



**芯易德科技**  
**SOLIDIC**

---

## **SDI5229系列：用户手册**

面向测量应用的8位Flash MCU  
- 集成低功耗24位ADC

V1.5

## 版本更新记录

版本	日期	说明	备注
V1.0	Oct12_19	■	000
V1.2	Dec02_19	■	000
V1.3	Nov30_20	■	000
V1.4	Feb04_21	■	001
V1.5	Feb28_21	■	001

1-主要特征.....	5
2-引脚.....	6
3-特殊功能寄存器 .....	8
4-存储器 .....	9
4.1 RAM.....	9
4.2 FLASH.....	9
5-IO 口.....	10
5.1.1 I/O 口模式配置.....	10
5.2 相关特殊寄存器.....	11
6-定时/计数器 .....	15
6.1特殊寄存器.....	15
6.2 定时/计数器的四种模式.....	17
7-中断 .....	20
7.1相关特殊寄存器.....	20
8-串口(UART).....	23
8.1相关特殊寄存器.....	23
9-SRA 8 位 ADC .....	24
9.1 相关特殊寄存器.....	24
9.2 (SAR) 8 位 ADC 参考程序.....	25
10-时钟系统、电源管理及复位 .....	26
10.1 时钟系统概述.....	26
10.1.3 MCU 的时钟特性.....	27
11-看门狗与低频唤醒 .....	33
11.1 相关寄存器 .....	34
11.2 WDCON 的“访问窗口” .....	35
11.3 “掉电模式”下的看门狗运行 .....	35
12-脉宽调制模块(PWM) .....	36
12.1 相关寄存器 .....	36
12.2 工作说明 .....	37
13-(SIGMA-DELTA) 24 位 ADC.....	38
13.1 配置说明 .....	38
13.2 相关寄存器 .....	42

13.3 噪声性能: .....	46
<b>14-温度传感器 .....</b>	<b>47</b>
14.1 温度传感器的转换数值 .....	47
14.2 相关寄存器 .....	47
14.3 温度测量样例程序 .....	47
<b>15-FLASH 操作说明 .....</b>	<b>51</b>
15.1 FLASH 数据区的“读” .....	52
15.2 相关寄存器 .....	52
15.3 “FLASH 数据区”的操作保护 .....	53
15.4 “FLASH 数据区”的“擦除” .....	53
15.5 “FLASH 数据区”的“写入” .....	54
15.6 FLASH 的抗干扰程序样例 .....	55
<b>16-升压模块于液晶驱动 .....</b>	<b>57</b>
16.1 升压模块 .....	57
16.2 液晶驱动模块 .....	58
<b>17-I2C 接口 .....</b>	<b>65</b>
17.1 寄存器说明 .....	65
<b>18-在线 ISP 程序烧录 .....</b>	<b>67</b>
<b>19-电器特性 .....</b>	<b>67</b>
19.1 极限条件 .....	67
19.2 直流特性 .....	68
19.3 ADC 参数 .....	68
<b>20-封装信息 .....</b>	<b>69</b>
20.1 LQFP48 (7.0*7.0*1.40) 外形尺寸 .....	69
20.2 SSOP24 (0.635-D1.4) 外形尺寸 .....	69

## 面向测量应用的 8 位 Flash MCU: SDI5229 系列

### 主要特征:

- **内核:**
  - 1T 指令周期 (8051 单片机兼容)
  - 可配置代码专利安全模式, 强抗干扰
- **程序空间: 30K 字节 Flash**
  - Flash, 超过100,000 次的烧写寿命
  - 内置ISP功能 (SDA、SCL两线烧录)
  - 可配置用户ISP
- **RAM: 768 字节**
  - 256 Bytes 内部RAM
  - 512 Bytes 内嵌外部寻址RAM (XDATA)
- **时钟:** (主震荡、看门狗均可采用外部晶振)
  - 9.83MHz 内部RC震荡 (可倍频至19.66MHz)
  - 32KHz 内部低频时钟
  - ◇ 可实现4ms~1s等不同间隔唤醒MCU
- **电源/低功耗:**
  - 工作电压: 1.8V - 5.5V
  - 工作功耗:
    - ◇ < 1.5mA (@3.3V, 9.83MHz)
    - ◇ < 300uA (@3.3V, 600KHz)
  - 休眠功耗:
    - ◇ 通常STOP2模式: < 2uA
    - ◇ 关机模式: < 0.2uA
- **主要外设:**
  - 3通道24位Sigma-Delta高精度低电压ADC
    - ◇ 工作电压: 1.55V - 3.3V
    - ◇ **数据快速稳定:** 工作后第1个数据就稳定
    - ◇ **基准电压输出** (1.55v、2.0v、2.4v、3.0v)
    - ◇ **基准电流输出** (50uA - 1mA)
    - ◇ 温度传感器
  - 液晶驱动:
    - ◇ 可配置80点 (4\*20) 的液晶驱动
    - ◇ 可配置128点 (8\*16) 的液晶驱动
    - ◇ **低功耗电容模式液晶驱动**
  - 内置升压模块:
    - ◇ 正常模式: (2.5v、2.8v、3.3v、5.0v)
    - ◇ 低功耗模式: 3.3V
    - ◇ 升压模块可配置为LDO模式
  - 8通道8位低精度ADC
- **其他:**
  - I/O口: 35个通用IO、可配置IO驱动强度
  - 3个16 位定时/计数器
  - 13个中断源, 3级优先级: 可配置掉电中断
  - 通讯接口: 1组UART、1组I2C接口
  - 看门狗集成: 定时唤醒、计时功能
  - 2路16位脉宽调制 (PWM) 输出

### 应用系统:

- 电子秤、气压计
- 数字压力传感器;
- 血压计等医疗仪器;
- 微弱信号测量及工业控制;

### 型号及封装:

型号	封装	说明		
		Flash	RAM	其他
SDI5229T	LQFP48	30KByte	768Byte	
SDI5209P-X2	SSOP24	30KByte	768Byte	

## 2-引脚

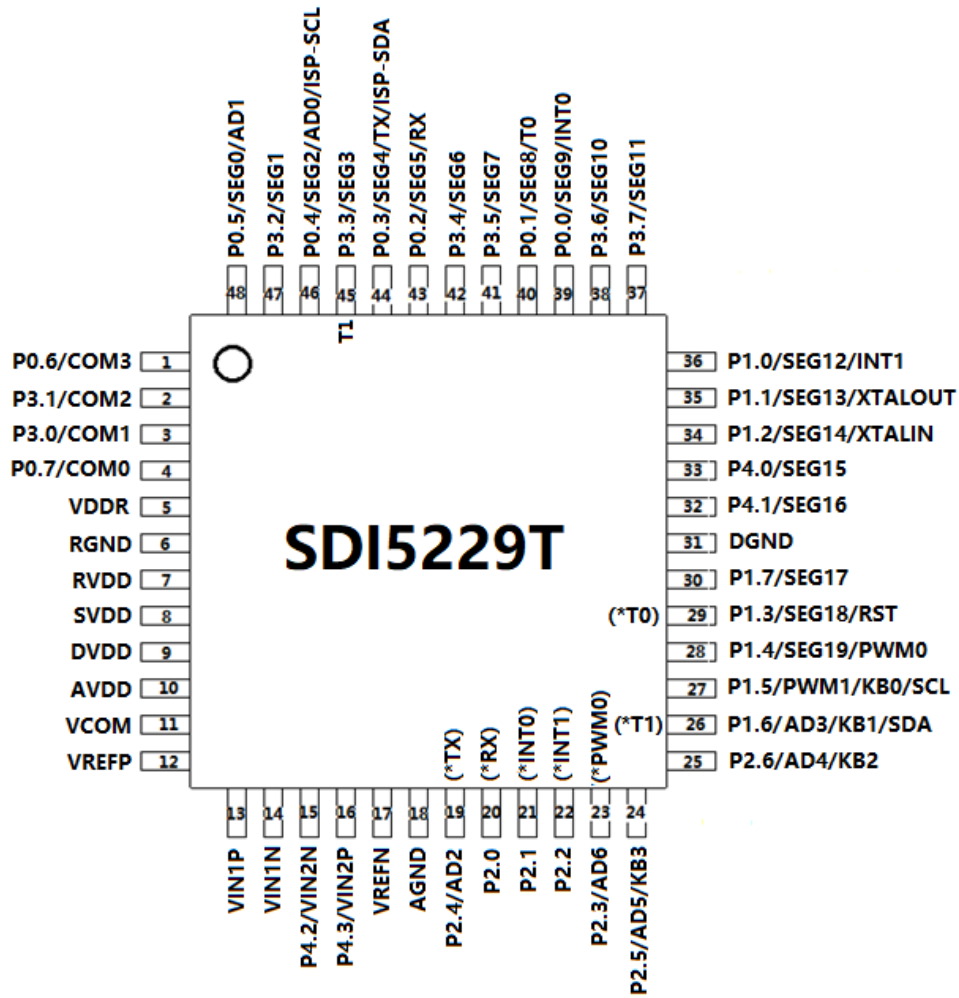


图 2-1: SDI5229T 管脚图

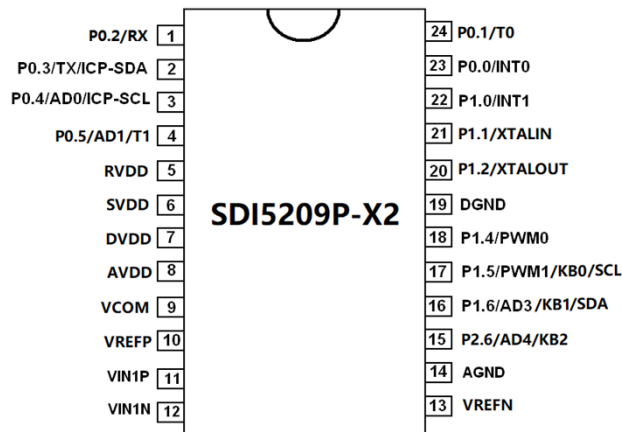


图 2-2: SDI5209P-X2 管脚图

表 2-1: 管脚说明

	类型	说明
<b>电源、地</b>		
DVDD	P	数字电源
SVDD	P	升压输入: 由升压输入电源串接电感到 SVDD
RVDD	P	1: 数字 IO 电源: 给“P0、P1、P2、P3”, “P4.0\4.1”供电。 2: 内部升压输入; 如不用升压, 可直接将 SVDD 和 RVDD 的关系配置为 LDO, 或者直接相连
AVDD	P	模拟电源: 给模拟部分供电, 给 IO“P4.2\4.3”供电
RGND	P	数字地
DGND	P	数字地
AGND	P	模拟地
VDDR	P	内部数字电路电源滤波输出管脚: -- 通常为 1.8V, 可外接 10nF 或者 100nF 电容
<b>普通 IO</b>		
P0.0 – P0.7	I/O	普通 IO (由数字电源 RVDD 供电)
P1.0 -- P1.7	I/O	普通 IO (由数字电源 RVDD 供电)
P2.0 – P2.6	I/O	普通 IO (由数字电源 RVDD 供电)
P3.0 – P3.7	I/O	普通 IO (由数字电源 RVDD 供电)
P4.0 – P4.1	I/O	普通 IO (由数字电源 RVDD 供电)
P4.2 – P4.3	I/O	普通 IO (由数字电源 AVDD 供电)
<b>24 位 ADC 相关 IO</b>		
VCOM	O	内部 LDO 输出, 可配置为: 1.55v\2.0v\2.4v\3.0v\AVDD 输出 <b>建议 VCOM 外挂 10nF 的电容 (无电容也没关系)</b>
VREFP	I	正参考电压输入(24 位 ADC)
VREFN	I	负参考电压输入(24 位 ADC)
VIN1P	I	差分输入 1 通道: 正端信号
VIN1N	I	差分输入 1 通道: 负端信号
VIN2P	I	差分输入 2 通道: 正端信号 (复用 P4.3)
VIN2N	I	差分输入 2 通道: 负端信号 (复用 P4.2)
<b>液晶驱动 IO</b>		
COM0 - COM3	O	液晶 COM 信号: 开启液晶模式即有效
SEG0 - SEG3	O	液晶 SEG 信号: 开启液晶模式即有效 -- 8COM 模式时, 自动切换为变为 COM4 - COM7
SEG4 - SEG11	O	液晶 SEG 信号: 开启液晶模式即有效
SEG12 - SEG19	O	液晶 SEG 信号: 可以手动配置是否开启液晶模式还是普通 IO
<b>其他复用 IO</b>		
PWM0\PWM1	I/O	16 位脉宽调制模块输出 (复用 P1.5、P1.4) <b>1: 当 P1.4 被配置为 SEG19 使用时, PWM0 自动切换到 P2.3</b>
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\INT1	I/O	外部中断 0\1 (复用 P0.0\P1.0) <b>1: 烧录配置“第二复用模式”, INT0\1 切换到 P2.1\P2.2 复用</b>
RX\TX	I/O	UART 通讯接口: 接收\发送信号(复用 P0.2\P0.3) <b>1: 烧录配置“第二复用模式”, RX\TX 切换到 P2.0\P2.4 复用</b>
SCL\SDA	I/O	I2C 通讯接口: 时钟\数据 (复用 P1.5\P1.6)

AD0 - AD6	I	8 位 ADC 的输入 (复用:P0.4\P0.5\P2.4\P1.6\P2.6\P2.5\P2.3)
T0\T1	I	定时器 T0/T1 作为计数器其使用时的外部计数脉冲输入端 <b>1: 烧录配置“第二复用模式”, T0\T1 切换到 P1.3\P1.6 复用</b>
<b>ISP 烧录信号</b>		
ISP-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、电源)
ISP-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、电源)

### 3-特殊功能寄存器

表 3-1 寄存器地址列表

	8	9	A	B	C	D	E	F
F8	EIP	LCD_DAT0 *						
F0	B							
E8	EIE	STATUS	IODRV_CON	LCD_CON	LCD_CLKSEL	PFCON #	PFDATA #	
E0	ACC							IOM_LOCK
D8	EICON							
D0	PSW		SGADCON5X	SGADCON3X	SGADCON4X #	PDM0_IOM #	IOLCD_CON	WDCON2 #
C8	I2CCON	I2CIN	I2COUT	***	***	TL2	TH2	I2CCON2
C0	P4	FLASH_DATA	FLASH_ADDL	FLASH_ADDH	FLASH_ENA *	FLASH_ENB *	FLASH_ENC *	FLASH_CON
B8	IP	PSTATE	***	***	ICPENB #	ICPENB #	ICPENC #	PCON3 #
B0	P3	SGADCON	SGADC3	SGADC2	SGADC1	***	***	***
A8	IE	WDCON #	WD_TA * #	SARCON	SARDATA	PWM0_H	PWM1_H	***
A0	P2	SGADCON2 #	PD_CON	保留	保留	保留	保留	PCON2 #
98	SCON	SBUF	PWMF_H	PWMF_L	PWM0	PWM1	PWM_CON	P4M0
90	P1	EXIF	P0M0	P0M1	P1M0	P1M1	P2M0	P2M1
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	P3M0
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  
#: PDM0 电源域内的寄存器



## 4-存储器

### 4.1 RAM

对于SDI5229系列，内建 768 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~1FFh 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](#)。

SDI5229系列RAM空间:

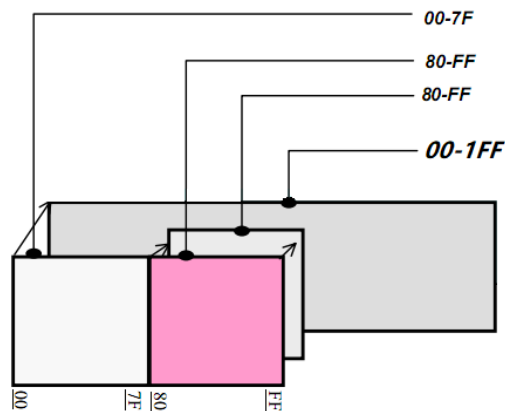


图4-1: RAM空间

### 4.2 FLASH

SDI5229系列的用户Flash空间在烧录时可分配为：“用户程序空间”、“AIP数据空间”以及“用户ISP空间”

- ✧ 用户程序空间：用户程序存储空间
- ✧ AIP 数据空间：非易失性数据存储空间，程序可修改此空间数据
- ✧ 用户 ISP 空间：用于存放用户引导程序，开机后先运行用户引导程序，执行完成后，在跳转到用户程序执行。

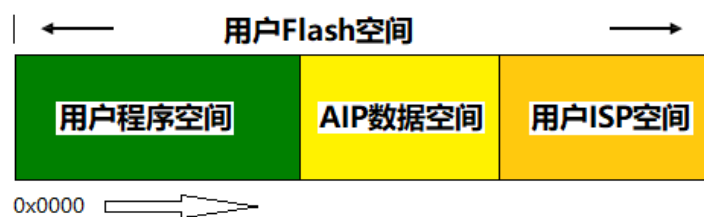


图 4-2: Flash 空间

## 5-IO 口

### ■ 特征概述:

- ◇ 35个通用IO, 且可配置“CMOS推拉输出”模式的IO上拉驱动强度
- ◇ IO可配置LCD模式, 用于驱动液晶
- ◇ 注意: 除P4.2\4.3由AVDD供电外, 其他均有RVDD供电
- ◇ 注意: 为了避免IO模式改动, 某些重要IO的模式可以被锁定

### 5.1 IO模式及配置说明

SDI5229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(高阻)
1	1	集电极开路

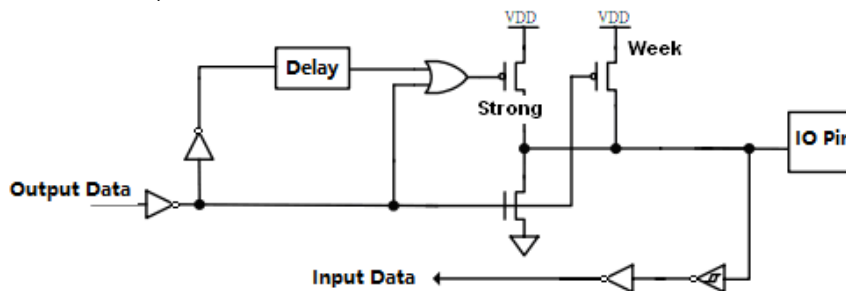
注意: 配置为“仅输入”“集电极开路”时, IO口的对应“数据寄存器”必须置1, 否则IO口将被拉到地。

要注意: IO口的配置锁定功能 以及 液晶驱动模式下的IO口

#### 5.1.1 I/O 口模式配置

##### ■ 标准51 输出模式

PxM1n= 0, PxM0n = 0

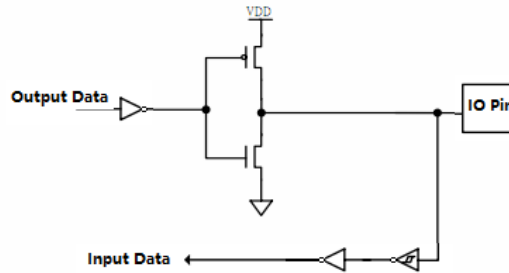


说明:

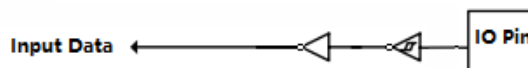
- ◇ 输出数据上升沿会启动100ns的强上拉, 然后强上拉关闭, 只剩下弱上拉
- ◇ 弱上拉相当50K的电阻。此配置下, IO口输出低电平, 将持续有电流消耗。低功耗是需要注意!!

##### ■ CMOS推拉输出

PxM1n = 0; PxM0n = 1;



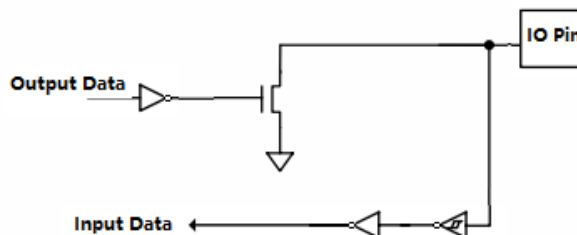
- 仅输入模式(高阻)  
PxM1n = 1; PxM0n = 0;



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;



## 5.2 相关特殊寄存器

### 5.2.1 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
P07	P06	P05	P04	P03	P02	P01	P00

- POM0(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
POM07	POM06	POM05	POM04	POM03	POM02	POM01	POM00

- POM1(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
POM17	POM16	POM15	POM14	POM13	POM12	POM11	POM10

### 5.2.2 P1口: P1.1\|P1.2\|P1.4\|P1.5 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
P17	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
P1M07	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00

■ **P1M1(0x95):** P1 口模式配置寄存器1 (默认值: 0x36)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
P1M17	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10

5.2.3 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
	P26	P25	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
	P2M06	P2M05	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
	P2M16	P2M15	P2M14	P2M13	P2M12	P2M11	P2M10

5.2.4 P3口: 默认配置“标准51输出模式”

注意: P3\_MOD (IOM\_LOCK[2]) 控制 P3 口的 IO 模式

■ **P3 (0xB0):** P3 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
P37	P36	P35	P34	P33	P32	P31	P30

■ **P3M0(0x8F):** P3 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
P3M07	P3M06	P3M05	P3M04	P3M03	P3M02	P3M01	P3M00

◇ P3\_MOD: 0 P3 口配置为“标准51输出模式”和“CMOS 推拉输出”  
通过 P3M0 对应位控制:

- 0: “标准51输出模式”
- 1: “CMOS 推拉输出”

◇ P3\_MOD: 1 P3 口配置为“仅输入(高阻)”和“CMOS 推拉输出”  
通过 P3M0 对应位控制:

- 0: “仅输入(高阻)”
- 1: “CMOS 推拉输出”

5.2.5 P4口: 默认配置“标准51输出模式”

注意: P3\_MOD (IOM\_LOCK[2]) 控制 P4.0\P4.1 口的 IO 模式

■ **P4 (0xC0):** P4 口数据寄存器 (默认值: 0x0F)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				P43	P42	P41	P40

■ **P4M0(0x9F):** P4 口模式配置寄存器0 (默认值: 0x30)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		P4M13	P4M12	P4M03	P4M02	P4M01	P4M00

- ◇ P3\_MOD: 0      P4.0\P4.1 配置为“标准51 输出模式”和“CMOS 推拉输出”  
通过 P4M0 对应位控制:
  - 0: “标准51 输出模式”
  - 1: “CMOS 推拉输出”
- ◇ P3\_MOD: 1      P4.0\P4.1 配置为“仅输入(高阻)”和“CMOS 推拉输出”  
通过 P4M0 对应位控制:
  - 0: “仅输入(高阻)”
  - 1: “CMOS 推拉输出”
- ◇ P4M0[5]\P4M0[3]; P4M0[4]\P4M0[2] 分别组成 {P4M13、P4M03}、{P4M12、P4M02} 控制 P4.2、P4.3 的 IO 模式:
  - 00      集电极开路
  - 01      集电极开路  
(带内部高电平钳位, 电平不能高于 1.5V);
  - 10      “仅输入(高阻)”;
  - 11      “CMOS 推拉输出”, 上拉驱动<0.3mA  
(带内部高电平钳位, 不能高于 1.5V);

### 5.2.6 IO 锁定及液晶模式

■ PD\_CON(0xA2): 休眠辅助寄存器 (默认值: 0x02)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
XIO_LCK	P4_LCK1	P4_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)

P4\_LCK1 \P4\_LCK0: P4.2\P4.3 口 IO 锁定

- 11、10、01: 锁定 P4.2、P4.3 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, sgadcon3x, sgadcon4x, sgadcon5x  
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位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

■ IOM\_LOCK(0xE7): 液晶 IO 锁定寄存器 (默认值: 0x01)

注意: 修改 IOM\_LOCK 条件: 要开启 WD\_TA 时间窗口

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				SEG19_EN	P3_MOD	IOM_LK[1:0]	

- SEG19\_EN: 控制 P1.4 是否配置为液晶驱动的 SEG19  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG19\_EN == 1)
- P3\_MOD: P3 口\P4.0\P4.1 的模式切换:  
具体参考: P3M0 和 P4M0
- IOM\_LK[1:0]: IOLCD\_CON 的控制寄存器  
10: 允许 IO 被配置为 LCD 模式 (IOLCD\_CON 配置生效)  
其他: 允许 IO 被配置为 LCD 模式 (忽略 IOLCD\_CON 的配置)

■ IOLCD\_CON(0xD6): 液晶 IO 配置寄存器 (默认值: 0x00)

注意: 修改 IOLCD\_CON 条件: 首先配置 IOM\_LK (IOM\_LOCK[1:0]), 允许 IO 被配置为 LCD 模式。同时要开启 WD\_TA 时间窗口

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SEG18_EN	SEG17_EN	SEG16_EN	SEG15_EN	SEG14_EN	SEG13_EN	SEG12_EN	LCDIO_EN

- SEG18\_EN: 控制 P1.3 是否配置为液晶驱动的 SEG18  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG18\_EN == 1)
- SEG17\_EN: 控制 P1.7 是否配置为液晶驱动的 SEG17  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG17\_EN == 1)
- SEG16\_EN: 控制 P4.1 是否配置为液晶驱动的 SEG16  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG16\_EN == 1)
- SEG15\_EN: 控制 P4.0 是否配置为液晶驱动的 SEG15  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG15\_EN == 1)
- SEG14\_EN: 控制 P1.2 是否配置为液晶驱动的 SEG14  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG14\_EN == 1)
- SEG13\_EN: 控制 P1.1 是否配置为液晶驱动的 SEG13  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG13\_EN == 1)
- SEG12\_EN: 控制 P1.0 是否配置为液晶驱动的 SEG12  
满足如下三个条件: (IOM\_LK == 2) (LCDIO\_EN == 1) (SEG12\_EN == 1)
- LCDIO\_EN: 开启 IO 液晶模式  
满足如下两个条件, 则配置相关 IO 为 SEG0 - SEG11、COM0 - COM3  
(IOM\_LK == 2) (LCDIO\_EN == 1)

5.2.7 键盘中断及IO驱动能力

■ IODRV(0xEA): 液晶 IO 配置寄存器 (默认值: 0x7f)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	KBI_MASK[4:0]					iodrv_con[1:0]	

KBI\_MASK[4:0]: 键盘中断允许控制: 为 0 允许响应的信号产生中断请求, 为 1 禁止  
 KBI\_MASK[4]: KB5: 对应信号为: 震动检测电路的输出信号  
 KBI\_MASK[3]: KB0: 对应信号 IO 为 P1.5  
 KBI\_MASK[2]: KB1: 对应信号 IO 为 P1.6  
 KBI\_MASK[1]: KB2: 对应信号 IO 为 P2.5  
 KBI\_MASK[0]: KB3: 对应信号 IO 为 P2.6

iodrv\_con[1:0]: 控制“CMOS 推拉输出”模式下, 高电平输出驱动能力:  
 (11: 最弱) -> (00: 最强)

## 6-定时/计数器

(相关中断, 请参考第7章“7-中断”)

### ■ 特征概述:

- ◇ SDI5222系列提供了三个16 位定时/计数器 T0、T1、T2
- ◇ T0时钟可配置为: “CLK\_OSC/12”、“CLK\_OSC/4”
- ◇ T1时钟可配置为: “CLK\_OSC/12”、“CLK\_OSC/4”、“CLK\_OSC”
- ◇ T2时钟固定为: “CLK\_OSC/4”
- ◇ 注意: 使用内部震荡提供串口时钟时, 定时器尽量采用“CLK\_OSC/4”或者“CLK\_OSC”, 这样能配置各种串口速率。

### 6.1 特殊寄存器

#### 6.1.1 相关控制寄存器

##### ■ 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, TLO 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：0x02）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		CKDIV[2]	T1M[1:0]		T0M	CKDIV[1:0]	

T0M: 定时器T0 时钟选择  
0: CLK\_MCU /12  
1: CLK\_MCU /4

T1M[1:0]: 定时器T1 时钟选择  
00: CLK\_MCU /12  
01: CLK\_MCU /4  
10/11: CLK\_MCU

CKDIV[2:0]: 用于MCU时钟分频  
000: CLK\_MCU = CLK\_OSC  
010: CLK\_MCU = CLK\_OSC / 2 (默认)  
001: CLK\_MCU = CLK\_OSC / 4  
011: CLK\_MCU = CLK\_OSC / 6  
100: CLK\_MCU = CLK\_OSC / 8  
101: CLK\_MCU = CLK\_OSC / 16

■ EICON(0xD8)（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
TF2	TR2	EPFI	PFI	WDFI	CLIP	CLPCS	WTFI

TF2: T2 溢出标志位  
当T2 溢出时，该位会自动置1。 需要手动清0

TR2: 0: 停止T2  
1: 开始T2

EPFI: 掉电中断允许位:  
0: 禁止掉电中断，电压监测BOD只用复位  
1: 允许掉电中断，电压监测BOD只用产生中断

PFI: 掉电中断标志位: 掉电中断发生时置1，需要手动清0

WDFI: “看门狗-计数器”溢出，该位会自动置1。 需要手动清0



- CLIP: 写1清除中断全部中断嵌套屏蔽 (IE[6]为1时有效)  
只持续1个CLK的时间, 自动清0
- CLPCS: 写1清除中断地址堆栈 (IE[6]为1时有效, 且安全模式下有效)  
只持续1个CLK的时间, 自动清0
- WTFI: “看门狗-计时器”溢出, 该位会自动置1. 需要手动清0

### 6.1.2 计数寄存器

- **TLO(0x8A)** 定时器0低8位计数器 (默认值: 0x00)

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

- **TL1(0x8B)** 定时器1低8位计数器 (默认值: 0x00)

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

- **TH0(0x8C)** 定时器0高8位计数器 (默认值: 0x00)

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

- **TH1(0x8D)** 定时器1高8位计数器 (默认值: 0x00)

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

- **TL2(0xCD)** 定时器2低8位计数器 (默认值: 0x00)

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

- **TH2(0xCE)** 定时器2高8位计数器 (默认值: 0x00)

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

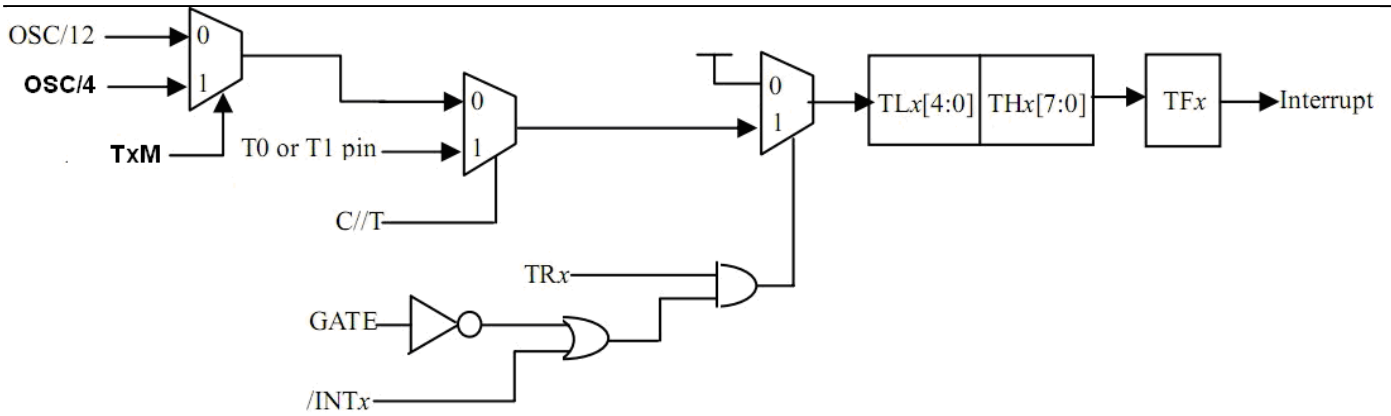
注意: 定时器2只有一种16位模式, 固定采用4分频 “CLK\_OSC /4”

## 6.2 定时/计数器的四种模式

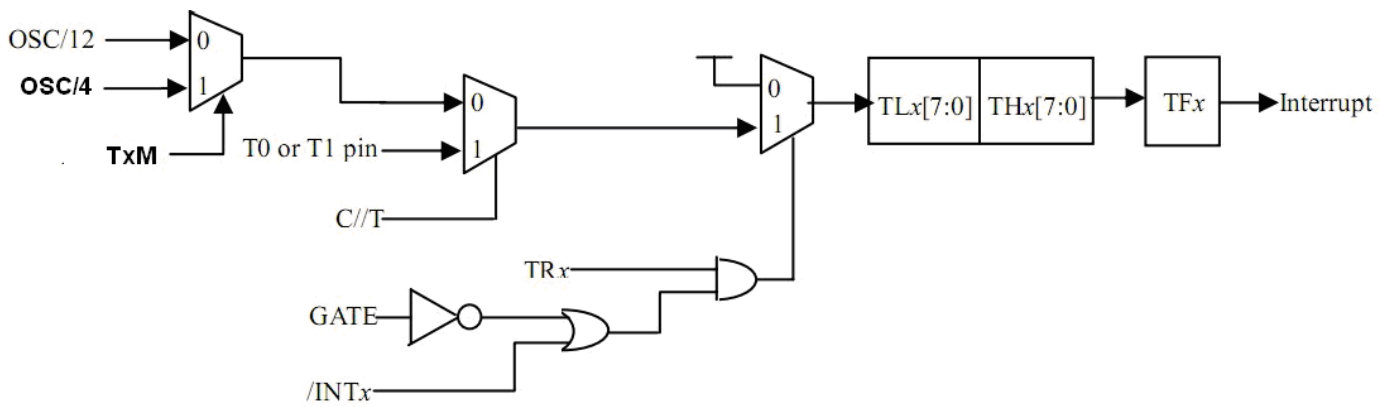
参考51定时器的相关定时器说明

下图中: T0(P0.1)、T1(P3.3)、INTx (INT0\INT1)

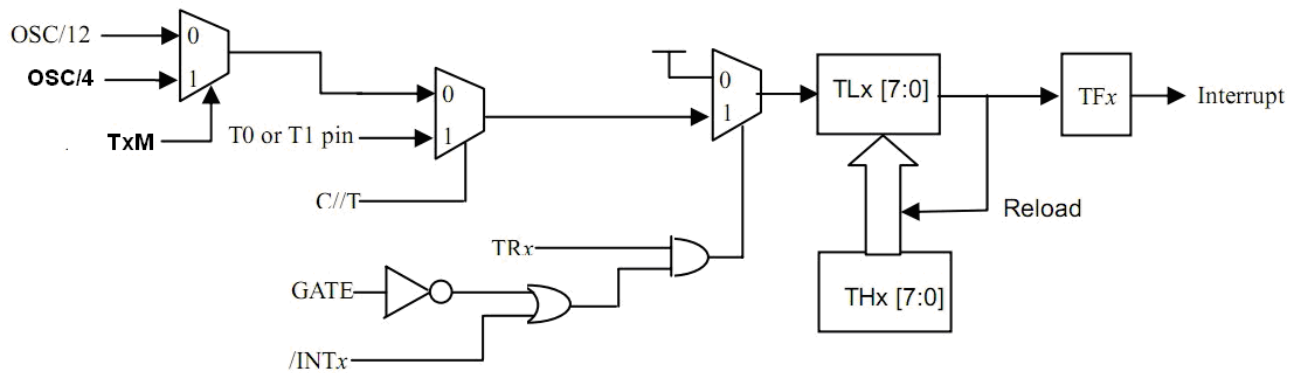
- M1,M0 = 0,0 : 模式0  
13 位定时/计数器



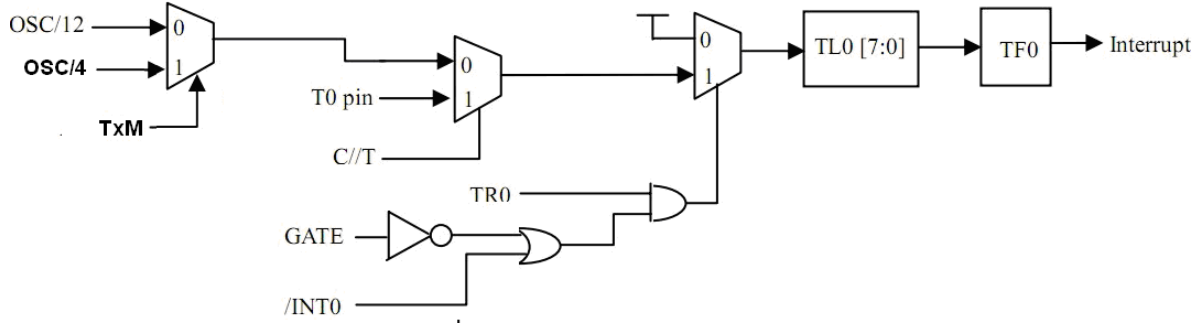
- M1,M0 = 0,1 : 模式1  
16 位定时/计数器



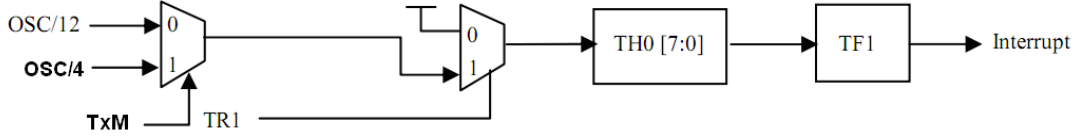
- M1,M0 = 1,0 : 模式2  
8 位自动重载定时/计数器



- M1,M0 = 1,1 : 模式3  
TLO 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

SDI5229系列，提供13个中断源，4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
掉电中断	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
键盘中断	53H	11
定时器2	5BH	12
I2C中断	63H	13

### ■ 特征概述:

- ◇ 掉电中断具有最高优先级2；其他中断根据优先级配置为1或者0；优先级顺序为2>1>0
- ◇ “外部中断0、1” “键盘中断”可配置响应电平：高电平或者低电平响应；上升或者下降沿响应；
- ◇ 看门狗中断包含两部分：“看门狗中断”和“低频计时中断”，他们共用1个中断向量资源，具有共同的有优先级，采用不同的标志位区分，并有不同的使能位。
- ◇ “中断嵌套清除”以及“中断堆栈清除”（这两项通常情况下并不需要使用）
  - ❖ “中断嵌套清除”：执行中断程序是，不能被同优先级的中断打断。如果在强干扰情况下，中断程序A没有执行返回就跳出进入了主循环，这时将一直无法响应同优先级的中断。所以，可以在进入主循环时，清除中断占用标志。
  - ❖ “中断堆栈清除”：用于“安全代码模式下”  
中断异常跳出，清空中断返回地址堆栈
- ◇ 可以通过IIP\_LO\IIP\_L1\IIP\_PF（STATUS[4:2]）读取到“实时中断优先级标志”，即正在发生的中断优先级

### 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	CLEN	ET2	ES	ET1	EX1	ET0	EX0

EA:           0: 禁止所有中断  
              1: 使能中断

- CLEN: 允许进行中断异常控制:  
(CLIP) 清除中断嵌套调用屏蔽; (CLPCS)安全模式下清除中断地址堆栈  
0: 禁止  
1: 使能
- ET2: 定时器2 中断使能位  
0: 禁止  
1: 使能
- 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
		PT2	PS	PT1	PX1	PT0	PX0

IP[5:0] 配置 IE[5:0] 中中断的优先级, 对应位为 1, 则优先级为 1; 为 0 则优先级为 0  
优先级 1 高于优先级 0

■ 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
	EI2C	EWTI	EWDI	EX5	EX4	EX3	EX2

- EI2C: “I2C接口中断” 使能位  
0: 禁止  
1: 使能
- EWTI: “低频计时中断” 使能位  
0: 禁止  
1: 使能
- 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
	PI2C		PWDI	PX5	PX4	PX3	PX2

◇ EIP[5:0] 配置 EIE[5:0] 中中断的优先级，对应位为 1，则优先级为 1；为 0 则优先级为 0

◇ 注意：“看门狗中断”和“低频计时中断”使用同样的优先级 PWDI

■ 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	IE2	IL2	IL1	IL0	

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。

IE2: “键盘中断”标志

当“键盘中断”产生时，该位会自动置1，该位需要手动清零。

IL2: “键盘中断”的响应电平

0: 低电平响应

1: 高电平响应

IL1: “外部中断1”的响应电平

0: 低电平响应

1: 高电平响应

IL0: “外部中断0”的响应电平

0: 低电平响应

1: 高电平响应

■ EICON(0xD8)（默认值：0x00）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
TF2	TR2	EPFI	PFI	WDFI	CLIP	CLPCS	WTFI

TF2: T2 溢出标志位

当T2 溢出时，该位会自动置1。 需要手动清0

TR2: 0: 停止T2

1: 开始T2

EPFI: 掉电中断允许位:

0: 禁止掉电中断，电压监测BOD只用复位

1: 允许掉电中断，电压监测BOD只用产生中断

PFI: 掉电中断标志位: 掉电中断发生时置1，需要手动清0

WDFI: “看门狗-计数器”溢出，该位会自动置1。 需要手动清0

CLIP: 写1清除中断全部中断嵌套屏蔽（IE[6]为1时有效）

只持续1个CLK的时间，自动清0

CLPCS: 写1清除中断地址堆栈（IE[6]为1时有效，且安全模式下有效）

只持续1个CLK的时间，自动清0

WTFI: “看门狗-计时器”溢出，该位会自动置1。 需要手动清0

## 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”)

### ■ 特征概述:

- ◇ 注意：使用内部震荡提供串口时钟时，定时器尽量采用“CLK\_OSC /4”或者“CLK\_OSC”，这样能配置各种串口速率。
- ◇ 可以通过UART\_T\UART\_R (STATUS[1:0]) 读取UART接口状态标 (接收和发送是否在进行)

### 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”)

### ■ 特征概述:

- ◇ SDI5229系列提供了8通道的ADC，其中一路采用内置的分压电阻。
- ◇ 当相关IO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。
- ◇ 参考电压可以选择 DVDD 和 ALDO 的输出（通常为VCOM）

### 9.1 相关特殊寄存器

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

SAR ADC的参考电压为(SARVREF-DGND)

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

SARVREF 是ADC的参考电压，可选择DVDD或者VCOM

#### ■ 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
SAR_SEL[2]	SAR_PSEL	SAR_FCON[1:0]		SAREN	SAR_STA	SAR_SEL[1:0]	

SAR\_PSEL: ADC参考电压选择

0 ADC参考电压选择: **SARVREF = DVDD**

1 ADC参考电压选择: **SARVREF = VCOM** (VCOM为可作参考电压)

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

0,0 160个CLKMCU 时钟周期 (完成转换)

0,1 320个CLKMCU 时钟周期 (完成转换)

1,0 640个CLKMCU 时钟周期 (完成转换)

1,1 40个CLKMCU 时钟周期 (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\_SEL[2:0]: 通道选择位

000 (AD0通道)设置P0.4 为ADC 通道(default)

001 (AD1通道)设置P0.5 为ADC 通道

010 (AD2通道)设置P2.4 为ADC 通道

011 (AD3通道)设置P1.6 为ADC 通道

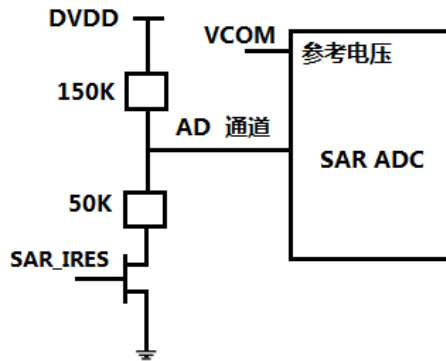
100 (AD4通道)设置P2.6 为ADC 通道

101 (AD5通道)设置P2.5 为ADC 通道

110 (AD6通道)设置P2.3 为ADC 通道



- 111 (AD7通道)选择内部分压电阻通道
- ◇ 内部分压电阻（信号大小为：DVDD/4），配合参考电压选择VCOM，可通过内部分压电阻测试DVDD的电压，如下图，选同时内部开关才打开



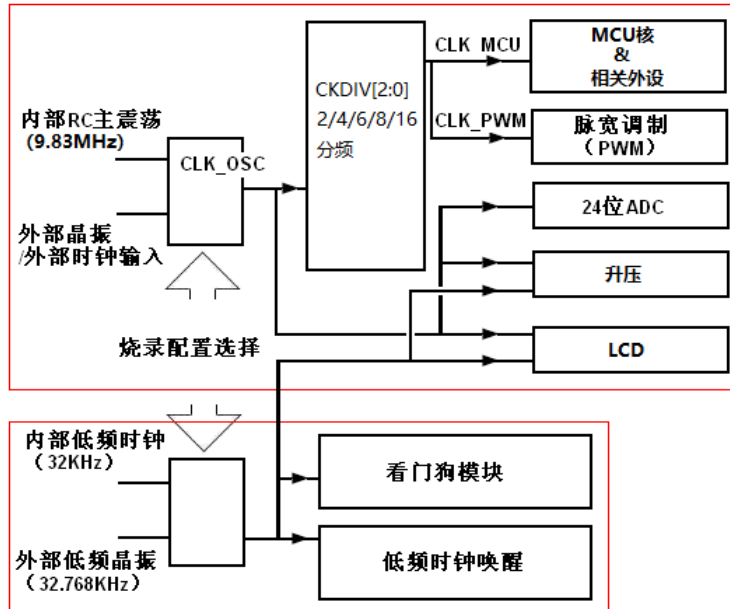
## 9.2 (SAR) 8位ADC参考程序

```

//--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、电源管理及复位

## 10.1 时钟系统概述



SDI5229 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）

### ■ 特征概述:

- ◇ 主(RC)震荡: 9.83MHz
  - ❖ CKDIV 默认两分频
- ◇ 内部低频(RC)震荡: 32KHz
- ◇ 烧录配置:
  - ❖ 配置是否使用外部振荡、外置振荡用于“主震荡”还是“低频震荡”
  - ❖ 外部震荡不起振时，将自动切换到内部RC震荡（前提是低频振荡器不能关闭），可以通过XOSC\_OK（STATUS[5]）读到外置震荡是否启启振
  - ❖ 主震荡是否采用倍频（如：可将9.83MHz倍频到19.66MHz）
- ◇ 注意 MCU 各种休眠状态时的时钟:
  - ❖ IDLE 模式: 不关闭各个时钟
  - ❖ STOP1 模式: 关闭 CLK\_MCU
  - ❖ STOP2 模式: 关闭 CLK\_OSC

### 10.1.1 相关寄存器

■ 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
		CKDIV[2]	T1M[1:0]		TOM		CKDIV[1:0]

TOM: 定时器TO 时钟选择

	0:	CLK_MCU /12
	1:	CLK_MCU /4
T1M[1:0]:	定时器 T1 时钟选择	
	00:	CLK_MCU /12
	01:	CLK_MCU /4
	10/11:	CLK_MCU
CKDIV[2:0]:	用于MCU时钟分频	
	000:	CLK_MCU = CLK_OSC
	010:	CLK_MCU = CLK_OSC / 2 (默认)
	001:	CLK_MCU = CLK_OSC / 4
	011:	CLK_MCU = CLK_OSC / 6
	100:	CLK_MCU = CLK_OSC / 8
	101:	CLK_MCU = CLK_OSC / 16

■ STATUS(0xE9): 状态寄存器 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		XOSC_OK	IIP_L0	IIP_L1	IIP_PF	UART_T	UART_R

XOSC_OK:	外部震荡启振标志
0:	外部震荡没启振 (没有开启外部震荡时也是0)
1:	外部震荡启振
IIP_L0\IIP_L1\IIP_PF:	实时中断优先级标志
IIP_L0 = 1:	表示有中断优先级0的中断正在发生
IIP_L1 = 1:	表示有中断优先级1的中断正在发生
IIP_PF = 1:	表示有掉电中断正在发生
UART_T\UART_R:	UART接口状态标志:
✧	UART_T表示串口正在接收, 只要写SBUF就会马上置1, 直到发送完毕清0
✧	UART_R表示串口正在发送:
❖	置1:
	模式0: 写SBUF0(清除动作)的周期的下一周期末置1
	模式1/2/3: 检测到“RX”下降沿马上就置1;
❖	清0: 接收完毕清0

### 10.1.2 外部震荡

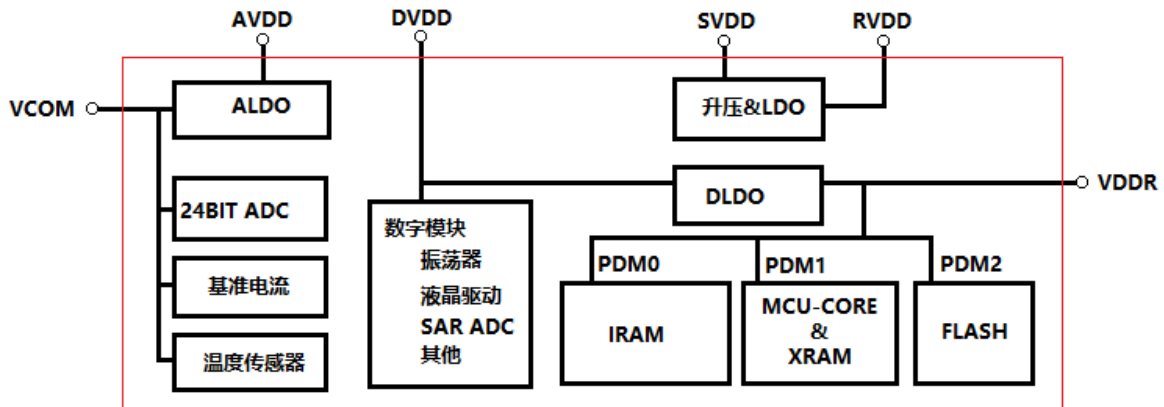
当配置位外部震荡时 (以下两个时钟均适合):

- ✧ P1.1,P1.2 为连接外部晶振的管脚, 分别接 12pF 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, 直到外部震荡重新起振。
- ✧ “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计时功能

### 10.1.3 MCU 的时钟特性

- ✧ SDI5229 系列采用 1T 模式, 1 个机器周期对应 1 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

## 10.2 MCU 电源系统及管理模式



SDI5229 电源系统如上图：

### ■ 特征概述：

- ◇ AVDD：给模拟部分供电  
模拟部分ALDO生产稳压电源（1.5V - 3.0V）给“24位ADC供电”并从VCOM输出
- ◇ DVDD：数字内部模块供电电源，数字DLDO给数字逻辑供电（1.8V），并从VDDR输出  
数字逻辑分为三块，可在采用不同的关闭模式，以降低漏电电流：
  - ❖ STOP2： PDM0、PDM1、PDM2均不关闭
  - ❖ STOP2D： PDM0、PDM1开启；PDM2关闭；
  - ❖ STOP2R： PDM0开启；PDM1、PDM2关闭；
  - ❖ 关机模式： 直接关闭DLDO（此时VDDR为0）
- ◇ SVDD：SVDD到RVDD可采用三种模式：升压、LDO、内部开关直连
  - ❖ 升压时：从 DVDD 串 2.2uH 或 4.7uH 电感到 SVDD；RVDD 接滤波电容（建议 >1uF）；
  - ❖ 即便处于串接电感的电路，也可以直接切换到 LDO 模式
  - ❖ 不采用“升压”、“LDO”模式，或者休眠关闭升压模块时，均处于内部开关直连模式
- ◇ VDDR：数字电源内部数字电路电源滤波输出管脚  
通常为 1.8V，可外接 10nF 或者 100nF 电容。

### 10.2.1 MCU 电源管理模式

电源低功耗管理模式有如下几种：不同模式关闭的模块不同，功耗有区别

- ◇ 休眠模式（IDLE）
- ◇ 掉电模式 1（STOP1）
- ◇ 掉电模式 2（STOP2\STOP2D\STOP2R）
- ◇ 关机模式（PFMOD）

	状态描述	进入方式	退出方式
IDLE	<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]写 1	<ul style="list-style-type: none"> <li>✓ 所有中断</li> </ul>
STOP1	<ul style="list-style-type: none"> <li>✓ 关闭CLK_MCU，整个MCU核停止运行</li> </ul>	PCON[1]写 1	<ul style="list-style-type: none"> <li>✓ 外部中断 0, 1</li> </ul>

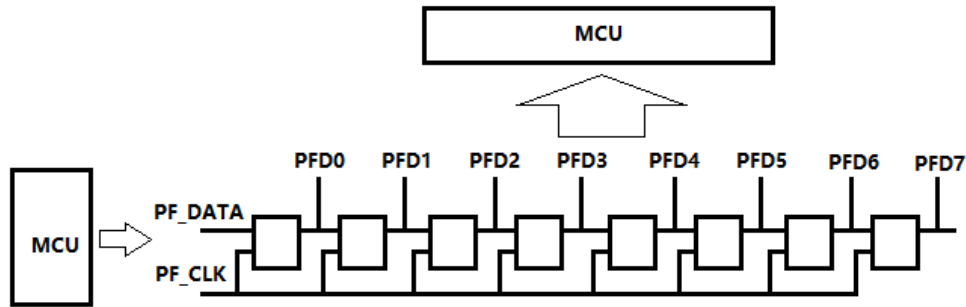
		<ul style="list-style-type: none"> <li>✓ 其他外设是否关闭，需要用户自己配置相关标志位！</li> </ul>		<ul style="list-style-type: none"> <li>✓ 键盘中断</li> <li>✓ 掉电中断</li> <li>✓ 串口 RX 低电平</li> <li>✓ 低频时钟唤醒中断</li> <li>✓ 24 位 ADC 中断</li> </ul>
S T O P 2	STOP2	<ul style="list-style-type: none"> <li>✓ 关闭CLK_OSC（MCU及全部外设停止运行）</li> <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> <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[4]写 1	<ul style="list-style-type: none"> <li>✓ 外部中断 0, 1</li> <li>✓ 键盘中断</li> <li>✓ 掉电中断</li> <li>✓ 串口 RX 低电平</li> <li>✓ 低频时钟唤醒中断</li> </ul>
	STOP2D	<ul style="list-style-type: none"> <li>✓ 关闭PDM2</li> </ul>	PCON3[3]=0 PCON[4]写 1	
	STOP2R	<ul style="list-style-type: none"> <li>✓ 关闭PDM1、PDM2</li> </ul>	PCON3[4]=0 PCON[4]写 1	<ul style="list-style-type: none"> <li>✓ 外部 I0 唤醒</li> <li>✓ 低频时钟唤醒</li> </ul>
PFMOD		<ul style="list-style-type: none"> <li>✓ 直接关闭DLDO（此时VDDR为0）</li> </ul>		<ul style="list-style-type: none"> <li>✓ 外部 I0 唤醒</li> </ul>
<p><b>注意事项：</b></p> <ul style="list-style-type: none"> <li>◇ 唤醒“IDLE”“STOP1”“STOP2、STOP2D、STOP2R”时，需要将相关中断的开启标志打开 唤醒和总中断标志 EA 是否打开无关，如果 EA 为 1 则唤醒并执行中断，EA 为 0 则只是唤醒</li> <li>◇ 唤醒“IDLE”“STOP1”“STOP2、STOP2D”时，需要将相关中断的开启标志打开 “STOP2R/关机模式”，由于 MCU 核逻辑电源关闭，唤醒和中断标志是否打开无关</li> <li>◇ “STOP1”注意： 如果进入 STOP1 和“24 位 ADC 中断”同时发生，可能导致“24 位 ADC 中断”无法唤醒“STOP1”。所以如果使用“STOP1”，尽可能开启“低频时钟唤醒中断”，这样即便在较低概率撞上时，仍然能唤醒 STOP1 在 CLK_MCU 频率不是很高时，如 2 分频，STOP1 的功耗和 IDLE 差不多，所以不一定要用 STOP1</li> </ul>				

### 10.2.2 “STOP2R/关机模式”说明

- “STOP2R/关机模式”：会关闭相关模块电源，所以：
  - ◇ （STOP2R）： XRAM 中的数据会丢失
  - ◇ （关机模式）： 全部 RAM 中的数据会丢失
  - ◇ 唤醒后，“用户程序”会复位执行
- 如何判断“用户程序”复位执行时是由于“上电复位”还是“STOP2R/关机模式”：
  - ◇ （STOP2R）： 可以通过 PDM0 中的寄存器来判断，进入 STOP2R 前，先把某个寄存器赋特定值： 建议使用 ICPENA、ICPENB、ICPENC 这三个用户寄存器，这三个寄存器没有控制相关设备。
  - ◇ （关机模式）： 可以通过关机模块中的数据来区分是否是“上电复位”。
- 如何确保“STOP2R/关机模式”时，IO 状态不发生变化
  - ◇ 进入“STOP2R/关机模式”前，通过对 PCON3[7]写入 1，将 IO 的输出以及模式锁定
  - ◇ 唤醒后，注意在“用户程序”中对全部 IO 以及 IO 模式配置正确后，将 PCON3[7]清 0 解锁，否则无法对 IO 进行操作。
- “STOP2R”的唤醒：
  - ◇ 可配置“低频时钟唤醒”
  - ◇ 外部 IO（P1.0、P2.2）低电平唤醒，具体参见寄存器：PDM0\_IOM(0xD5)

- 需要在“烧录配置”中允许低功耗，才能进入“STOP2R/关机模式”

### 10.2.3 “关机模块”



- PFD0 - PFD7 为移位寄存器：
  - ◇ PFD0 - PFD5 直接送到 MCU，能被读到；
  - ◇ PFD6 或 PFD7 逻辑后，合并为一个信号送到 MCU；
  - ◇ PFD0 - PFD7 “上电复位”为 1
- PFDATA(0xEE): 关机模式数据寄存器（默认值：0x7F）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	PFD7    PFD6	PFD5	PFD4	PFD3	PFD2	PFD1	PFD0

- PFD0 - PFD4 分别控制： P1.0、P1.3、P1.6、P2.0、P2.2 唤醒“关机模式”，对应位为 0，表示对应 IO 可以低电平唤醒“关机模式”。
- 进入“关机模式”的条件
  - ◇ D0 - D5 至少有一位为 0
  - ◇ PFD7 和 PFD6 均为 0
- 移位时序控制  
通过对寄存器 PFCON(0xED) 的操作，控制移位寄存器的时序，配置关机模块的唤醒信号

- PFCON(0xED): 关机模式控制寄存器（默认值：0x01）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
					PF_DATA	PF_CLK	PF_LOCK

- PF\_DATA: “关机模块”的数据信号
- PF\_CLK: “关机模块”的时钟信号
- PF\_LOCK: = 1: 锁定 PFCON，不能写入  
= 0: 解锁 PFCON

### 10.2.4 相关寄存器

系统三种工作模式，通过寄存器 PCON 来配置，在配置 PCON 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:	写 1 软件复位（弱复位），仅复位“用户程序”
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”

■ **PCON2(0xA7):** 电源控制寄存器2（默认值：0x01）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
		CPMP_LDO	PF_CON	PORH_CON[1:0]		CPMP_MD	PCON23_LK

CPMP_LDO:	= 0, 升压模块配置为 升压模式 = 1, 升压模块配置为 LDO模式
PF_CON:	= 1, 进入“关机模式”（前提是关机模块配置正常）
PORH_CON[1:0]:	“电压检测模块（PORH）”的检测电平控制： = 00: 1.7V; = 01: 2.4V; = 10: 3.2V; = 11: 4.0V;
CPMP_MD:	升压模块工作模式： = 0: 普通模式 = 1: 低功耗模式
PCON23_LK:	= 1, 锁定对寄存器：PCON2\PCON3 的写操作 <b>（注意：除 PCON23_LK，其它位操作需要解除锁定）</b>

■ **PCON3(0xBF):** 电源控制寄存器3（默认值：0xdc）

**（注意：PCON3操作需要通过PCON23\_LK解除锁定，同时需要开启WD\_TA时间窗口）**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
IOLOCK	PORH_EN	PORH_SPD	PDM1_SPD	PDM2_SPD	WDOSC_SPD[1:0]		DLDO_SPD

IOLOCK:	= 0, 锁定IO的输出以及配置模式
PORH_EN:	= 1, 开启“电压监测模块（PORH）” = 0, 关闭“电压监测模块（PORH）” <b>安全起见，不要直接关闭 PORH，最好设定为同步 STOP2 关闭</b>
PORH_SPD:	= 1, “电压检测模块（PORH）”在进入“STOP2”模式时同步关闭
PDM1_SPD:	= 0, “数字逻辑：PDM1”在进入“STOP2”模式时同步关闭
PDM2_SPD:	= 0, “数字逻辑：PDM2”在进入“STOP2”模式时同步关闭
WDOSC_SPD[1:0]:	= 01, “低频时钟振荡器”在进入“STOP2”模式时同步关闭
DLDO_SPD:	= 1, “DLDO”在进入“STOP2”模式时同步进入低功耗模式

■ **PDM0\_IOM(0xD5):** “STOP2R”的IO唤醒控制（默认值：0x3f）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
			PDM0MS4				PDM0MS0

PDM0MS4: = 0, 配置P2.2低电平唤醒 “STOP2R”  
PDM0MS0: = 0, 配置P1.0低电平唤醒 “STOP2R”

■ ICPENA(0xBC): 用户寄存器A (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				ICPENA-3	ICPENA-2	ICPENA-1	ICPENA-0

■ ICPENB(0xBD): 用户寄存器B (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				ICPENB-3	ICPENB-2	ICPENB-1	ICPENB-0

■ ICPENC(0xBE): 用户寄存器C (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
				ICPENC-3	ICPENC-2	ICPENC-1	ICPENC-0

### 10.3 “复位及电源监测”

■ 特征概述:

- ◇ “硬件复位”和“软件复位”
  - ❖ “硬件复位”: 由硬件产生的“电源复位信号”, SDI5229内部设置“电压检测模块(PORH)”以及“商店复位模块(PORL)”,两个模块均能产生“电源复位信号”。
  - ❖ “软件复位”: 可以由程序控制产生的复位信号
- ◇ “强复位”和“弱复位”
  - ❖ “强复位”: 整个MCU复位,并重新启动运行。“硬件复位”均是“强复位”
  - ❖ “弱复位”: 只复位运行“用户程序”

#### 10.3.1 “硬件复位”

- “上电复位模块(PORL)” (PORL):
  - ◇ 上电复位门限为 1.6v、掉电检测门限为 1.4v
  - ◇ 电流约 0.1uA,无法被关闭
- “电压监测模块(PORH)” (PORH):
  - ◇ 检测门限可配置(具体参见寄存器: **PCON2(0xA7):**)
  - ◇ 信号可用于复位或者掉电中断
  - ◇ 电流约 20uA,可以被关闭
  - ◇ **安全起见,如果要关闭 PORH,可能的话最好设定为同步 STOP2 关闭**
- “电压监测模块(PORH)”以及“上电复位模块(PORL)”只要任何一个进入有效复位电压,则将产生复位信号
  - ◇ 当复位信号保持超过门限 30uS 以上时,将整个 MCU 复位。
  - ◇ MCU 的掉电速度最好能保持在 5mV/us 以下

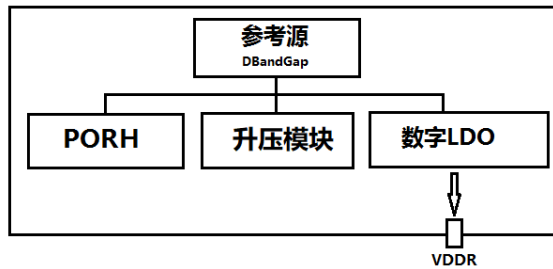
#### 10.3.1 “软件复位”

- “看门狗复位”: “弱复位”
- “软件复位 1”: “弱复位”: 控制位: **SOFT\_RST(PCON[6])** 写 1 启动



- “软件复位 2”：“强复位”：控制位：GF1、GF2(PCON[3:2])写 11 启动

### 10.2.5 “低功耗说明”



“STOP2”时，绝大部分模块都进入休眠功能，但默认状态下仍然有几个电流较大的模块处于工作状态：

- ◇ “电压监测模块（PORH）”
  - ◇ “数字 LDO”（为内部数字逻辑提供电压）
  - ◇ “参考源：DBandGap”（为“PORH”\“数字 LDO”\“升压模块”提供参考电流）  
当关闭：“PORH”\“数字 LDO”\“升压模块”时，参考源将自动关闭
- 总共一起约 150uA 左右

需要进一步降低待机电流，需要按照如下表格操作：

依次进行如下关闭动作	待机电流	说明
STOP2：关闭 OSC	150uA	
关闭 PORH	130uA	PORH 电流约 20uA
DLDO 跟随 STOP2 进入低功耗模式（DBandGap 自动关闭）	22uA	
STOP2D：关闭 PDM2	4uA	建议大部分情况下，执行 STOP2D
STOP2R：关闭 PDM1	1.5uA	
关闭看门狗	0.8uA	
关机模式	0.2uA	

## 11-看门狗与低频唤醒

（相关中断，请参考第 7 章“7-中断”）

- “看门狗模块”特征概述：
  - ◇ “看门狗定时”中断或者复位：
    - ❖ 可设定：0.25 秒、0.5 秒、1 秒、4 秒的时长
    - ❖ 可设定中断或者复位
    - ❖ “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
  - ◇ “低频唤醒”中断：
    - ❖ 可设定 0.004 秒 - 1 秒的时长，唤醒除“关机模式”外的各低功耗模式，如：STOP2R
    - ❖ “低频唤醒”启动后，如果没有处于“STOP1/2”，其计数器清零。当进入“STOP1/2”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“STOP1/2”

- ◇ “低频计时”中断：
  - ❖ 可产生“1秒”的标准中断，用于计时（用“看门狗定时”中断也可计时，但通常其会用于复位）
  - ❖ “看门狗中断”和“低频计时中断”，他们共用1个中断向量资源，具有共同的有优先级，采用不同的标志位区分，并有不同的使能位。
- ◇ “看门狗模块”全部定时采用 32KHz 的时钟，当使用内置低频振荡器时，会有一些的偏差，程序设计时注意：
  - ❖ 如果对精确度要求比较高，可以采用程序校准
- ◇ WDCON 的“访问窗口”
  - ❖ “看门狗控制寄存器：WDCON (0xA9)”，除了“Bit0”外，对其他位的访问需要先开启“访问窗口”。
- ◇ 请注意：看门狗引起的复位并不复位“寄存器 WDCON”，也就是看门狗如果开启的话，看门狗复位后仍然是开启的。

## 11.1 相关寄存器

### ■ 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
WUEN	WUT[1:0]		WDEN	WDT[1:0]		WDRST	WDCLR

### ■ WDCON2 (0xD7): 看门狗控制寄存器2（默认值：0x01）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
							WUT[2]

WUEN: 低频唤醒  
1: 开启  
0: 关闭

WUT[2:0]: 低频唤醒时间选择  
000: 0.004s  
001: 0.012s  
010: 0.02s  
011: 0.06s  
100: 0.15s  
101: 0.20s  
110: 0.4s  
111: 1.0s

WDEN: 看门狗  
1: 开启  
0: 关闭

WDT[1:0]: 看门狗时间选择  
00: 0.25s  
01: 0.5s  
10: 1s  
11: 4s

WDRST: 看门狗复位标志（此位之能写，读取的话输出一直为0）

WDCLR: 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  
看门狗计数器清 0（不需要开启访问窗口）  
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

## 11.2 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

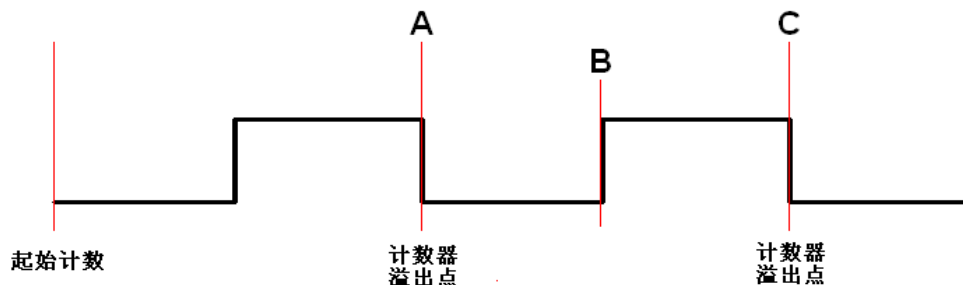
对 WD\_TA 依次连续写入：0x05，0x0A，将开启“一条指令的时间窗口”访问寄存器“WDCON”  
如：

```
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;
```

## 11.3 “掉电模式”下的看门狗运行

“掉电模式：STOP1/2”下：

- ◇ “看门狗复位”：“看门狗定时”溢出后，依然能够复位 MCU；
- ◇ “看门狗中断”和“低频计时中断”：因为此时 CLK\_MCU 停止，所以中断标志不会被响应。
- ◇ “掉电模式”下的看门狗运行：
  - “掉电模式：STOP1/2”，“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”和“低频计时中断”。
  - ✓ 计数器溢出后，计数器重新从 0 开始继续计数。
  - ✓ 同时，中断标志将会保留到，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。

- ✓ 所以，通常情况下，将“低频唤醒”时长设置小于“看门狗中断”和“低频计时中断”的一半，将不回丢失中断标志

# 12-脉宽调制模块(PWM)

## 12.1 相关寄存器

- **PWMF\_H (0x9A):** PWM频率控制高位寄存器 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT3	BIT2	BIT1	BIT0	BIT15	BIT14	BIT13	BIT12

- **PWMF\_L (0x9B):** PWM 频率控制低位寄存器 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4

- **PWM0\_H (0xAD):** PWM通道0的门限-高位 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT3	BIT2	BIT1	BIT0	BIT15	BIT14	BIT13	BIT12

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

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4

- **PWM1\_H (0xAE):** PWM通道1的门限-高位 (默认值: 0x00)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT3	BIT2	BIT1	BIT0	BIT15	BIT14	BIT13	BIT12

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

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4

- **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
			CLKMCU_O	PWM_CY_N	PWM_EN	PWM1_INV	PWM0_INV

CLKMCU\_O: =1: MCU 时钟 CLK\_MCU 从 PWM1 对应管脚输出

PWM\_CY\_N: PWM 周期辅助控制:

可以先往该位写 1, PWM 溢出时将自动清零

PWM\_EN: PWM 模块

1: 开启

0: 关闭

PWM1\_INV: PWM1 的输出配置位

1: (PWM\_EN=1) 当 PWMF <= PWM1 时, pwm1\_out 输出 0

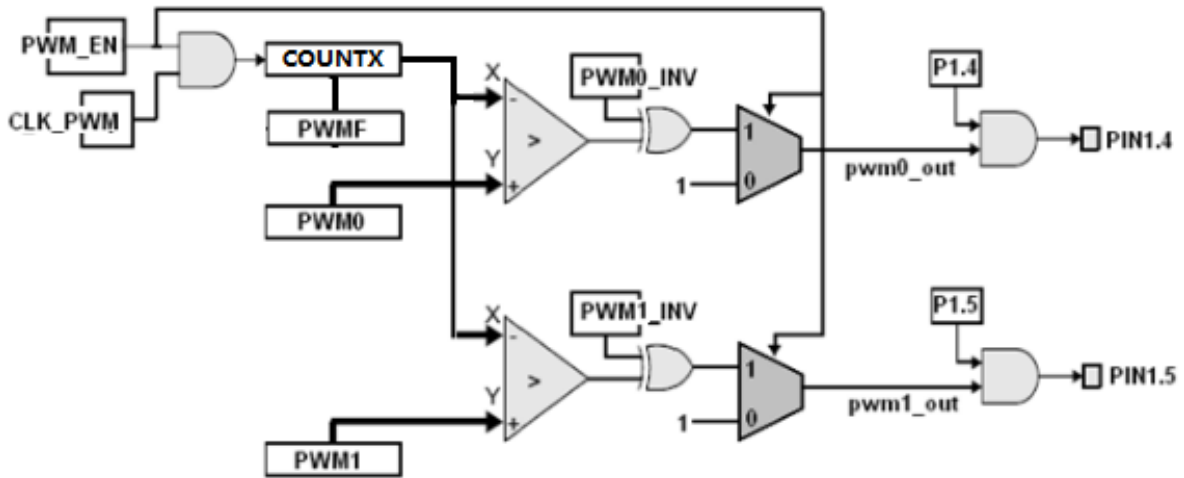
0: (PWM\_EN=1) 当 PWMF <= PWM1 时, pwm1\_out 输出 1

PWM0\_INV: PWM0 的输出配置位

1: (PWM\_EN=1) 当 PWMF <= PWM0 时, pwm0\_out 输出 0

0: (PWM\_EN=1) 当 PWMF <= PWM0 时, pwm0\_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 PWMF 时从新回到 0 计数;
  - ✧ PWMF 频率控制寄存器为 16 位,放在寄存器“PWMF\_H”“PWMF\_L”中。
  - ✧ PWM0 和 PWM1 分别为 16 位门限计数器。
  - ✧ PWM 关闭 (PWM\_EN = 0): pwm1\_out、pwm0\_out 输出 1。  
PWM 开启 (PWM\_EN = 1):
    - ✓ (PWM0/1\_INV = 0):
      - 当 COUNTX <= PWM0 时, pwm1/0\_out 输出 1
      - 当 COUNTX > PWM0 时, pwm1/0\_out 输出 0
    - ✓ (PWM0/1\_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 PWM1\_INV 、PWM0\_INV 控制输出反相)
- ✧ PWM 模块的时钟频率为 CLK\_PWM
    - ✓ CLK\_PWM 频率和 CLK\_MCU 相同,从 CLK\_OSC (9.83MHz) 分频得到
    - ✓ PWMF 最大为 16 位,如果不考虑 CLK\_OSC 分频最低可实现的频率为(9.83MHz / 2^16)

# 13-(SIGMA-DELTA) 24 位 ADC

(相关中断, 请参考第 7 章“7-中断”)

■ 概述:

◇ 低功耗, 高精度:

- ❖ 工作电压: 1.55V - 3.3V, 方便各种低功耗应用
- ❖ 注意: 如果VCOM输出AVDD, 则AVDD不能高于3.3V(可在烧录器下载时配置禁止VCOM输出AVDD, 只允许采用LDO模式输出)

◇ 数据快速稳定:

- ❖ 最快可配置为“工作后第1个数据就稳定”, 方便通道切换以及休眠恢复

◇ 提供各种放大倍数:

◇ 提供各种输出速率:

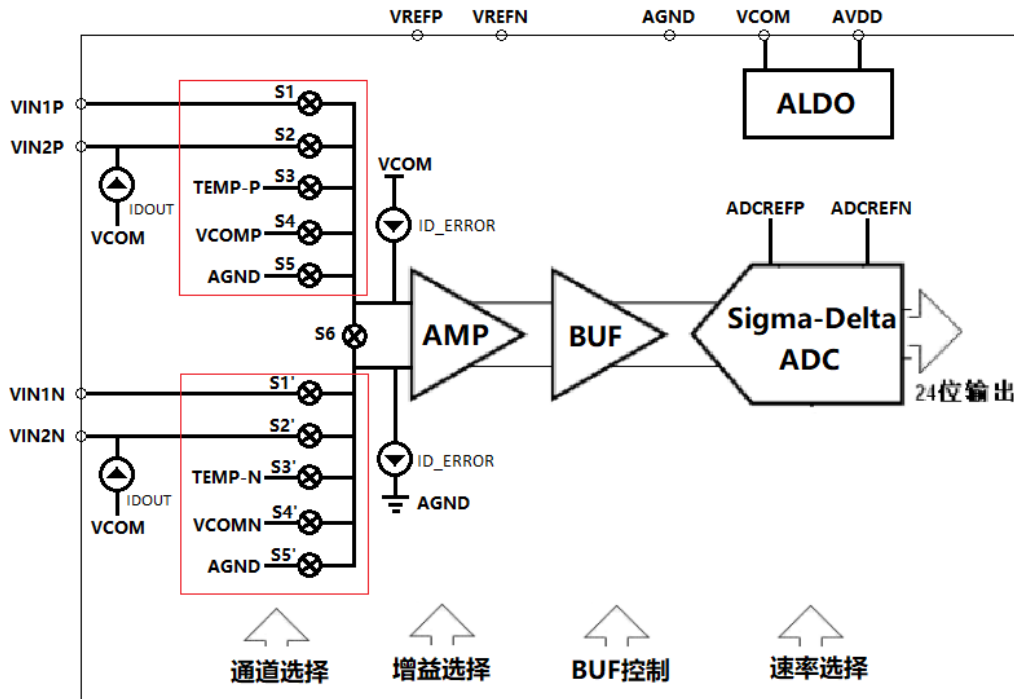
◇ 基准电压: AVDD 经过 ALDO 输出电压基准: VCOM

- ❖ 输出可通过配置位“ALDO\_SEL[2:0]”调节: 1.55v、2.0v、2.4v、3.0v、AVDD
- ❖ 方便驱动电压形式的传感器
- ❖ 输出端 VCOM 建议外接 10nF-0.1uF 电容 (不接电容 ALDO 也是稳定的, 如在 VCOM 没封装出来的时候)

◇ 基准电流: ALDO 供电给电流基准, 并从 VIN2P、VIN2N 输出恒定电流:

- ❖ 输出可通过配置位“IDOUT\_CON[1:0]”调节
- ❖ 50uA - 1mA
- ❖ 方便驱动电流形式传感器

## 13.1 配置说明



■ 整体说明

- ◇ AVDD 给 ALDO 供电, 输出基准电压 VCOM
- ◇ VCOM 为整个 24 位 ADC 提供电源, AGND 是模拟地

- ◇ ADCREFP 和 ADCREFN 为 ADC 的正负参考电压
  - ❖ 通常情况下，ADCREFP 和 ADCREFN 分别为：VREFP 和 VREFN
  - ❖ “VCOM 和 VREFP”、“VREFN 和 AGND”之间内部有开关，可控制 VREFP 和 VREFN 分别连接到 VCOM 和 AGND
  - ❖ 选择温度通道时：ADCREFP 和 ADCREFN 直接为 VCOM 和 AGND

## ■ 信号通路

微弱信号路径上依次有：“通道选择”、“放大器 AMP”、“BUF”、“24 位 ADC”

- ◇ 前置放大器 AMP：可通过控制位“SGAIN\_AMP(SGADCON[3])”选择是否开启前置放大器 AMP 放大 64 倍，或信号直通（注意：此时没有增强输入电阻的 BUF 作用）。

- ◇ BUF:

增大 ADC 的输入电阻，隔离后级 ADC 对前级的影响，可以配置开启或者关闭

**BUF 采用“Rail-to-Rail”结构，其信号的输入范围为：AGND - VCOM**

参见控制位：“BUF\_EN(SGADCON4X[5])”“BUF\_LCAP(SGADCON4X[4])”

- ◇ ADC 输出值（参考电压输入：VREF = ADCREFP - ADCREFN）

输入信号如：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

## ■ 输入通道选择

- ◇ 24 位 ADC 提供三个通道数据测量通道，通过“SG\_CHA(SGADCON[5:4])”来控制。其中，通道 1 和通道 2 是普通的可外接的差分输入，通道三在连接内部的温度传感器。

SG_CHA[1:0]	
00	通道 1: VIN1P、VIN1N (S1\S1'闭合, 其他断开)
01	通道 2: VIN2P、VIN2N (S2\S2'闭合, 其他断开)
10/11	温度通道 (S3\S3'闭合, 其他断开)

- ◇ 注意：可通过 VINN\_REV (SGADCON3X[6]) 将通道 1 和通道 2 的负输入端反过来

VINN\_REV=0: 两路差分通道为: (VIN1P \ VIN1N)、(VIN2P \ VIN2N)

VINN\_REV=1: 两路差分通道为: (VIN1P \ VIN2N)、(VIN2P \ VIN1N)

SG_CHA[1:0]	
00	通道 1: VIN1P、VIN2N (S1\S2'闭合, 其他断开)
01	通道 2: VIN2P、VIN1N (S2\S1'闭合, 其他断开)

- ◇ 可以控制信号差分输入的一端接地:

- ❖ 参见控制位：“VINP\_GND(SGADCON2[5])”:

(S5 闭合, 负端看选择通道及是否反相, 其他断开)

- ❖ 参见控制位：“VINN\_GND(SGADCON2[4])”

(S5'闭合, 正端看选择通道, 其他断开)

- ◇ 可以控制差分输入信号为 0:
  - ❖ 参见控制位: “VIN\_ZERO(SGADCON2[2])”  
(S6 闭合, 负端无影响连接看选择通道及是否反相, 其他断开, 注意正端开关全部断开)
- ◇ 可以在差分输入端增加共模偏置, 这样不需要外部偏置:
  - ❖ 参见控制位: “VIN\_COM(SGADCON2[3])”  
(S4、S4' 闭合)  
(注意并不影响正负信号输入端, 其如何选择要看选择通道及是否反相, 其他断开)
  - ❖ VIN\_COM 配合 VINP\_GND、VINN\_GND 可以实现输入一端接内部共模, 一端外部输入 VIN\_COM 开启后, 如果:
    - VINP\_GND = 1: 表示 VINP 和外部断开, 只是内部接共模
    - VINN\_GND = 1: 表示 VINN 和外部断开, 只是内部接共模
- ◇ 可以在差分输入端增加上下拉小电流:
  - ❖ 参见控制位: “ID\_ERROR(SGADCON4X[1])”
- ◇ 可以配置 VIN2P 和 VIN2N 输出基准电流:
  - ❖ 参见控制位: “IDOUT\_CON(SGADCON4X[7:6])”

#### ■ 增益控制

- ◇ 内部可编程增益: 1、2、4、16、32、128、256、512、1024、2048、... 等增益  
ADC 信号通路上有四类增益放大器
  - ❖ GAMP: 前置运算放大器 (控制位: SGAIN\_AMP(SGADCON[3]))  
1、64 倍;
  - ❖ GCLK / ACLK: 时钟积分增益放大  
GCLK: 1、2、4、8(默认)、16 (控制位: GCLK\_CON(SGADCON3X [5:3]))  
ACLK: 1、2、4(默认)、8 (控制位: ACLK\_CON(SGADCON3X [2:1]))  
注意:
    - ✓ 正常配置时, 必须满足: GCLK > ACLK; 信号通路至少有 2 倍放大。
    - ✓ 如果需要: GCLK = ACLK, 则必须将 FCCH(SGADCON3X [0]) = 0
    - ✓ GLCK 不能比 ACLK 小
  - ❖ GCAP: 电容放大: (控制位: SGG\_CAP(SGADCON2 [2]))  
1、4 倍;
  - ❖ GREF: REF 放大 (控制位: SGG\_REF(SGADCON4X [3]))  
1、4 倍;

$$\text{ADC 的增益} = \text{GAMP} * (\text{GCLK} / \text{ACLK}) * \text{GCAP} * \text{GREF}$$

- ❖ 通常精度要求高的情况下, 极可能使用需要使用 “GAMP”、“GCLK / ACLK” 来配置放大倍数
- ❖ 但是 “GAMP” 开启前置放大器会增大 ADC 功耗 (大约 0.8mA), 所以, 在精度要求低的时候, 可以不要 “GAMP”, 通过 “GCAP”、“GREF”、“GCLK / ACLK” 来配置放大倍数;



◇ 如（未列出的配置情况请按照增益公式计算）：**注意 GCLK=ACLK 时必须 FCCH=0**

GAMP	GCLK	ACLK	GCAP	GREF	ADC 增益
1	4	4	1	1	1
1	8	4	1	1	2
1	16	4	1	1	4
1	16	2	1	1	8
1	16	1	1	1	16
1	8	1	4	1	32
<b>64</b>	<b>4</b>	<b>4</b>	<b>1</b>	<b>1</b>	<b>64</b>
<b>64</b>	<b>8</b>	<b>4</b>	<b>1</b>	<b>1</b>	<b>128(默认)</b>
64	16	4	1	1	256
<b>64</b>	<b>16</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>512</b>
64	16	1	1	1	1024

### ■ ADC 控制

◇ ADC 采样速率调整

ADC 对输入模拟差分信号的采样基础频率  $F_s$  (76.8KHz)，可通过配置调整为：

- ❖ 采样频率： $ACLK * F_s$
- ❖  $ACLK = 1、2、4(默认)、8$

◇ ADC 输出速率选择

❖ ADC 数据输出速率为：“配置速率” \*  $ACLK$

❖ “配置速率”通过 SG\_SP (SGADCON[2:0]) 控制：

SG_SP[2:0]:	24 位 ADC: 输出速率选择
000:	2.5Hz * (ACLK)
001:	5Hz * (ACLK)
010:	10Hz * (ACLK)
011:	20Hz * (ACLK)
100:	40Hz * (ACLK)
101:	80Hz * (ACLK)
110:	160Hz * (ACLK)
111:	320Hz * (ACLK)

❖  $ACLK$  为 ADC 速度倍率，通过  $ACLK\_CON(SGADCON3X [2:1])$  控制：

$ACLK\_CON[2:0]$ :	控制 $ACLK$
00:	$ACLK = 1$
01:	$ACLK = 2$
10:	$ACLK = 4$
11:	$ACLK = 8$

◇ ADC 滤波控制

❖ 滤波模式控制： $FILB\_MODE (SGADCON5X[5:4])$

滤波强度越大，信号精度越高，恢复速度变慢； 通常情况，如果不需要极高的精度，采用“滤波模式 1”就可以了，此时不需要丢弃数据，第 1 个数据就是稳定的：

$FILB\_MODE[1:0]$ :	数字滤波器模式：
<b>00:</b>	<b>滤波模式 3 (输出速率为：<math>640Hz * ACLK</math>)</b>

- 01: 滤波模式 1 (工作后, 第 1 个数就稳定)
  - 10: 滤波模式 2 (工作后, 第 2 个数就稳定)
  - 11: 滤波模式 3 (工作后, 第 3 个数就稳定) (默认)
- ❖ 控制 ADC 工作后, 自动丢弃的数据个数, 丢弃完后新数据才产生中断标志: FILB\_MODE (SGADCON5X[5:4]):
- RESTORE\_NUM[1:0]:
- 00: 不丢弃
  - 01: 丢弃 1 个数
  - 10: 丢弃 2 个数
  - 11: 丢弃 3 个数

## 13.2 相关寄存器

- 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

- SGADCON(0xB1): 休眠辅助寄存器 (默认值: 0x09)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
SG_EN	SG_IDLE_N	SG_CHA[1:0]		SGAIN_AMP	SG_SP[2:0]		

- 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\_CHA[1:0]: 24 位 ADC: 通道选择  
00: 通道 1 (VIN1P/VIN1N)  
01: 通道 2 (VIN2P/VIN2N)  
10/11: 温度通道
- SGAIN\_AMP: 24 位 ADC: 是否开启 64 倍前置运算放大器(GAMP)  
1: 开启 (GAMP = 64)  
0: 不开启 (GAMP = 1)  
(温度通道请选择 0: 不开启, 否则信号将溢出)

SG\_SP[2:0]: 24 位 ADC: 输出速率选择

000:	2.5Hz	*	(ACLK)
001:	5Hz	*	(ACLK)
010:	10Hz	*	(ACLK)
011:	20Hz	*	(ACLK)
100:	40Hz	*	(ACLK)
101:	80Hz	*	(ACLK)
110:	160Hz	*	(ACLK)
111:	320Hz	*	(ACLK)

(ACLK 默认为 4, 参见寄存器“SGADCON3X”)

■ **SGADCON2 (0xA1):** 24位ADC控制寄存器2 (默认值: 0x41)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ALDO_SEL[1:0]		VINP_GND	VINN_GND	VIN_COM	VIN_ZERO	SGG_CAP	REF_SW

ALDO\_SEL[2:0]: ALDO 输出选择寄存器

000:	AVDD	
001:	输出 1.5v	(默认配置)
010:	输出 2.0v	
011:	输出 2.4v	
1XX:	输出 3.0v	

ALDO\_SEL2 参见寄存器“SGADCON4”

VINP\_GND: =1: 24位ADC差分输入端: 正端和外部断开, 内部接地

VINN\_GND: =1: 24位ADC差分输入端: 负端和外部断开, 内部接地

VIN\_COM: “24位ADC差分输入端”共模偏置

=1: 芯片内部给正负端一个  $(VREFP - VREFN)/2$  偏置, 输出负载为25K欧  
V

VIN\_ZERO: 24位ADC差分输入为0IN\_COM=1时: VINP\_GND\VINN\_GND=1表示VINP\VINN  
不和外部接通

=1: 24位ADC差分输入端: 正端和外部断开, 内部接到负端

SGG\_CAP: 信号放大: GCAP 放大

1: GCAP = 4, 放大4倍

0: GCAP = 1, 不放大

REF\_SW: 控制: VCOM和VREFP、VREFN和AGND 之间的开关

1: 开关闭合

0: 开关断开

■ **SGADCON3X (0xD3):** 24位ADC控制寄存器3 (默认值: 0x9d)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
CCEN	VINN_REV	GCLK_CON[2:0]			ACLK_CON[1:0]		FCCH

CCEN: (系统配置) 内部测试信号

请配置: 1 (默认值)

VINN\_REV: 将通道 1 和通道 2 的负输入端反过来

=0: 两路差分通道为: (VIN1P\VIN1N)、(VIN2P\VIN2N)

=1: 两路差分通道为: (VIN1P\VIN2N)、(VIN2P\VIN1N)

GCLK\_CON[2:0]: 控制GCLK  
 000: GCLK = 1  
 001: GCLK = 2  
 010: GCLK = 4  
 011: GCLK = 8  
 1xx: GCLK = 16

ACLK\_CON[2:0]: 控制ACLK  
 00: ACLK = 1  
 01: ACLK = 2  
 10: ACLK = 4  
 11: ACLK = 8

FCCH: (系统配置) 内部测试信号  
 请配置: 1 (默认值)

■ **SGADCON4X(0xD4):** 24位ADC控制寄存器4 (默认值: 0x30)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
IDOUT_CON[1:0]	BUF_EN	BUF_LCAP	SGG_REF	TEMP_REF_	ID_ERROR	ALDO_SEL[	

IDOUT\_CON[1:0]: VIN2P、VIN2N 电流基准输出控制  
 00: 不输出  
 01: 50uA  
 10: 200uA  
 11: 500uA

BUF\_EN: VIN2P和VIN2N合并起来, 可以输出最大1mA的电流  
 输入通道 BUF 控制 (BUF 的输入输出为 Rail-to-Rail)  
 1: 开启内部BUF  
 0: 关闭内部BUF

BUF\_LCAP: BUF\_LCAP 属性 (该属性会影响 BUF 的 Rail-to-Rail 属性, 如果 BUF 输入比较靠近 VCOM 和 AGND, 建议关闭)  
 1: 开启  
 0: 关闭

SGG\_REF: 信号放大: GREF 放大模式  
 1: GREF = 4, 放大4倍  
 0: GREF = 1, 不放大

ID\_ERROR: 24 位 ADC 差分输入端上下拉电流控制:  
 正端加 4uA 上拉电流, 负端加 4uA 下拉电流  
 1: 开启上下拉电流  
 0: 关闭上下拉电流

■ **SGADCON5X (0xD2):** 24位ADC控制寄存器5 (默认值: 0x31)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
	RMCH	FILB_MODE[1:0]	RESTORE_NUM[1:0]	REV1	REVO		

RMCH: (系统配置) 内部测试信号  
 请配置: 0 (默认值)

- FILB\_MODE[1:0]: 数字滤波器模式:  
 00: 滤波模式 3 (输出速率为: 640Hz \* ACLK)  
 01: 滤波模式 1 (工作后, 第 1 个数就稳定)  
 10: 滤波模式 2 (工作后, 第 2 个数就稳定)  
 11: 滤波模式 3 (工作后, 第 3 个数就稳定)
- RESTORE\_NUM[1:0]:控制ADC工作后, 自动丢弃的数据个数, 丢弃完后新数据才产生中断标志  
 00: 不丢弃  
 01: 丢弃 1 个数  
 10: 丢弃 2 个数  
 11: 丢弃 3 个数
- REV1: (系统配置) 内部测试信号  
 请配置: 0 (默认值)
- REV0: (系统配置) 内部测试信号  
 请配置: 1 (默认值)

■ **PD\_CON(0xA2):** 休眠辅助寄存器 (默认值: 0x02)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
XIO_LCK	P4_LCK1	P4_LCK0	SG_LCK1	SG_LCK0	ALDO_EN	ALDO_SYN	TEMP_EN

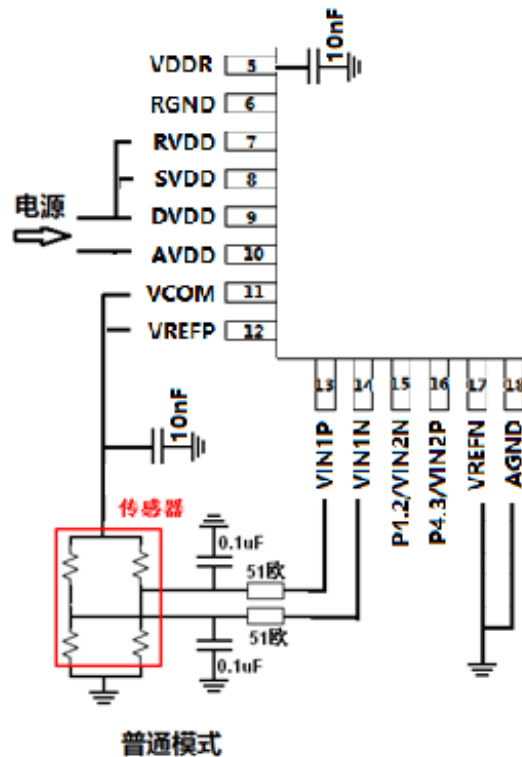
- XIO\_LCK: 外置振荡复用 IO 锁定  
 1: 锁定 P1.1、p1.2 为仅输入模式  
 0: 解锁(IO 模式由对应的模式控制位控制)
- P4\_LCK1 \P4\_LCK0: P4.2\P4.3 口 IO 锁定  
 11、10、01: 锁定 P4.2、P4.3 为仅输入模式  
 00: 解锁(IO 模式由对应的模式控制位控制)
- SG\_LCK1\SG\_LCK0: 24 位 ADC 核心寄存器锁定  
 11、10、01: 锁定寄存器: sgadcon, sgadcon2, sgadcon3x, sgadcon4x, sgadcon5x 忽略写入功能  
 00: 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位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

### 13.3 噪声性能:

配置说明	等效输入噪声有效值	有效位	不动码位数
<b>测试条件: (VCOM = 3.0V) (VREF = 3.0V)</b>			
放大: 2 倍 (GAMP=1; GCLK / ACLK = 8 / 4; ) 输出速率: 10Hz 其他配置: 默认	734nV	21.1	18.6
放大: 128 倍 (GAMP=64; GCLK / ACLK = 8 / 4; ) 输出速率: 10Hz 其他配置: 默认	23nV	20.1	17.8
<b>关于 ADC 精度配置说明</b>			
1: 提高 ACLK, 有利于提高 ADC 的精度 -- 提高 ACLK, 在其他配置不变的情况下, ADC 的输出速率提高 1 倍, 但精度下降非常少。 -- 提高 ACLK, 配置 ADC 输出速率不变的情况下, 精度明显提高;			

### 13.4 24 位 ADC 使用图例

- 下图是 SDI5229 用在重量传感器上的一组接线方式, 以供参考!



# 14-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；
- ◇ 不同电源电压下，温度通道的信号即便上是相同的，转换出来的数据只和 ADC 的参考电压大小成反比关系，程序处理时注意转换。
- ◇ 不同芯片，“温度传感器的信号大小”对比“芯片温度”的斜率是基本相同的，只是可能会存在整体偏差，导致温度测量在+/-3 度左右。  
如果有条件，可以采用单点校准的方法即可消除该偏差。

## 14.1 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):	3.315V
0 度的输出	2322335
温度每增加 1 度的输出变大	(10 度以上) 807
	(10 度以下) 749
	可以不区分大致设定统一线斜率: 780

## 14.2 相关寄存器

参考“13-(SIGMA-DELTA) 24 位 ADC”

## 14.3 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  
从通道 1 切换到温度通道，检测完温度后，重新切换回通道 1

```

//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];
};
//--参考电压 3.315v 的数据-- (780/1 度)
#define TEMP_VREF_B 3315 // (3.315V 参考)参考电压
#define TMEP_ZERO_CODE_B 2322335 // (3.315V 参考)温度 0 时的输出码
    
```

```

#define TEMP_PER_D_CODE_B1 749          //(3.315V 参考) 10 度以下: 变化 1 度时的输出码
#define TEMP_PER_D_CODE_B2 807          //(3.315V 参考) 10 度以上: 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_B (TEMP_ZERO_CODE_B - ZERO_SIGNAL_CODE)
#define TEMP_VREF_REAL    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,per_temp_code_acture2;
    //-----
    EA = 0; //禁止中断
    //-- 切换到温度通道 -- 160Hz --
    PD_CON = 0x07;          //开启温度通道/LDO
    SGADCON = 0xf4;        //切换到 温度通道,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 = (unsigned long)(TEMP_CODE_DELTA_B) * (unsigned long)(TEMP_VREF_B) / (unsigned long)(TEMP_VREF_REAL);
```

```
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 = (unsigned long)(TEMP_PER_D_CODE_B1) * (unsigned long)(TEMP_VREF_B) / (unsigned long)(TEMP_VREF_REAL);
```

```
per_temp_code_acture2 = (unsigned long)(TEMP_PER_D_CODE_B2) * (unsigned long)(TEMP_VREF_B) / (unsigned long)(TEMP_VREF_REAL);
```

```
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(ADcode_pre > (per_temp_code_acture * 10))//10 度以上
```

```
{
    ADcode_pre = ADcode_pre - (per_temp_code_acture * 10);
    Temperature = ADcode_pre / per_temp_code_acture2;
}
```

```
    Temperature = Temperature + 10;
}
else
{
    Temperature = ADcode_pre / per_temp_code_acture;
}

if(temp_sign)
    Temperature = -Temperature;
//--通道恢复：重新切换到通道 1 --
SGADCON = 0xc9;    //1 通道,20Hz
PD_CON = 0x7e;    //定义 "PD_CON" 的默认值 (锁定 P2 锁定 sgadcon, sgadcon2)
EXIF &= 0xbf;    //清除标志
//-- 丢掉通道切换造成的不稳定的数据 --
i = 0;
while (i<3)
{
    while((EXIF & 0x40) == 0x00) //如果没有数据
        ;
    EXIF &= 0xbf;    //清除标志
    i++;
}
}
```

# 15-Flash 操作说明

## 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。
- ◇ 用户Flash空间在烧录时可分配为：“用户程序空间”、“AIP数据空间”以及“用户ISP空间”，每个空间的大小可以调节：
  - ❖ AIP 数据空间： 非易失性数据存储空间，“用户程序”可修改此空间数据
  - ❖ 用户程序空间： “用户程序”存储空间
  - ❖ 用户 ISP 空间： 用于存放“用户引导程序”，开机后先运行用户引导程序，执行完成后，在跳转到用户程序执行。在执行“用户引导程序”时，可以对“用户程序空间”、“AIP 数据空间”进行修改。

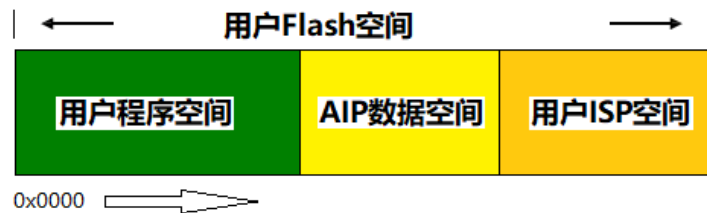
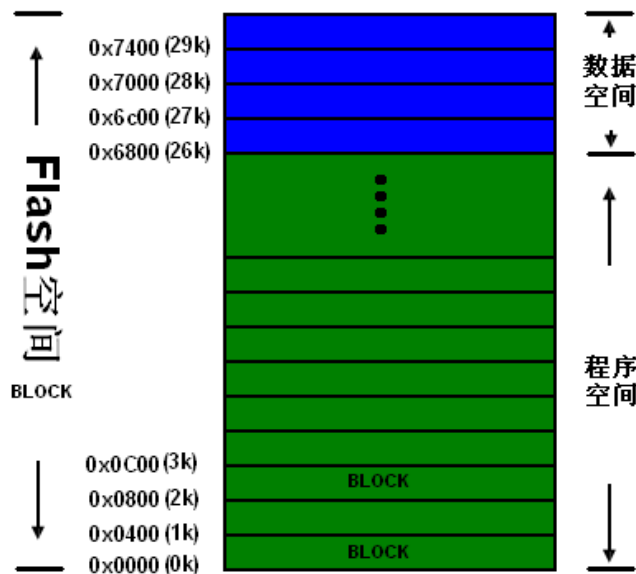


图 4-2：Flash 空间

- ❖ 从“用户引导程序”跳转到“用户程序空间”执行程序，需要如下操作：
  - 将寄存器“PSTATE(0xB9)”写 0xFF： PSTATE= 0xFF;
  - 执行软件复位： PCON |= 0x40;
- ◇ 请注意：“16.6 Flash 的抗干扰程序样例”
- ◇ 如下以 Flash-30K，其中数据区 4K 为例绘制出示意图：



## 15.1 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 MOV<sub>C</sub> 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);
}
```

## 15.2 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	<b>MAS0</b>	ERASE1	ERASE0	FWR1	FWRO

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

### 15.3 “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。

### 15.4 “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;
}

```

## 15.5 “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;
}

```

## 15.6 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
- ✧ “**NRM\_securty\_a**”、“**NRM\_securty\_b**”放在主程序进入操作 Flash 的不同程序分支中，不要放在只是顺序执行就能到操作 Flash 的程序段

```

//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;  
}
```

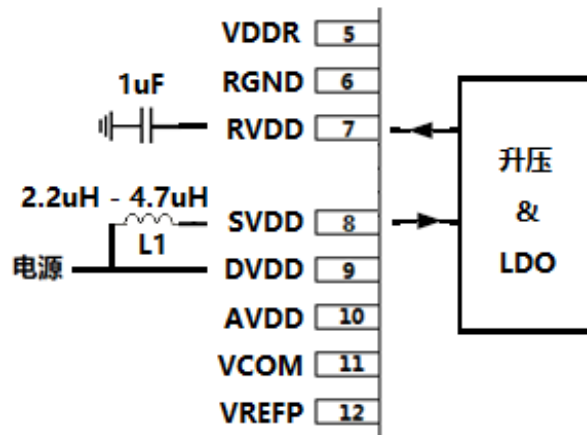


## 16-升压模块于液晶驱动

### ■ 概述

- ◇ 内置升压模块：
  - ❖ 正常升压模式：2.5v、2.8v、3.3v、5.0v（可配置输出电压）
  - ❖ 低功耗升压模式：3.3V（固定输出电压）
  - ❖ 升压模块可配置为LDO模式：输出电压配置和“正常升压模式”时相同
  - ❖ 内部开关直连模式：
  
- ◇ 液晶驱动：
  - ❖ 可配置80点（4\*20）的液晶驱动
  - ❖ 可配置128点（8\*16）的液晶驱动
  - ❖ “电阻驱动模式”和“低功耗电容模式”液晶驱动
  - ❖ 液晶驱动时，注意将相关IO配置为液晶模式（参见IO部分）

### 16.1 升压模块



- SDI5229 内部集成“升压模块”，可以将 SVDD 输送来的低电压升压，并稳定输出到 RVDD。
  - ◇ RVDD 给除“P4.2\P4.3”外的其他 IO 供电，其中包含全部液晶驱动 IO，RVDD 为液晶提供驱动基准电压。
  - ◇ 升压模块有四种工作模式：“正常升压模式”、“低功耗升压模式”、“LDO 模式”、“内部开关直连模式”。
  
- “正常升压模式”
  - ◇ 外部电路为：从 DVDD 串 2.2uH 或 4.7uH 电感到 SVDD；RVDD 接滤波电容（建议 >1uF）。
  - ◇ 通过 CPMP\_VOL(LCDSEL[5:4]) 可配置输出电压：2.5v、2.8v、3.3v、5.0v

◇ 此时升压时钟采用：CLK\_OSC的分频时钟，所以不能进入STOP2低功耗模式

■ “低功耗升压模式”：“控制位：CPMP\_MD (PCON2[1])”

◇ 外部电路为：和“正常升压模式”相同。

◇ 不能配置输出电压，固定为3.3V

◇ 此时升压时钟采用：低频时钟，升压时，MCU可以进入STOP2、STOP2D低功耗模式；但不能进入STOPR及关机模式。

■ “LDO模式”：“控制位：CPMP\_LDO (PCON2[5])”

◇ 外部电路为：即便处于串接电感的电路，也可以直接切换到LDO模式。

◇ 通过 CPMP\_VOL(LCDSEL[5:4])可配置输出电压：和“正常升压模式”相同

◇ LDO模式不需要时钟，MCU可进入STOP2、STOP2D低功耗模式；但不能进入STOPR及关机模式。

■ “内部开关直连模式”：

◇ 内部开关直接将 SVDD 和 RVDD 连接在一起

◇ 不采用“升压”、“LDO”模式，或者休眠关闭升压模块时，均处于内部开关直连模式

## 16.2 液晶驱动模块

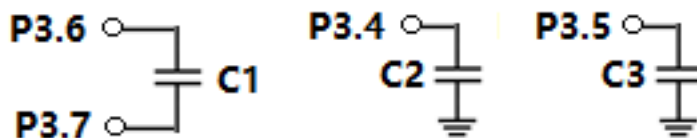
### 16.2.1 LCD 偏置电压

■ 配置电阻驱动模式：

由内部分压电阻产生驱动液晶的偏置电压，静态功耗的大小与分压电阻的阻值有关，分压电阻越大驱动能力越弱。

■ 配置电容驱动模式：

通过电容切换模式来产生偏置电压，功耗非常低，但是需要占用外部 IO 口



C1= C2 = C3: 10nF 或 100nF

### 16.2.2 LCD 寄存器说明

■ LCDCON(0xEB): 液晶控制寄存器 (默认值: 0x58)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
LCDEN	LCDWS	LCDRES	LDCBIAS	LCDDUTY[1:0]		LCDCLK	CPMPEN

- LCDEN:** 液晶驱动使能  
 1: 开启液晶驱动  
 0: 关闭液晶驱动  
 开启仅表示内部液晶模块工作，驱动液晶还需将相关 IO 配置为液晶模式
- LCDWS:** 液晶驱动波形选择  
 =0: A波形（位模式）  
 =1: B波形（帧模式）
- LCDRES:** 电阻驱动模式驱动输出内阻选择：  
 = 0 电阻50K模式  
 = 1 电阻16K模式
- LCDBIAS:** 液晶偏压选择：  
 = 0 1/2偏压  
 = 1 1/3偏压
- LCDDUTY[1:0]:** LCD 驱动 DUTY 选择  
 = 00(1/2duty);01(1/3duty);10(1/4duty);11(1/8duty);
- LCDCLK:** 选择驱动液晶的时钟源：  
 = 0 低频时钟分频：  
 = 1 主时钟分频：
- CPMPEN:** 升压模块使能  
 1: 开启升压模块  
 0: 关闭升压模块

■ LCDSEL(0xEC): 液晶选择寄存器（默认值：0x10）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
LCDCAP_EN[1:0]		CPMP_VOL[1:0]		LCDREG_SEL[3:0]			

- LDCAP\_EN[1:0]:** 液晶驱动模式  
 00: 电阻模式  
 01: 保留  
 10/11: 电容模式A
- CPMP\_VOL[1:0]:** 升压模块输出电压控制：  
 00: 2.5V  
 01: 2.8V  
 10: 3.3V  
 11: 5.0V
- LCDREG\_SEL[3:0]:** LCDDATA 的选通地址：  
 0000: 选通LCD\_DATA0  
 0001: 选通LCD\_DATA1  
 0010: 选通LCD\_DATA2  
 0011: 选通LCD\_DATA3  
 0100: 选通LCD\_DATA4  
 0101: 选通LCD\_DATA5  
 0110: 选通LCD\_DATA6  
 0111: 选通LCD\_DATA7  
 1000: 选通LCD\_DATA8

1001:	选通LCD_DATA9
1010:	选通LCD_DATA10
1011:	选通LCD_DATA11
1100:	选通LCD_DATA12
1101:	选通LCD_DATA13
1110:	选通LCD_DATA14
1111:	选通LCD_DATA15

■ LCDDAT(0xF9): 液晶数据寄存器 (默认值: 0x00)

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

◇ 该寄存器只写，不能读取

### 16.2.3 LCD 数据说明

■ 数据寄存器操作说明:

◇ 数据 SDI5229 的液晶驱动共有 16 个数据寄存器 LCD\_DATA0 - LCD\_DATA15，他们共同占用寄存器地址“LCDDAT(0xF9)”。如果要对哪个数据进行写操作，需要先用控制位“LCDREG\_SEL (LCDSEL[3:0])”配置好地址，然后在写入。

◇ LCD\_DATA0 - LCD\_DATA15 是只写寄存器，不能读取出来（也就是不能进行逻辑指令操作）

■ 数据寄存器和显示的对应关系

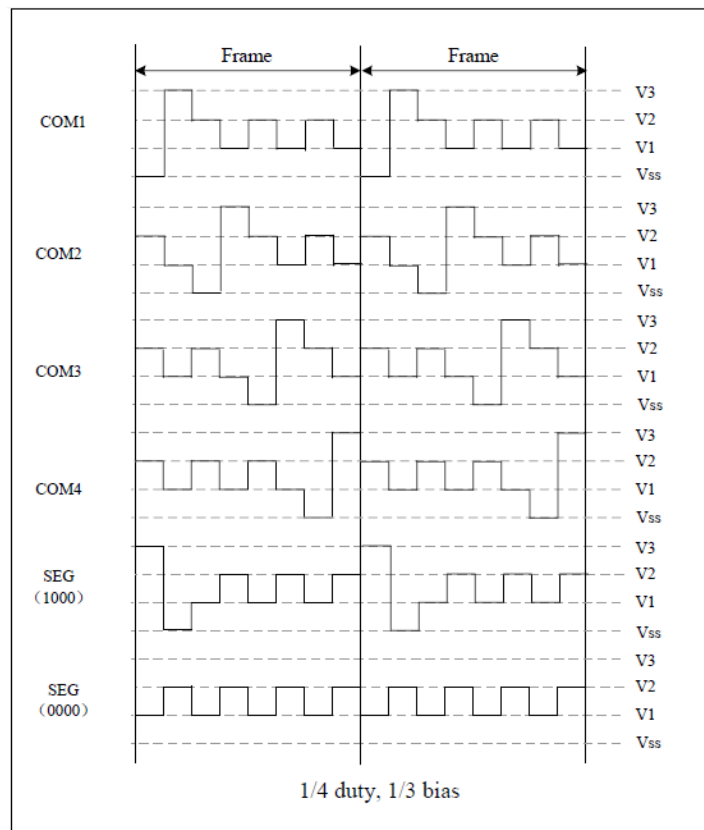
	4COM 模式		8COM 模式	
	Bit[7:4]	Bit[3:0]	Bit[7:4]	Bit[3:0]
LCD_DATA0	SEG1[3:0]	SEG0[3:0]	SEG5[7:4]	SEG4[7:4]
LCD_DATA1	SEG3[3:0]	SEG2[3:0]	SEG7[7:4]	SEG6[7:4]
LCD_DATA2	SEG5[3:0]	SEG4[3:0]	SEG5[3:0]	SEG4[3:0]
LCD_DATA3	SEG7[3:0]	SEG6[3:0]	SEG7[3:0]	SEG6[3:0]
LCD_DATA4	SEG9[3:0]	SEG8[3:0]	SEG9[3:0]	SEG8[3:0]
LCD_DATA5	SEG11[3:0]	SEG10[3:0]	SEG11[3:0]	SEG10[3:0]
LCD_DATA6	SEG13[3:0]	SEG12[3:0]	SEG13[3:0]	SEG12[3:0]
LCD_DATA7	SEG15[3:0]	SEG14[3:0]	SEG15[3:0]	SEG14[3:0]
LCD_DATA8	SEG17[3:0]	SEG16[3:0]	SEG17[3:0]	SEG16[3:0]
LCD_DATA9	SEG19[3:0]	SEG18[3:0]	SEG19[3:0]	SEG18[3:0]
LCD_DATA10			SEG9[7:4]	SEG8[7:4]
LCD_DATA11			SEG11[7:4]	SEG10[7:4]
LCD_DATA12			SEG13[7:4]	SEG12[7:4]
LCD_DATA13			SEG15[7:4]	SEG14[7:4]
LCD_DATA14			SEG17[7:4]	SEG16[7:4]

4				
LCD_DATA1			SEG19[7:4]	SEG18[7:4]
5				

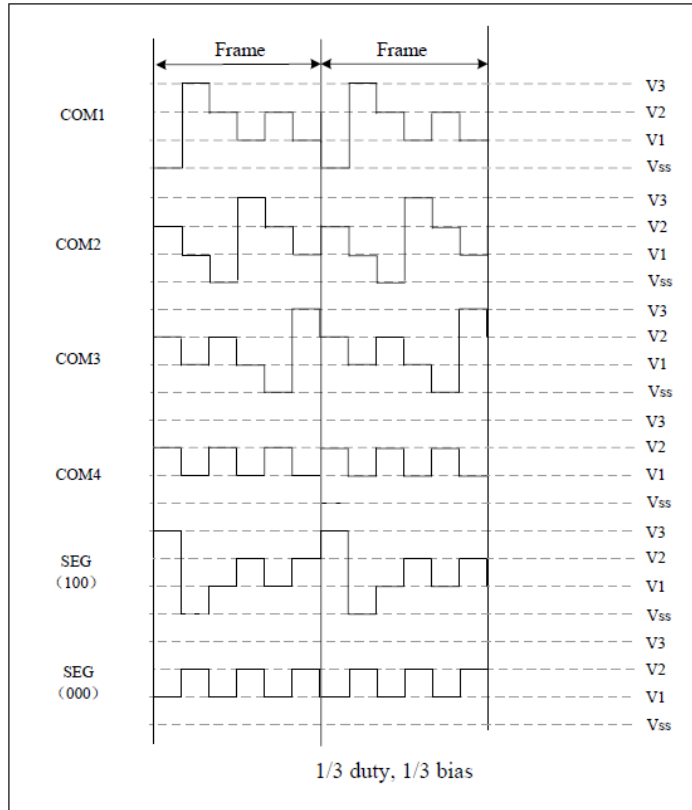
- ◇ 4COM 模式:
  - ❖ 仅使用 LCD\_DATA0 - LCD\_DATA9
  - ❖ SEG0[3:0]表示显示在 SEG0 对应的 COM3:COM0 的数据
- ◇ 8COM 模式:
  - ❖ SEG0[7:0]表示显示在 SEG0 对应的 COM7:COM0 的数据

### 16.2.4 LCD 驱动波形

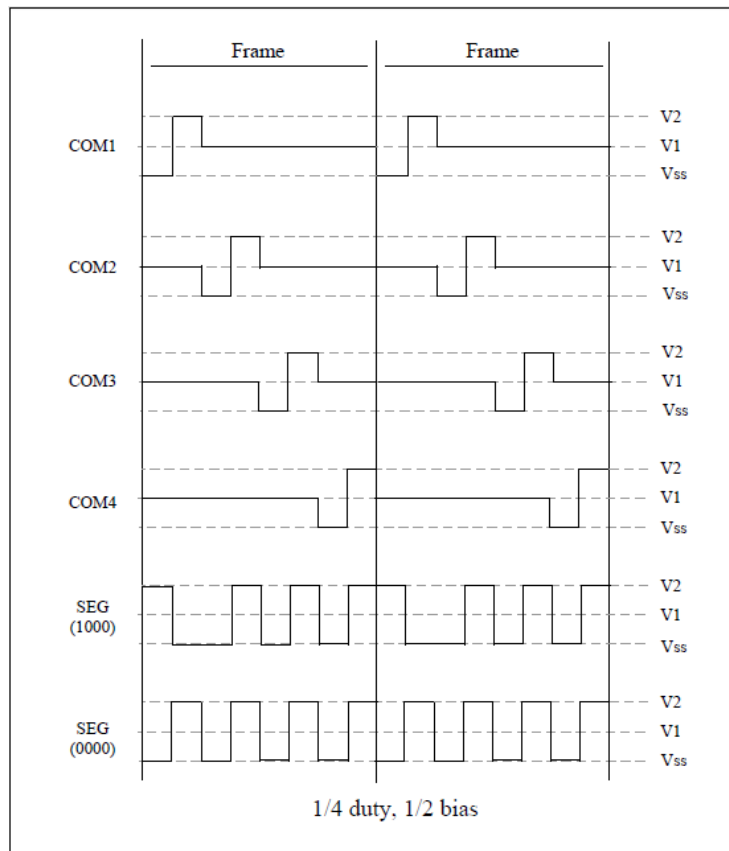
LCD 驱动波形分为 A 和 B 两种波形，通过寄存器 LCDWS 来选择：



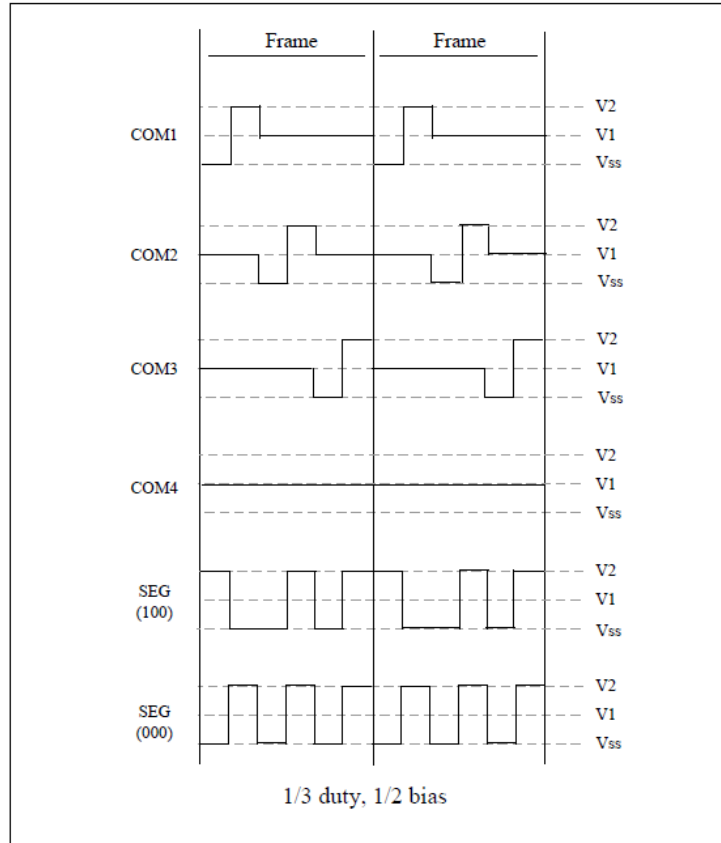
LCD 的 1/4duty 1/3bias 电源系统的时钟（A 波形）



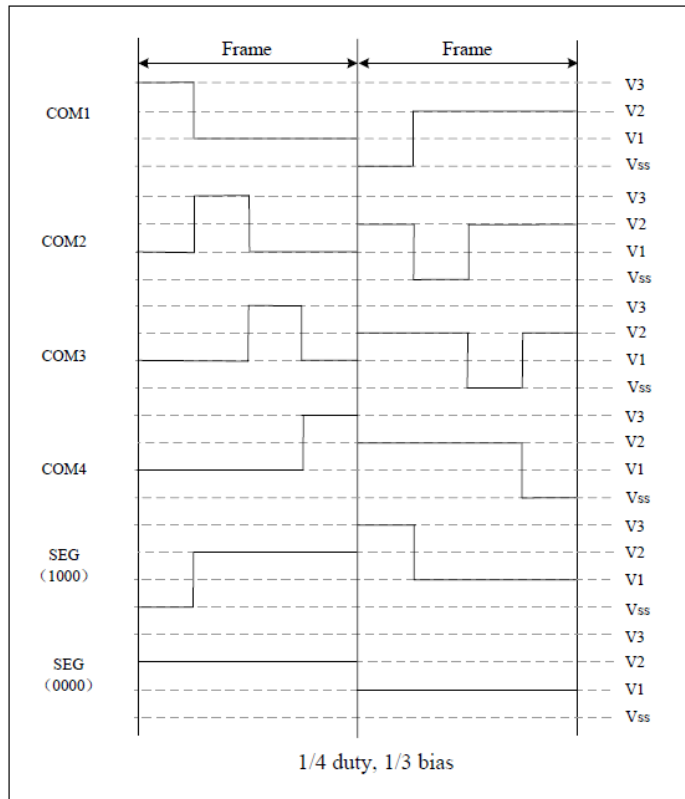
LCD 的 1/3duty 1/3bias 电源系统的时钟 (A 波形)



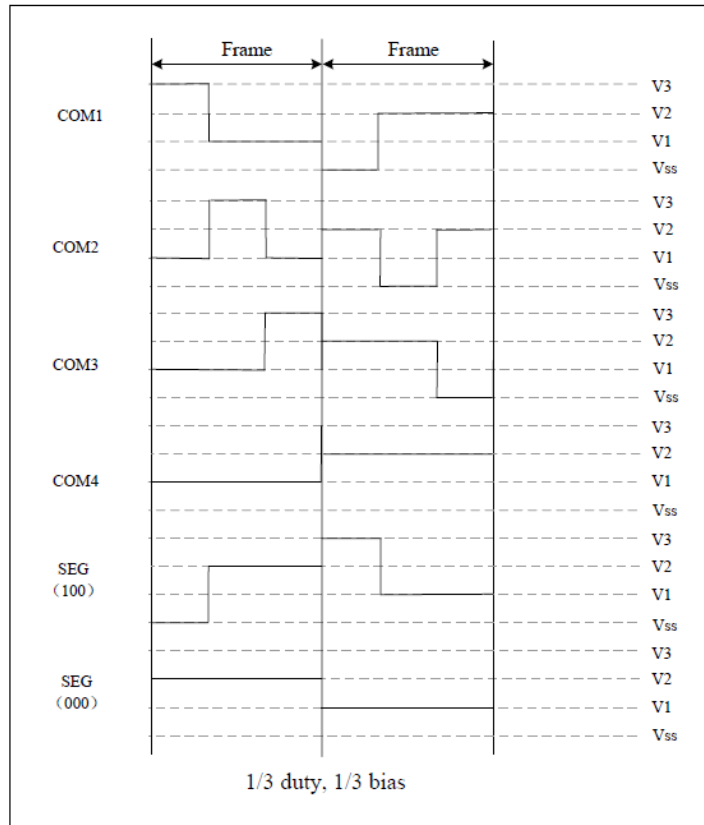
LCD 的 1/4duty 1/2bias 电源系统的时钟 (A 波形)



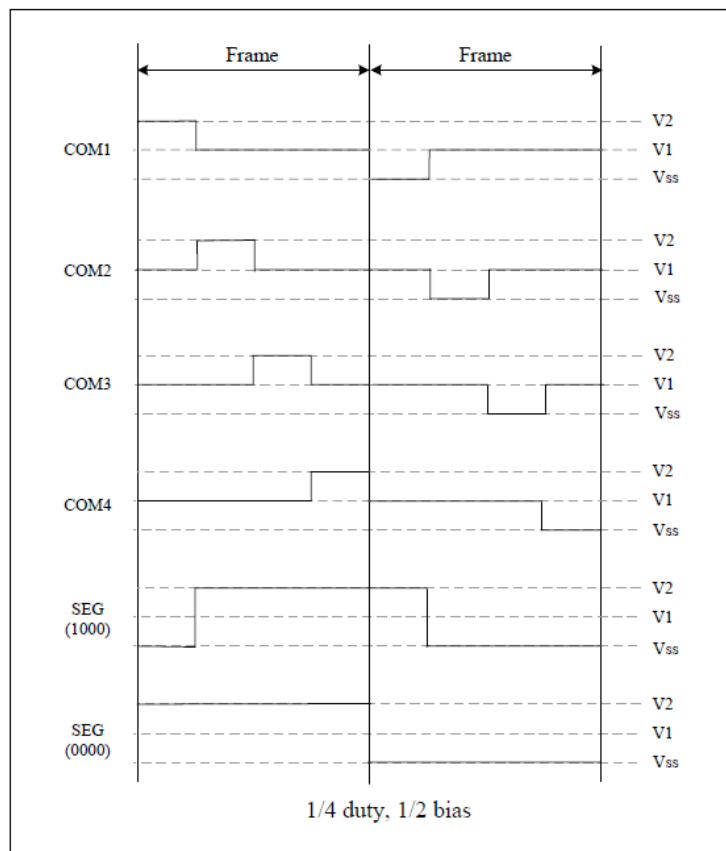
LCD 的 1/3duty 1/2bias 电源系统的时钟 (A 波形)



LCD 的 1/4duty 1/3bias 电源系统的时钟 (B 波形)

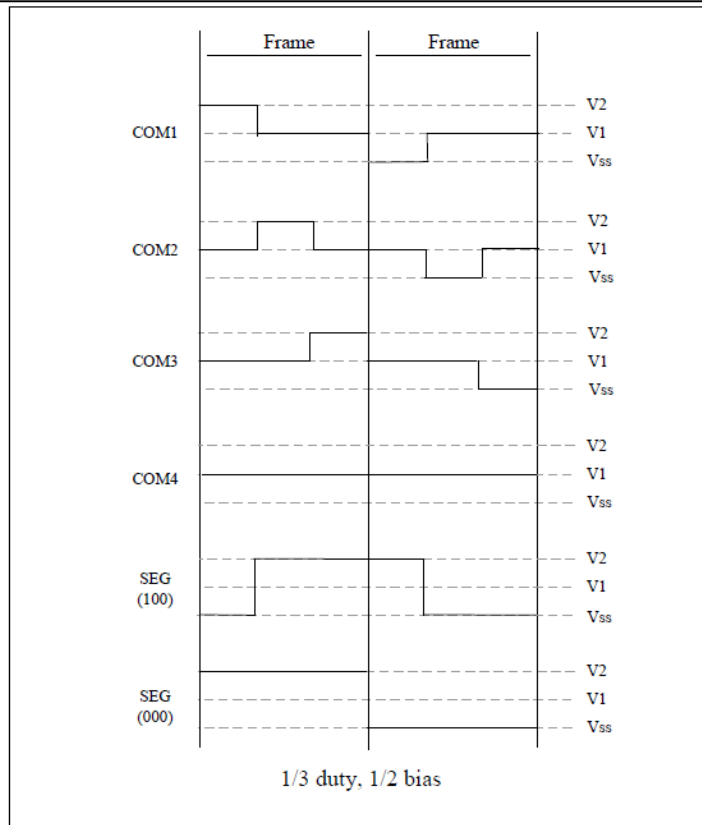


LCD 的 1/3duty 1/3bias 电源系统的时钟 (B 波形)



LCD 的 1/4duty 1/2bias 电源系统的时钟 (B 波形)





LCD 的 1/3duty 1/2bias 电源系统的时钟（B 波形）

## 17-I2C 接口

### ■ 概述

- ◇ 从机模式：
  - ❖ 收发基本“字节”过程由硬件自动处理,并产生相应中断, 能实现各种从机功能
  - ❖ I2C引脚和“按键中断”复用, 可以方便的通过唤醒处于休眠状态下的从机
  - ❖ 从机在收到主机数据后切换到发送模式时注意：
    - 要先切换模式, 才清除接收中断标志RXBUSY, 否则可能导致错误。
- ◇ 主机模式：
  - ❖ 整个收发过程由软件模拟,仅利用硬件自动识别“起始位”、“结束位”、“总线占用情况”：用于识别总线是否被占用(STOPEN)、数据通信时是否异收到“起始位”、“停止位”时的及时处理(STARTBIT)(STOPBIT);
- ◇ 高速时钟时, 需要在 SDA、SCL 信号 IO 外部采用上拉电阻（内部电阻为 30K 左右）

### 17.1 寄存器说明

#### ■ I2CCON(0xC8): I2C控制寄存器（默认值：0x10）

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0
ADDRBYTE	STOPEN	RXACK	RXMODE	RXINT_DIS	TXREQ	TXBUSY	RXBUSY

ADDRBYTE: “起始位”后第一字节标志：通常用来表示 I2C 通讯时的器件地址字节  
=1: 第一字节; =0: 其他字节;

STOPEN: I2C 通讯标志：表示通讯状态

	起始位时置0； 结束位置1；发送状态时，没有收到应答信号也回置1；
	❖ 主机模式时：可以用于识别线上是否有信号传输
	❖ 从机模式时：发送时可识别主机是否应答，没有就停止发送
RXACK:	用于“从机模式”：接收数据时是否产生应答信号 =1：产生应答信号； =0：不自动产生应答信号；
RXMODE:	用于“从机模式”：控制“发送模式”还是“接收模式” =1：接收模式 =0：发送模式
	❖ “结束位”和“发送时可识别主机是否应答”时会自动变为接收模式
RXINT_DIS:	用于“从机模式”：用于禁止产生接收中断 =1：禁止 =0：允许
	❖ 多个从机时，在第一字节核对地址后，希望不相关的从机不响应后续的数据字节时，可以将该位置1。起始位后，会自动将该位清0，所以该位不会阻止器件接收“第一字节”
TXREQ:	用于“从机模式”：发送完成中断标志 =1：发送完成，需要手动清0
TXBUSY:	用于“从机模式”：置1则启动发射（先准备好数据） ❖ 从机发送数据时，先将该位置1，当发送完成后，会自动清0。从机可以根据这个标志位决定是否要发送下一字节。 ❖ 如果在还没有发送完就写入I2COUT下一字节，会引起错误 ❖ 查询时和TXREQ功能差不多；TXREQ用于中断标志
RXBUSY:	用于“从机模式”：接收数据中断标志 ❖ 从机接收到数据后，会自动将该位置1，如果没有清除0，会一直将SCL拉低，用于通知主机，从机还没来得及处理该数据，不能发后续字节。

■ I2CCON2(0xCF): I2C控制寄存器2（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	I2CEN	STOPBIT	STARTBIT	SLAVEMOD

I2CEN:	I2C功能使能： =1：开启I2C功能 =0：关闭 ❖ “结束位”和“发送时可识别主机是否应答”时会自动变为接收模式
STOPBIT:	停止位标志： =1：表示目前产生了停止位； ❖ 起始位时会自动将该位清0
STARTBIT:	起始位标志： =1：表示目前产生了起始位； ❖ 停止位时会自动将该位清0
SLAVEMODE:	主从机模式设定

=1: 从机模式

=0: 主机模式

- ❖ 主机模式: SDI5229的主机模式并没有收发中断, 仅能通过相关标志查看“线上是否有数据”、“起始位”、“停止位”等等。用于配合多机通讯。
- ❖ 从机模式: SDI5229主要是适合硬件的从机模式

■ I2CIN (0xC9): I2C 接收数据寄存器 (默认值: 0x00)

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

◇ 仅用于“从机模式”接收: 接收到的数据存放在该寄存器

■ I2COUT(0xCA): I2C发送数据寄存器 (默认值: 0x00)

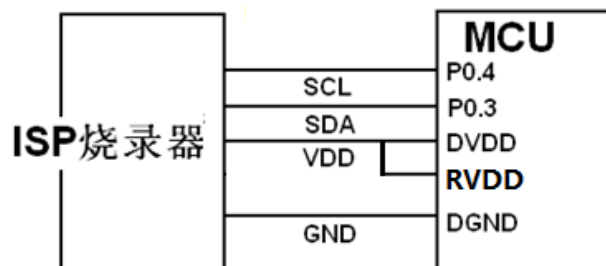
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit0

◇ 仅用于“从机模式”发射:

- ❖ 将发射数据写到该寄存器
- ❖ 切换到“发送模式”

## 18-在线 ISP 程序烧录

- ◇ SDI5229 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

## 19- 电器特性

### 19.1 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	-40 - 100	°C
结温	150	°C

### 19.2 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	1.8		5.5	V	
工作电流	IDD1		1.5		mA	9.83MHz
	IDD2		300u		uA	600KHz
STOP2 模式电流	IDD3		5		uA	<b>参考: 10.2.5 “低功耗说明”</b>
IO			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) 振荡器变化偏差小于 1%
IO 口驱动电流			30mA			
VCOM 输出参考 源温度系数			50		ppm	

### 19.3 ADC 参数

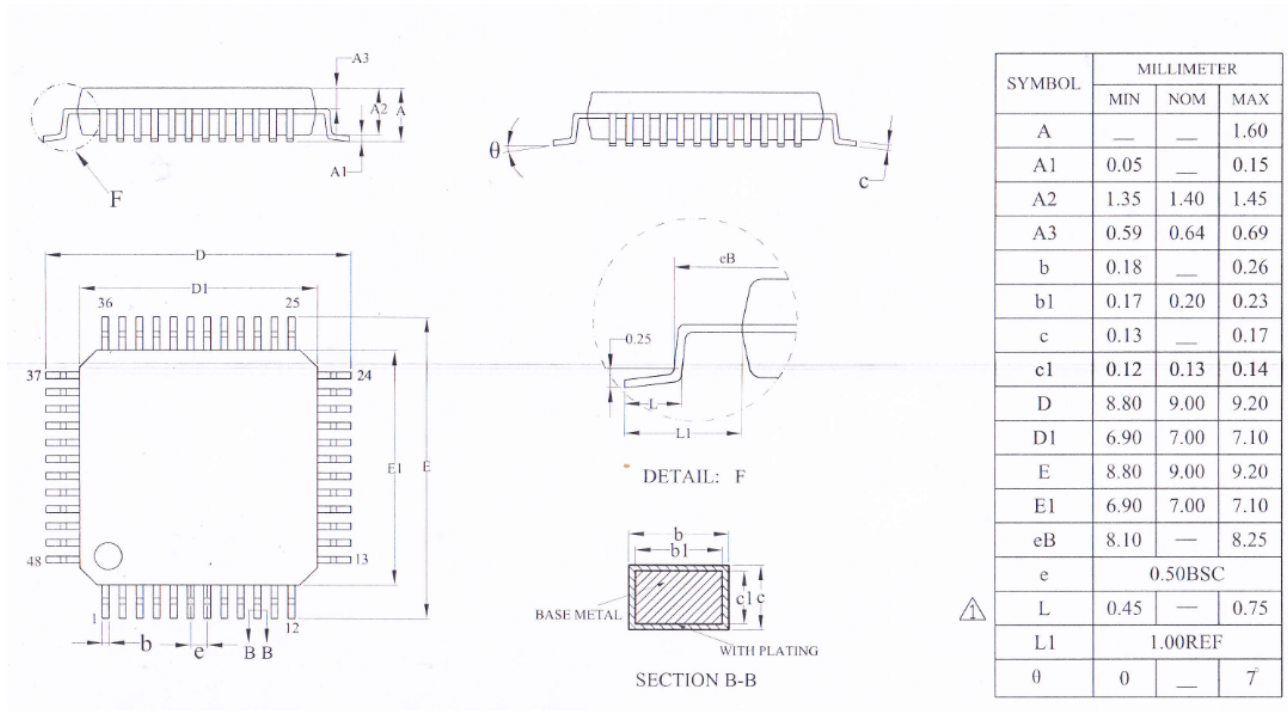
测试条件（如无特殊说明均采用此条件）：

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

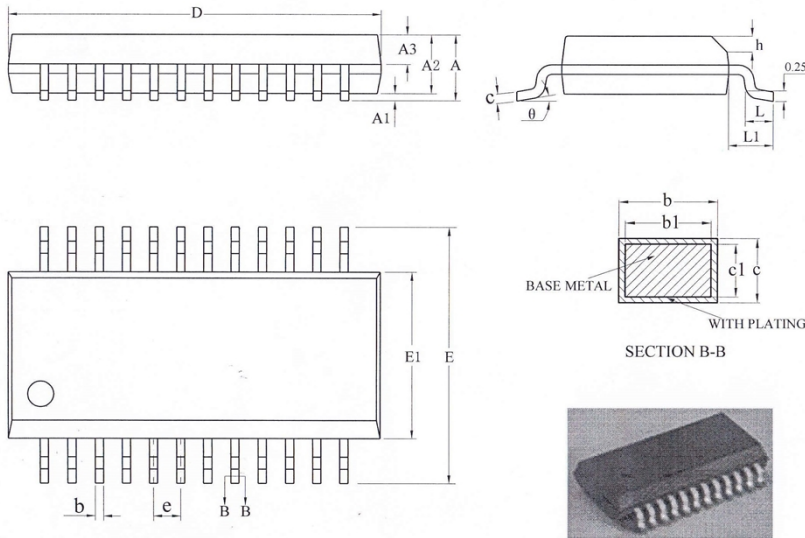
参数	符号	测试数据				测试条件 9.83MHz
		最小	典型	最大	单位	
差分电压输入 (VINP - VINN)		+/- 0.5 VREFP /128			V	
积分非线性 (INL)	PGA = 2		0.0002	0.001	% of FS	
	PGA = 128		0.0005	0.001	% of FS	
输入失调	PGA = 128		3	5	ppm of FS	
输入失调温漂			+/-10		$nV/^{\circ}C$	
增益误差			0.01		% of FS	
ADC 精度						参考：“14.3 噪声性能”

## 20-封装信息

### 20.1 LQFP48 (7.0\*7.0\*1.40) 外形尺寸



20.2 SSOP24 (0.635-D1.4) 外形尺寸



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	0.15	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	—	0.31
b1	0.22	0.25	0.28
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	8.55	8.65	8.75
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	0.635BSC		
h	0.30	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°