 关闭SARADC模块的电源; 1: 打开SARADC模块的电源

SAR\_STA 置1 后, SARADC 开始转换, 转换完成后自动清零。

SAR\_SEL1, SAR\_SELO: 通道选择位

0, 0	(AD0通道)设置P0.4 为ADC 通道(默认)
0, 1	(AD1通道)设置P0.5 为ADC 通道
1, 0	(AD2通道)设置P2.4 为ADC 通道
1, 1	(AD3通道)设置P1.6 为ADC 通道

### 9.2 （SAR）8位ADC参考程序

```

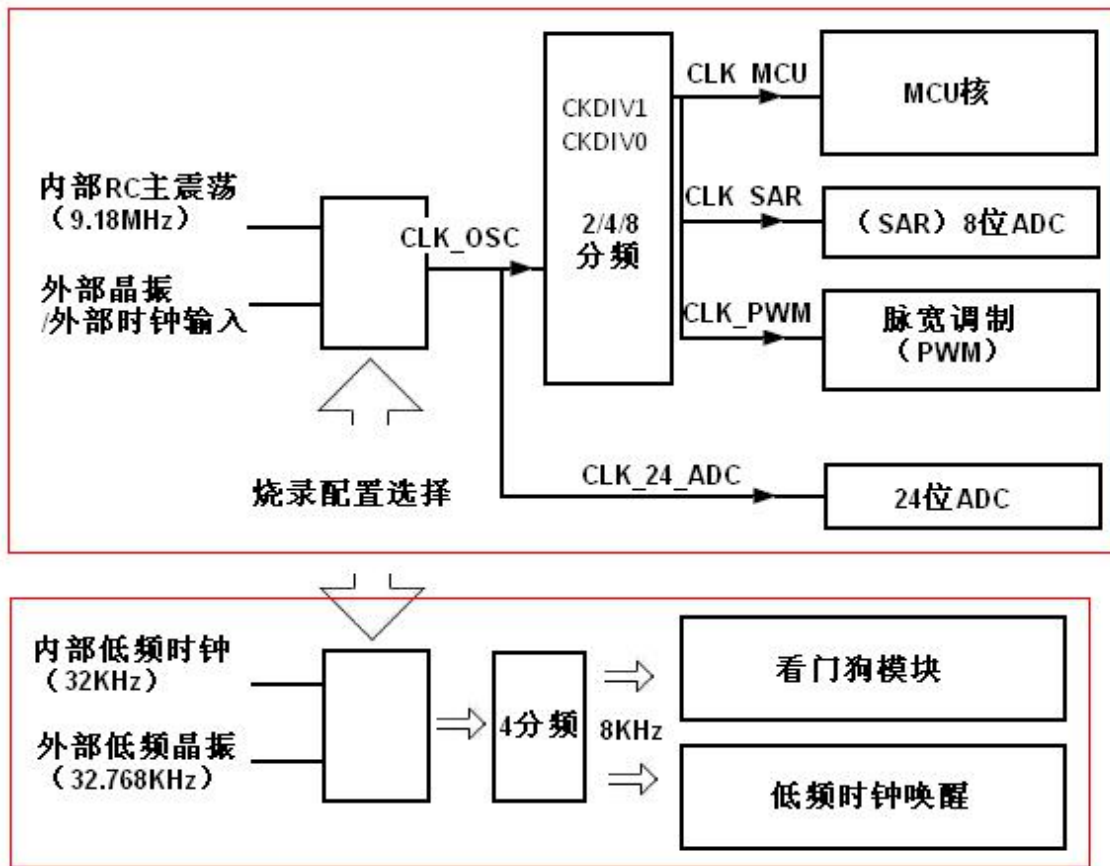
//--测试通道 3 的电压
SARCON = 0x0b;           //通道 3, 转换时钟最快, 打开 Sar-ADC
                          //-- 注意配置相关 IO 口
if(!(SARCON & 0x04))     // ADC 不忙则开始转换
{
    SARCON |= 0x04;       //ADC 开始转换
    while(SARCON & 0x04) // 等待转换结束
    {
    }
}
//得到转换数据在 SARDATA 中
SARCON &= 0xf7;         //关闭 ADC

```



# 10-时钟系统、电源管理

## 10.1 时钟系统概述



SDI5216 系列的时钟分为两部分：“主震荡”和“低频时钟”，两者均可在烧录时配置是否采用外部晶振，外部震荡管脚和 P1.1, P1.2 复用。（任何时候，只能有一个时钟被配置为外部晶振）

上图详细描述了各个模块对应的时钟。

- ◇ 主 (RC) 震荡： 9.83MHz
- ◇ 内部低频 (RC) 震荡： 32KHz

■ **CKCON(0x8E):** 时钟辅助寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				T1M	T0M	CKDIV1	CKDIV0

T0M: 定时器T0 时钟选择

0: CLK\_MCU /12

1: CLK\_MCU /4

T1M: 定时器T1 时钟选择

0: CLK\_MCU /12

1: CLK\_MCU /4

CKDIV1/CKDIV0: 用于MCU时钟分频（CLK\_MCU、CLK\_SAR、CLK\_PWM的时钟频率相同）

00: CLK\_MCU = CLK\_OSC

01: CLK\_MCU = CLK\_OSC / 2

10: CLK\_MCU = CLK\_OSC / 4

11: CLK\_MCU = CLK\_OSC / 8



## 10.2 外部震荡

当配置位外部震荡时（以下两个时钟均适合）：

- ◇ P1.1, P1.2 为连接外部晶振的管脚，分别接 22pF 的电容到地。
- ◇ 程序执行过程中，禁止将修改 P1.1、P1.2 的工作模式（寄存器“PD\_CON”锁定 P1.1、P1.2 为仅输入模式，防止被修改，具体参考寄存器“PD\_CON”）。
- ◇ 内部具备时钟检测模块，当外部震荡不起振时，自动切换到内部震荡，指导外部震荡重新起振。
- ◇ “低频时钟”采用外置“32.768KHz”的晶振，配合“看门狗”，可以实现准确的时钟记时功能

## 10.3 MCU 的时钟特性

- ◇ SDI5216 系列采用 4T 模式，1 个机器周期对应 4 个时钟周期。
- ◇ 可通过时钟分频控制给到 MCU 以及相关外设的时钟频率

## 10.4 MCU 工作模式

系统具备三种工作模式，可以通过寄存器 PCON 来配置，在配置 PCON 进入休眠前，最好加一条空指令“NOP”。

■ PCON(0x87): 电源控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SMOD0	SOFT_RST	OSC_PD	STOP2	GF1	GF2	STOP1	IDLE

- SMOD: = 1时，串行口的波特率加倍
- SOFT\_RST: 和用户程序无关（系统标志）
- OSC\_PD: = 1: “掉电模式二”时，不关闭“主振荡器”  
= 0: “掉电模式二”时，关闭“主振荡器”
- STOP2: = 1, 单片机进入“掉电模式二”
- GF1/ GF2: 通用标志位
- STOP1: = 1, 单片机进入“掉电模式一”
- IDLE: = 1, 单片机进入“休眠模式”

MCU 核的省电工作模式，主要是通过关闭相关部分的时钟来实现的

	状态描述	进入方式	退出方式
IDLE: “休眠模式”	<ul style="list-style-type: none"> <li>✓ MCU 停止读取指令，进入等待</li> <li>✓ 相关的时钟并不关闭</li> </ul>	PCON[0]置 1	<ul style="list-style-type: none"> <li>✓ 所有中断</li> <li>✓ 退出自动清除 PCON[0]</li> </ul>
STOP1: “掉电模式一”	<ul style="list-style-type: none"> <li>✓ 关闭CLK_MCU</li> <li>✓ 关闭CLK_SAR</li> <li>✓ 关闭CLK_PWM</li> <li>✓ 其他外设是否关闭，需要用户自己配置相关标志位！</li> </ul>	PCON[1]置 1	<ul style="list-style-type: none"> <li>✓ 外部中断 0, 1</li> <li>✓ 低频时钟唤醒中断</li> <li>✓ 24 位 ADC 中断</li> <li>✓ 退出自动清除 PCON[1]</li> </ul>
STOP2: “掉电模式二”	<ul style="list-style-type: none"> <li>✓ 关闭CLK_MCU</li> <li>✓ 关闭CLK_SAR</li> <li>✓ 关闭CLK_PWM</li> <li>✓ 关闭CLK_24_ADC</li> <li>✓ 无视其他配置强制关闭（24位 ADC、内部LDO、温度传感器）</li> <li>✓ 当OSC_PD=0时，关闭主震荡器</li> </ul>	PCON[4]置 1	<ul style="list-style-type: none"> <li>✓ 外部中断 0, 1</li> <li>✓ 低频时钟唤醒中断</li> <li>✓ 退出自动清除 PCON[4]</li> </ul>

### 注意事项：

- ✓ 低频 32K 时钟始终运行，方便通过“低频时钟唤醒中断”唤醒“掉电模式二”



- ✓ 可唤醒“STOP1/STOP2”模式的四个“可唤醒中断”（外部中断 0/1、Sigma-Delta 中断、低频唤醒），如果“可唤醒中断”和“STOP1/2 进入指令”同时发生，将忽略“STOP1/2 进入指令”。
- ✓ 其他“非唤醒中断”和“STOP1/2 进入指令”同时发生时，仍然会进入 STOP1/2，同时可能导致无法退出 STOP1/2 模式！为了防止这种情况，在采用“STOP1/2 进入指令”的程序中，最好将“非唤醒中断”优先级设置为低（复位默认），用作唤醒的中断设置为高优先级！！

## 10.5 电压监测

SDI5216 内部设置有上电复位以及掉电监测模块

- ✧ 上电复位电压门限为 2.0v。
- ✧ 掉电监测门限为 1.9v。当电压低于此门限 30uS 时，会产生复位信号，将整个 MCU 复位。
- ✧ MCU 的掉电速度最好能保持在 5mV/us 以下

# 11-看门狗与低频唤醒

（相关中断，请参考第 7 章“7-中断”）

“看门狗”和“低频唤醒”均采用 8KHz 的时钟（32K 的 4 分频）。其中“低频唤醒”主要用于在“掉电模式”下的自动唤醒。

如下是“看门狗控制寄存器”，对它的访问需要先开启“访问窗口”

■ **WDCON (0xA9):** 看门狗控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
<b>WUEN</b>	<b>WUT1</b>	<b>WUT0</b>	WDEN	WDT1	WDT0	WDRST	WDCLR

**WUEN:** 低频唤醒

1: 开启

0: 关闭

**WUT1/ WUT0:** 低频唤醒时间选择

00: (计数 1200 溢出) 0.15s

01: (计数 1600 溢出) 0.20s

10: (计数 3200 溢出) 0.4s

11: (计数 8000 溢出) 1.0s

**WDEN:** 看门狗

1: 开启

0: 关闭

**WDT1/ WDT0:** 看门狗时间选择

00: (11 位计数器, 2^11 溢出) 0.25s

01: (12 位计数器, 2^12 溢出) 0.5s

10: (13 位计数器, 2^13 溢出) 1s

11: (15 位计数器, 2^15 溢出) 4s



- WDRST: 看门狗复位标志  
1: 看门狗溢出后，单片机复位  
0: 看门狗溢出后，单片机不复位
- WDCLR: 看门狗计数器清 0  
该位写 1，看门狗计数器清 0。该位在看门狗计数器清 0 后自动变为 0

### 11.1 WDCON 的“访问窗口”

正常情况下，寄存器“WDCON”处于保护状态，无法对其写入。对其操作时，需要通过寄存器“WD\_TA”开启“访问窗口”

■ **WD\_TA (0xAA):** 窗口访问控制寄存器 (不可读)

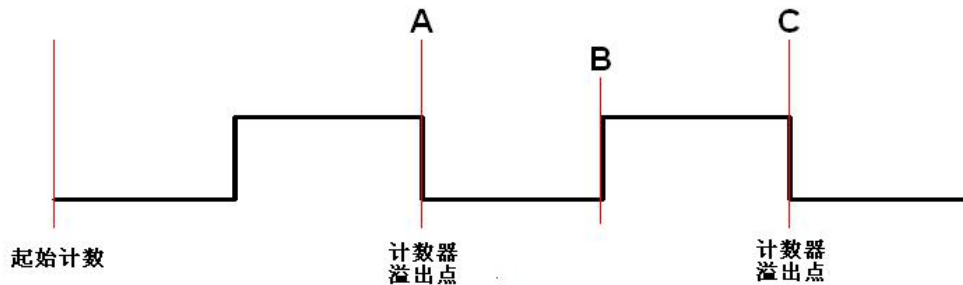
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

对 WD\_TA 依次连续写入：0x05，0x0a，将开启 4 个机器周期的时间访问寄存器“WDCON”如：

```
EA = 0;
WD_TA = 0x05;
WD_TA = 0x0a;
WDCON = 0xbf; //看门狗时间 4s 钟, 低频唤醒 0.2s
EA = 1;
```

### 11.2 “看门狗”

- ◇ “看门狗”启动后，需要清除看门狗计数器，使看门狗从确定状态开始运行
- ◇ 通过标志位“WDRST”可以配置“看门狗计数器”溢出后“系统复位”或者“执行中断”
- ◇ “掉电模式”下的看门狗运行：
  - “掉电模式一/二”时，“看门狗”启动的话，看门狗计数器正常运行。
  - ✓ 当计数器溢出后，由于处于“掉电模式”，所以系统并不响应“看门狗中断”或者“看门狗复位”。
  - ✓ 计数器溢出后，计数器重新从 0 开始继续计数。
  - ✓ 同时，“看门狗溢出”标志将会保留到，计数器重新开始计数的一半时间点。超过这个时间点，前一次溢出的标志将被清除。



看门狗计数器在 A 点溢出，由于处于“掉电模式”，不响应溢出。A 点的溢出标志将会被保留到 B 点。如果在 B 点前，退出了“掉电模式”会立刻监测到看门狗溢出。



如果是在 B 点后退出“掉电模式”，则只能等到 C 点溢出时才响应新的溢出。

### 11.3 “低频唤醒”

- “低频唤醒”启动后，如果没有处于“掉电模式一/二”，其计数器是不工作的。当进入“掉电模式一/二”后，“低频唤醒”计数器自动启动，并在计满相应的时间后产生“低频唤醒”中断，使 MCU 退出“掉电模式”

## 12-脉宽调制模块(PWM)

### 12.1 相关寄存器

请注意相关保留位保持默认值

- **PWMF\_H (0x9A):** PWM计数器高位寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
保留	保留	保留	保留	保留	保留	保留	BIT8

- **PWMF\_L (0x9B):** PWM计数器低位寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

- **PWM0 (0x9C):** PWM通道0的门限（默认值：0xFF）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

- **PWM1 (0x9D):** PWM通道1的门限（默认值：0xFF）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0

- **PWMCON (0x9E):** PWM控制寄存器（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
保留	保留	保留	保留	保留	PWM_EN	保留	PWM0_INV

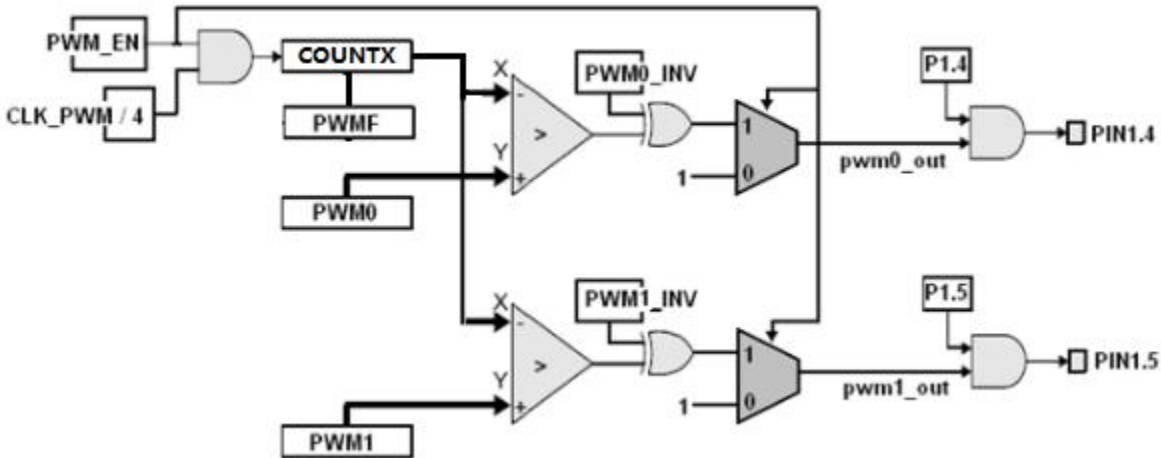
PWM\_EN: PWM 模块  
1: 开启  
0: 关闭

PWM0\_INV: PWM0 的输出配置位  
1: (PWM\_EN=1) 当  $PWMF \leq PWM0$  时, pwm0/1\_out 输出 0  
0: (PWM\_EN=1) 当  $PWMF \leq PWM0$  时, pwm0/1\_out 输出 1





12.2 工作说明



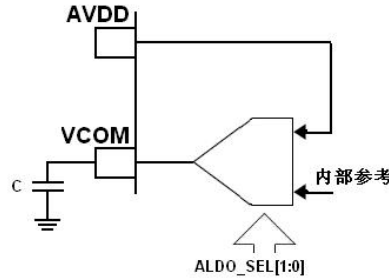
- ◇ COUNTX 为内部 16 位计数器, PWM 使能后开始从 0 计数, 当计数到门限 PWF 时从新回到 0 计数;
- ◇ PWF 计数器为 9 位, 放在寄存器 “PWF\_H” “PWF\_L” 中。
- ◇ PWM0 和 PWM1 分别为门限计数器, 分别为 8 位。
- ◇ PWM 关闭 (PWM\_EN = 0): pwm1\_out、pwm0\_out 输出 1。  
PWM 开启 (PWM\_EN = 1):
  - ✓ (PWM0\_INV = 0) :  
当 COUNTX <= PWM0 时, pwm1/0\_out 输出 1  
当 COUNTX > PWM0 时, pwm1/0\_out 输出 0
  - ✓ (PWM0\_INV = 1) :  
当 COUNTX <= PWM0 时, pwm1/0\_out 输出 0  
当 COUNTX > PWM0 时, pwm1/0\_out 输出 1

(可以通过 PWM0\_INV 控制输出反相)

- ◇ PWM 模块的时钟频率为 **CLK\_PWM 的 16 分频**
  - ✓ CLK\_PWM 可以通过 “CKDIV1/CKDIV0” 从 CLK\_OSC (9.83MHz) 分频  
“CKDIV1/CKDIV0” 最大可调节 8 分频
  - ✓ PWF 最大为 9 位, 如果不考虑 CLK\_OSC 分频最低可实现的频率为(9.83MHz / 16 / 2^9)
  - ✓ PWM0/ PWM1 比 PWF 少 1 位, 需要 PWM0\_INV 配合才能实现任意占空比的波形。



### 13-内部 LDO



- ◇ LDO 的输出可通过配置位“ALDO\_SEL[1:0]”调节
- ◇ LDO 必须在输出端 VCOM 外接 1.0uF - 10uF 的电容（当 LDO 配置输出 AVDD 时不需要）
- ◇ 上电后，LDO 默认开启，输出 AVDD

■ **SGADCON2 (0xA1):** 24位ADC控制寄存器2（默认值：0x30）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ALDO_SEL1	ALDO_SELO	保留	保留	CCA_1	CCA_0	CCB_1	CCB_0

ALDO\_SEL1/ ALDO\_SELO: 内部 LDO 输出选择寄存器

- 00: AVDD
- 01: 输出 2.5v
- 10: 输出 2.0v
- 11: 输出 1.5v

CCA\_1/ CCA\_0: (系统配置) 内部 PGA 的控制  
请选择: 00 (默认) 或者 01

CCB\_1 /CCB\_1: (系统配置) 内部 24 位 ADC 的控制  
请选择: 00 (默认)  
01 (在 CCA\_1/ CCA\_0 为 01 时)

(在无必要情况下, CCA\_1/ CCA\_0 ; CCB\_1 /CCB\_1 请保持默认值)

■ **PD\_CON(0xA2):** 休眠辅助寄存器（默认值：0x06）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
XIO_LCK	P2_LCK1	P2_LCK0	SG_LCK1	SG_LCK0	ALDO_EN	ALDO_SYN	TEMP_EN

**ALDO\_EN:** 内部LDO使能  
1: 开启内部LDO  
0: 关闭内部LDO

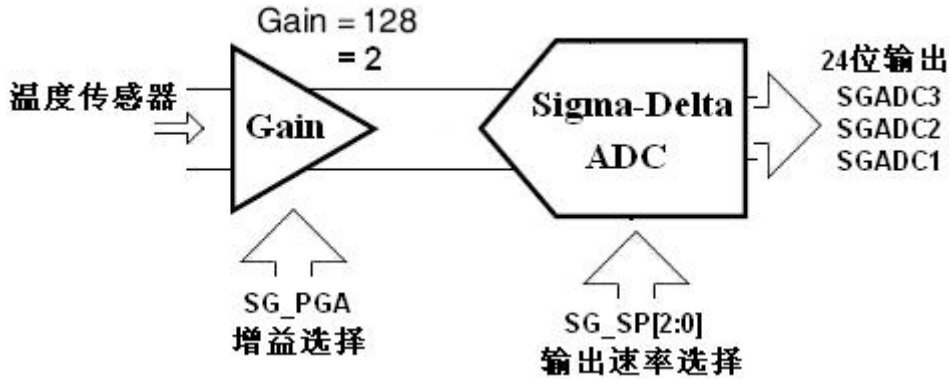
**ALDO\_SYN:** 内部LDO同步信号  
1: 内部LDO和24位ADC 同步休眠  
0: 不同步, 关掉24为ADC, 内部LDO正常工作



# 14-(SIGMA-DELTA) 24 位 ADC

(相关中断，请参考第 7 章“7-中断”) (注意：上电默认开启，如不需要请关闭)

## 14.1 概述



### ◇ $\Delta\Sigma$ ADC

高精度模数转换器的核心部分为采用 Sigma-Delta 调制器结构，集成可编程增益 (PGA = 2, 128)。该 ADC 对输入模拟差分信号的采样频率为 CLK\_OSC (9.83MHz) / 128，远高于模拟信号的最大带宽，因而简化了应用时输入通道的前置防混叠滤波器。

### ◇ 内部增益(PGA) (温度检测使用时，必须关闭 PGA)

可通过 SG\_PGA 调节是否开启内部增益模块。

开启：增益为 128 倍

关闭：增益为 2 倍

### ◇ ADC 输出速率选择 (SP)

通过 SG\_SP2、SG\_SP1、SG\_SP0 可以调节 ADC 的输出速率

10Hz、20Hz、40Hz、80Hz、160Hz、300Hz

### ◇ ADC 输出值 (参考电压输入: VREF = VCOM) (必须开启 LDO，并将 LDO 输出配置为 AVDD)

输入: AIN1P - AIN1N (输入范围: +/- 0.5VREF/增益)

采用补码输出	
输入为正	000001h - 7FFFFFFh
0000000h	000000h
输入为负	FFFFFFh - 800000h

## 14.2 相关寄存器

### ■ SGADCON(0xB1): 休眠辅助寄存器 (默认值: 0xc9)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SG_EN	SG_IDLE_N			SG_PGA	SG_SP2	SG_SP1	SG_SP0

SG\_EN:

24 位 ADC: 使能

1: 开启 0: 关闭



SG\_IDLE\_N: 24 位 ADC: 休眠控制  
1: 正常工作 0: ADC 挂起休眠

SG\_PGA: 24 位 ADC: 增益选择  
1: 128 倍 0: 2 倍  
(温度通道请选择 0: 2 倍增益, 否则信号将溢出)

SG\_SP2/ SG\_SP1/ SG\_SP0: 24 位 ADC: 输出速率选择  
000: 10Hz  
001: 20Hz  
010: 40Hz  
011: 80Hz  
100: 160Hz  
101: 300Hz

■ SGADC3(0xB2): 24位转换数据, 最高8位

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ADBT23	ADBT22	ADBT21	ADBT20	ADBT19	ADBT18	ADBT17	ADBT16

■ SGADC2(0xB3): 24位转换数据, 中间8位

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ADBT15	ADBT14	ADBT13	ADBT12	ADBT11	ADBT10	ADBT9	ADBT8

■ SGADC1(0xB4): 24位转换数据, 低8位

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ADBT7	ADBT6	ADBT5	ADBT4	ADBT3	ADBT2	ADBT1	ADBT0

■ SGADCON2 (0xA1): 24位ADC控制寄存器2 (默认值: 0x30)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ALDO_SEL1	ALDO_SELO			CCA_1	CCA_0	CCB_1	CCB_0

ALDO\_SEL1/ ALDO\_SELO: 内部 LDO 输出选择寄存器  
00: AVDD  
01: 输出 2.5v  
10: 输出 2.0v  
11: 输出 1.5v

CCA\_1/ CCA\_0: (系统配置) 内部测试信号  
请配置: 00 (默认)

CCB\_1 /CCB\_1: (系统配置) 内部测试信号  
请配置: 00 (默认)



■ PD\_CON(0xA2): 休眠辅助寄存器 (默认值: 0x06)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
XIO_LCK	P2_LCK1	P2_LCK0	SG_LCK1	SG_LCK0	ALDO_EN	ALDO_SYN	TEMP_EN

- XIO\_LCK:** 外置振荡复用 IO 锁定  
 1: 锁定 P1.1、p1.2 为仅输入模式  
 0: 解锁(IO 模式由对应的模式控制位控制)
- P2\_LCK1 \ P2\_LCK0:** P2 口 IO 锁定  
 11、10、01: 锁定 P2 为仅输入模式  
 00: 解锁(IO 模式由对应的模式控制位控制)
- SG\_LCK1 \ SG\_LCK0:** 24 位 ADC 核心寄存器锁定  
 11、10、01: 锁定寄存器: sgadcon, sgadcon2, 忽略写入功能  
 00: 解锁写权限
- ALDO\_EN:** 内部 LDO 使能 “参见章节: 12-内部 LDO”  
 1: 开启内部LDO  
 0: 关闭内部LDO
- ALDO\_SYN:** 内部LDO同步信号  
 1: 内部LDO和24位ADC 同步休眠  
 0: 不同步, 关掉24为ADC, 内部LDO正常工作
- TEMP\_EN:** 温度模块使能  
 1: 开启温度模块  
 0: 关闭温度模块

14.3 噪声性能:

增益	等效输入噪声有效值	等效输入噪声峰峰值	有效位	不动码位数
电源电压 = 3.3V ; VREFP = 3.3V ; 输出频率 = 10Hz				
2	1500nV	6.3uV	20	18
128	40nV	196nV	19.2	17

## 15-温度传感器

### 15.1 温度传感器概述

- ◇ 将 24 位 ADC 切换到温度通道, 打开温度模块, 可以通过 ADC 的转换值方便的测得温度信息;
- ◇ 不同参考电压下, ADC 的输出成比例变化。注意转换。



## 15.2 相关寄存器

参考“15-(SIGMA-DELTA) 24 位 ADC”

## 15.3 温度传感器的转换数值

具体温度下，对应的 ADC 转换数值的大小，用户可以自己实测

如下表给出了标准测量值，可以直接使用

注意：本表格的 ADC 输出数值均是右移了 2 位后的结果	
参考电压 (VREFP - VREFN)：	3.31V
0 度的输出	2317200
温度每增加 1 度的输出变大	760

## 15.4 温度测量样例程序

下面代码给出了参考电压为 3.0v 是的温度计算

```

//预定义
union  ADpattern          //定义联合体,数据可以采用字节和字两种方式访问;
{
    unsigned long  w ;
    unsigned char  b[4];
};

#define TEMP_VREF_03310           //(3.31V 参考)参考电压
#define TMEP_ZERO_CODE_0 2317200  //(3.31V 参考)温度 0 时的输出码
#define TEMP_PER_D_CODE_0 760     //(3.31V 参考)温度变化 1 度时的输出码
//--
#define ZERO_SIGNAL_CODE 2097152   //0x800000 / 4 理论上信号为 0 时的输出码
#define TEMP_CODE_DELTA (TMEP_ZERO_CODE_0 - ZERO_SIGNAL_CODE)
#define TEMP_VREF_1 3000           //实际的参考电压 (如 3000)

void temp_code_catch(bit mode)
{
    unsigned long  adcode_last = 0;
    union  ADpattern xdata temp;
    unsigned char i = 0;
    unsigned char j = 0;
    bit temp_sign;
    unsigned long  per_temp_code_acture = TEMP_PER_D_CODE_0;
    //-----
    EA = 0; //禁止中断
    //--温度-- 160Hz --

```



```
PD_CON = 0x07;      //开启温度通道/LDO
SGADCON = 0xf4;     //切换到 温度,160Hz
EXIF &= 0xbf;       //清除标志
//-- 得到温度编码 --
ADcode_pre = 0;
while (1)
{
    while((EXIF & 0x40) == 0x00) //如果没有数据
        ;
    EXIF &= 0xbf;      //清除标志
    //-----得到 ADC 的转换数据-----
    // 读取 的 转换数据
    temp.b[1] = SGADC3;
    temp.b[2] = SGADC2;
    temp.b[3] = SGADC1;
    temp.b[0] = 0;
    temp.w ^= 0x800000; // 因为输出为双极性, +0x800000 将负端平移上来
    temp.w &= 0x00ffff;
    temp.w >>= 2;
    //-----得到稳定的输出-----
    j++;
    if(labs(temp.w - adcode_last) <= 0x200) //丢掉 刚开始不稳定的输出
    {
        ADcode_pre += temp.w;
        i++;
    }
    else
    {
        adcode_last = temp.w;
        ADcode_pre = 0;
        i = 0;
    }
    if(i >= 4) //稳定的 ADC 数值超过 4 次
    {
        adcode_last = ADcode_pre >> 2;
        break;
    }
    else
    if(j >= 10)
    {
        adcode_last = temp.w;
        break;
    }
}
```



//-- 温度计算 --

//按照参考电压比例得到当前参考电压下“0度的输出码”

```
temp.w = TEMP_CODE_DELTA*TEMP_VREF_0 / TEMP_VREF_1;
```

```
temp.w += ZERO_SIGNAL_CODE;
```

//按照参考电压比例得到当前参考电压下“每1度对应的输出码”

```
per_temp_code_acture = TEMP_PER_D_CODE_0*TEMP_VREF_0/ TEMP_VREF_1;
```

```
Temperature = 0x00;
```

```
if(adcode_last >= temp.w) //零度以上
```

```
{
```

```
    ADcode_pre = adcode_last - temp.w;
```

```
    temp_sign = 0;
```

```
}
```

```
else
```

```
{
```

```
    ADcode_pre = temp.w - adcode_last;
```

```
    temp_sign = 1;
```

```
}
```

```
Temperature = ADcode_pre / per_temp_code_acture;
```

```
if(temp_sign)
```

```
    Temperature = -Temperature;
```

```
}
```



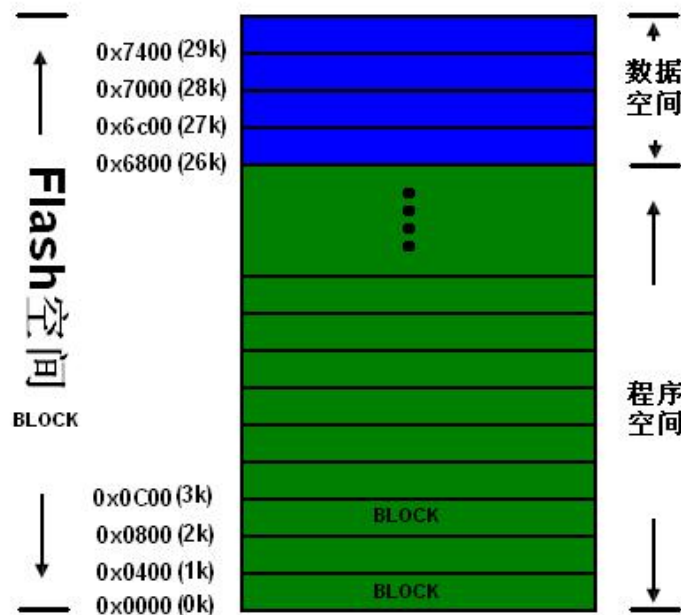


# 16-Flash 操作说明

## 16.1 Flash 概述

- ✧ Flash 存储器以“块(BLOCK-1K)”为单元组织起来,对 Flash 的擦除操作以“块(BLOCK)”为单元来进行。
- ✧ 数据区和程序区共用整个 Flash 区,烧录的时候可以配置数据区的大小。
- ✧ 数据区从 Flash 的顶端开始,程序区从 Flash 底端开始。
- ✧ 只有“数据空间”能在程序执行的过程中执行“擦除”“写入”操作。用户程序对程序区的操作无效!!

如下以 Flash-30K, 其中数据区 4K 为例绘制出示意图:



## 16.2 Flash 数据区的“读”

对 Flash 数据区的“读”操作和对程序区内部的数据读取是一样的,可用指令 MOVC 直接来进行,不需要用到下面 Flash 的相关寄存器。

下面给出 C 语言下的读取函数:

```
//读取 EEPROM 中的一个字节;
#include <absacc.h>
unsigned char nvm_data_read_byte(unsigned int addr)
{
    unsigned char    i;
    i = CBYTE[addr];
    return(i);
}
```



## 16.3 相关寄存器

■ FLASH\_DATA (0xc1): Flash 写数据寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

■ FLASH\_ADDRL (0xc2): Flash擦写地址，低8位寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
FADD7	FADD6	FADD5	FADD4	FADD3	FADD2	FADD1	FADD0

■ FLASH\_ADDRH (0xc3): Flash 擦写地址，低高位寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
FADD15	FADD14	FADD13	FADD12	FADD11	FADD10	FADD9	FADD8

■ FLASH\_ENA (0xc4): Flash操作保护字节A 寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

■ FLASH\_ENB (0xc5): Flash操作保护字节B 寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

■ FLASH\_ENC (0xc6): Flash操作保护字节C 寄存器

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

■ FLASH\_CON (0xc7): Flash擦写控制寄存器（默认值0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
INF1	INFO	MAS1	MAS0	ERASE1	ERASE0	FWR1	FWR0

FLASH\_CON :

= 0x0c 配置“擦除操作”，擦除指定的 BLOCK

= 0x03 配置“写入操作”，将 FLASH\_DATA 中的数据写入指定地址

## 16.4 “Flash 数据区”的操作保护

✧ 对 Flash 的“擦除”“写入”操作，需要先对保护字节 A、B、C 依次写入

FLASH\_ENA 写入 0x05

FLASH\_ENB 写入 0x0A

FLASH\_ENC 写入 0x09

完成后，系统开启对 Flash 的操作许可

**（执行 Flash 操作开启前，最好按照样例文件操作将保护状态复位，这样确操作许可正确打开）**

✧ Flash 执行完“擦除”“写入”操作后，保护字节将清 0。

✧ （如果可能的话），建议内存不要使用 0xc4 单元，C 程序包含如下定义

```
EXTERN unsigned char idata NRM_security_temp1_at_0xc4;
```

```
// 定义全局变量 NRM_security_temp1，占用 0xc4 的 RAM 空间，且不要操作该变量。
```



## 16.5 “Flash 数据区”的“擦除”

- 配置好地址后（地址只要指向目标区内就行），执行擦除操作，**将擦除地址所在的 BLOCK**（注意：如果地址指向程序区，擦除无效）

```
//C 语言样例程序
//EEPROM BLOCK(1k) 擦除
//addr = (0 - 31) * 1024 ,擦除对应的 Block 地址
//flash 操作关闭总中断，操作完后会开启总中断(注意)
void e2rom_erase(unsigned int addr)
{
    union INTpattern flash_addr;
    bit ea_save;
    flash_addr.i = addr;
    ea_save = EA;          // Save EA
    EA = 0;
    FLASH_ADDRH = flash_addr.b[0]; // point to the address you want to erase
    FLASH_ADDRL = flash_addr.b[1];
    //-- 状态清除 --
    FLASH_ENA = 0x00;
    FLASH_ENB = 0x00;
    FLASH_ENC = 0x00;
    FLASH_CON = 0x03; //状态清除的情况下，执行一次 FLASH_CON，将安全状态恢复到复位值
    _nop_();
    //-- 写安全码开启 Flash 操作许可 --
    FLASH_ENA = 0x05;
    FLASH_ENB = 0x0a;
    FLASH_ENC = 0x09;
    //-- Flash 操作 --
    FLASH_CON = 0x0c;
    //-- 清除安全码 --
    FLASH_ENA = 0x00;
    FLASH_ENB = 0x00;
    FLASH_ENC = 0x00;
    EA = ea_save;
}
```

## 16.6 “Flash 数据区”的“写入”

- 配置好地址以及要写入的数据后，执行“写入操作”（注意：如果地址指向程序区，擦除无效）

```
//C 语言样例程序
//往 EEPROM 中写入一个字节
//flash 操作关闭总中断，操作完后会开启总中断(注意)
void nvm_data_write_byte(unsigned int addr, unsigned char in_data)
{
```



```

union INTpattern flash_addr;
bit ea_save;
flash_addr.i = addr;
ea_save = EA;           // Save EA
EA = 0;
//-- 准备地址和数据 --
FLASH_ADDRH = flash_addr.b[0]; // point to the address you want to erase
FLASH_ADDRL = flash_addr.b[1];
FLASH_DATA = in_data;
//-- 状态清除 --
FLASH_ENA = 0x00;
FLASH_ENB = 0x00;
FLASH_ENC = 0x00;
FLASH_CON = 0x03; //状态清除的情况下，执行一次 FLASH_CON，将安全状态恢复到复位值
_nop_(); //延时
//-- 写安全码开启 Flash 操作许可 --
FLASH_ENA = 0x05;
FLASH_ENB = 0x0a;
FLASH_ENC = 0x09;
//-- Flash 操作 --
FLASH_CON = 0x03;
//-- 清除安全码 --
FLASH_ENA = 0x00;
FLASH_ENB = 0x00;
FLASH_ENC = 0x00;
EA = ea_save;
}

```

## 16.7 Flash 的抗干扰程序样例

- ✧ 为了防止 MCU 受到强干扰，直接跳转到 Flash 操作函数中，对“数据区”误操作，建议在 Flash 函数中增加安全字节“**NRM\_securty\_a**”、“**NRM\_securty\_b**”。这个两字节可定义为全局变量，当需要对 Flash 操作时，才赋值。这样，即使程序跳转错误，也不会对 Flash “数据区”误操作

```

//往 EEPROM 中写入一个字节
//调用前需要：
//NRM_securty_a,NRM_securty_b
//flash 操作关闭总中断，操作完后会开启总中断(注意)
void nvm_data_write_byte(unsigned int addr, unsigned char in_data)
{
    union INTpattern flash_addr;
    bit ea_save;
    flash_addr.i = addr;
    ea_save = EA;           // Save EA
    EA = 0;

```



```
//-- 准备地址和数据 --
FLASH_ADDRH = flash_addr.b[0]; // point to the address you want to erase
FLASH_ADDRL = flash_addr.b[1];
FLASH_DATA = in_data;
//-- 状态清除 --
FLASH_ENA = 0x00;
FLASH_ENB = 0x00;
FLASH_ENC = 0x00;
//-- 通过安全检验后才能启动 Flash 操作 --
if((NRM_securty_a == 0xaa)&&(NRM_securty_b == 0x55))
    FLASH_CON = 0x03; //状态清除，执行一次 FLASH_CON，将安全状态复位
_nop_(); //延时
//-- 写安全码开启 Flash 操作许可 --
FLASH_ENA = 0x05;
FLASH_ENB = 0x0a;
FLASH_ENC = 0x09;
//-- 通过安全检验后才能启动 Flash 操作 --
if((NRM_securty_a == 0xaa)&&(NRM_securty_b == 0x55))
    FLASH_CON = 0x03;
//-- 清除安全码 --
FLASH_ENA = 0x00;
FLASH_ENB = 0x00;
FLASH_ENC = 0x00;
EA = ea_save;
}

//EEPROM BLOCK(1k) 擦除
//addr = (0 - 31) * 1024 ,擦除对应的 Block 地址
//调用前需要：
//NRM_securty_a,NRM_securty_b
//flash 操作关闭总中断，操作完后会开启总中断(注意)
void e2rom_erase(unsigned int addr)
{
    union INTpattern flash_addr;
    bit ea_save;
    flash_addr.i = addr;
    ea_save = EA; // Save EA
    EA = 0;
    FLASH_ADDRH = flash_addr.b[0]; // point to the address you want to erase
    FLASH_ADDRL = flash_addr.b[1];
    //-- 状态清除 --
    FLASH_ENA = 0x00;
    FLASH_ENB = 0x00;
    FLASH_ENC = 0x00;
```



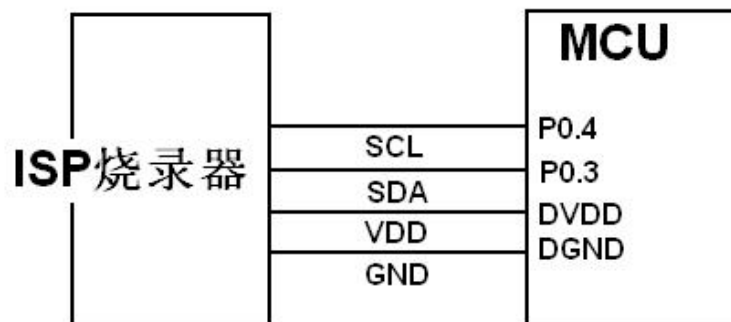
```

/-- 通过安全检验后才能启动 Flash 操作 --
if((NRM_securty_a == 0xaa)&&(NRM_securty_b == 0x55))
    FLASH_CON = 0x03; //状态清除，执行一次 FLASH_CON，将安全状态复位
_nop_();
/-- 写安全码开启 Flash 操作许可 --
FLASH_ENA = 0x05;
FLASH_ENB = 0x0a;
FLASH_ENC = 0x09;
/-- 通过安全检验后才能启动 Flash 操作 --
if((NRM_securty_a == 0xaa)&&(NRM_securty_b == 0x55))
    FLASH_CON = 0x0c;
/-- 清除安全码 --
FLASH_ENA = 0x00;
FLASH_ENB = 0x00;
FLASH_ENC = 0x00;
EA = ea_save;
}

```

## 17 在线 ISP 程序烧录

- ◇ SDI5219 系列 ISP 烧录需要用到 SDA、SCL、DVDD、DGND 四根线（如果 MCU 板自己提供电源的话 DVDD 可以不连）
- ◇ SDA 和 P0.3 复用；SCL 和 P0.4 复用
- ◇ 连接好相关连线后，MCU 上电，完成烧录
- ◇ 烧录时，可进行如下配置：
  - Flash 中“数据空间的大小”
  - 是否采用外置晶振，外置晶振的类型（主震荡或者低频震荡）
  - P1.3 是否为外部复位
  - 程序是否加密
  - 是否保留“数据区的内容”





为了不影响正常烧录，请注意如下事项：

- ✧ SDA、SCL 不要小电阻(5K 以下)下拉到地，否则会影响 ISP 烧录
- ✧ SDA、SCL 不要串接电阻到烧录口
- ✧ P0.1-尽可能避免下拉到地

## 18-电器特性

### 18.1 极限条件



虽然此集成电路带有 ESD 保护电路，但仍然在某些极端条件下的静电放电时遭到损坏。静电放电可能造成整个芯片不工作，也可能对芯片中某些精密电路造成影响，使之不能达到我们公开资料上的效果。因而在使用时应适当避免用手直接接触管脚，防止 ESD 的情况的发生。

极限条件：

参数	典型	单位
AVDD到AGND 压差	-0.3 - 5.5	V
DVDD到DGND 压差	-0.3 - 5.5	V
AGND到DGND 压差	-0.3 - +0.3	V
模拟输入电压	-0.3 - AVDD+0.3	V
数字输入电压	-0.3 - DVDD+0.3	V
最大工作温度范围	-30 - 100	°C
结温	150	°C

### 18.2 直流特性

测试条件（如无特殊说明均采用此条件）：

AVDD = DVDD = VREFP = +3.3V；温度范围：-25 - 80 摄氏度；

参数	符号	测试数据				测试条件 9.83MHz
		最小	典型	最大	单位	
电源电压	DVDD/AVDD	2.0		5.5	V	
正常工作电流	IDD1		2.5		mA	24 位 ADC 开启 (前置运算放大器开启)
STOP2 模式电流	IDD2		130		uA	关闭 RC 震荡
24 位 ADC			1100		uA	PGA= 128
			250		uA	PGA=2
I <sub>O</sub>			55		uA	普通上拉 输出为 0
内部主 RC 频率		9.6MHz	9.83MHz	10.0MHz	V	-40 摄氏度 50 摄氏度
I <sub>O</sub> 口驱动电流						参考：“5.3 I <sub>O</sub> 口驱动电流”
VCOM 输出参考 源温度系数			50		ppm	



## 18.3 ADC 参数

测试条件（如无特殊说明均采用此条件）：

AVDD = DVDD = VREFP = +3.3V；温度范围：-25 - 80 摄氏度；

参数	符号	测试数据				测试条件 9.83MHz
		最小	典型	最大	单位	
差分电压输入 (VINP - VINN)		+/- 0.5 VREFP/128			V	
积分非线性 (INL)	PGA = 2		0.0002	0.001	% of FS	
	PGA = 128		0.0005	0.001	% of FS	
输入失调	PGA = 128		3	5	% of FS	
输入失调温漂			+/-10		$nV/^{\circ}C$	
增益误差			0.01		% of FS	
ADC 精度						参考“14.3 噪声性能”