

*SinoMCU 8 位单片机*

# MC9902

## 用户手册

V1.1



上海晟矽微电子股份有限公司

Shanghai SinoMCU Microelectronics Co., Ltd.

## 目录

1	产品概要 .....	4
1.1	产品特性 .....	4
1.2	订购信息 .....	5
1.3	引脚排列 .....	5
1.4	端口说明 .....	6
2	电气特性 .....	7
2.1	极限参数 .....	7
2.2	直流电气特性 .....	7
2.3	交流电气特性 .....	8
2.4	ADC 特性参数 .....	8
3	CPU 及存储器 .....	9
3.1	指令集 .....	9
3.2	程序存储器 .....	11
3.3	数据存储器 .....	12
3.4	堆栈 .....	13
3.5	控制寄存器 .....	14
3.6	用户配置字 .....	17
4	系统时钟 .....	18
4.1	内部高频 RC 振荡器 .....	18
4.2	内部低频 RC 振荡器 .....	18
4.3	系统工作模式 .....	19
4.4	低功耗模式 .....	20
5	复位 .....	22
5.1	复位条件 .....	22
5.2	上电复位 .....	23
5.3	外部复位 .....	23
5.4	低电压复位 .....	23
5.5	看门狗复位 .....	23
6	I/O 端口 .....	24
6.1	通用 I/O 功能 .....	24
6.2	内部上拉电阻 .....	25
7	定时器 TIMER .....	26
7.1	看门狗定时器 WDT .....	26
7.2	定时器 T0 .....	26
7.3	定时器 T1 .....	28
8	模数转换器 ADC .....	31
8.1	ADC 概述 .....	31
8.2	ADC 相关寄存器 .....	31
8.3	ADC 操作步骤 .....	34
9	低电压检测 LVD .....	35
10	IIC 总线从机接口 .....	36
10.1	IIC 概述 .....	36

---

10.2	IIC 相关寄存器.....	36
10.3	IIC 通讯流程.....	37
11	中断.....	39
11.1	外部中断.....	39
11.2	定时器中断.....	39
11.3	ADC 中断.....	39
11.4	IIC 中断.....	40
11.5	中断相关寄存器.....	40
12	特性曲线.....	42
12.1	I/O 特性.....	42
12.2	功耗特性.....	44
12.3	模拟电路特性.....	46
13	封装尺寸.....	53
13.1	SOP14.....	53
13.2	SOP8.....	53
14	修订记录.....	54

## 1 产品概要

### 1.1 产品特性

- 8 位 CPU 内核
  - ◇ 精简指令集，6 级深度硬件堆栈
  - ◇ CPU 为双时钟，可在系统高/低频时钟之间切换
  - ◇ 系统高频时钟下 F<sub>CPU</sub> 可配置为 2/4/8/16 分频，系统低频时钟下 F<sub>CPU</sub> 固定为 2 分频
- 程序存储器
  - ◇ 2028×16 位 OTP 型程序存储器（烧录 1 次）
  - ◇ 1014×16 位 OTP 型程序存储器（烧录 2 次）
  - ◇ 可通过间接寻址读取程序存储器内容
- 数据存储器
  - ◇ 128 字节 SRAM 型通用数据存储器，支持直接寻址、间接寻址等多种寻址方式
- 2 组共 10 个 I/O
  - ◇ P0 (P00~P07)，P1 (P10~P11)
  - ◇ P11 为输入/开漏输出口，可复用为外部复位 RST 输入，编程时为高压 V<sub>PP</sub> 输入
  - ◇ 所有端口均内置上拉电阻，均可单独使能/禁用
- 系统时钟源
  - ◇ 内置高频 RC 振荡器 (8MHz)，可用作系统高频时钟源
  - ◇ 内置低频 RC 振荡器 (32KHz)，可用作系统低频时钟源
- 工作模式
  - ◇ 高速模式：CPU 在高频时钟下运行，低频时钟源工作
  - ◇ 低速模式：CPU 在低频时钟下运行，高频时钟源可选停止或工作
  - ◇ HOLD1 模式：CPU 停止运行，高频时钟源工作
  - ◇ HOLD2 模式：CPU 停止运行，高频时钟源停止工作，低频时钟源工作
  - ◇ 休眠模式：CPU 停止运行，所有时钟源停止工作
- 内部自振式看门狗计数器 (WDT)
  - ◇ 溢出时间可配置：16ms/64ms/256ms/1024ms/2048ms/4096ms
  - ◇ 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 2 个定时器
  - ◇ 8 位定时器 T0，可实现外部计数、BUZ 和 PWM 功能
  - ◇ 8 位定时器 T1，可实现外部计数、BUZ 和 PWM 功能
- 1 个 24 位全差分  $\Sigma$ - $\Delta$  型 ADC
  - ◇ 3 对外部差分输入通道：AN0/AN1、AN2/AN3、AN4/AN5，或 6 路负端内接 GND 的单输入通道：AN0~AN5
  - ◇ 内置可编程增益放大器 PGA：放大倍数 1/2/4/8/16/32/64/128/256/512 倍可选
  - ◇ 工作电压可选：VDD、内部 LDO 电压 V<sub>LDO</sub> (2.5V/2.8V/3.0V)
  - ◇ ADC 共模电压可通过端口 ACM 输出
- 1 组 IIC 总线从机接口
  - ◇ 支持 7 位地址编码的从机模式
  - ◇ 通讯速率最高支持 400Kbps
  - ◇ 地址匹配、接收或发送完成时触发中断

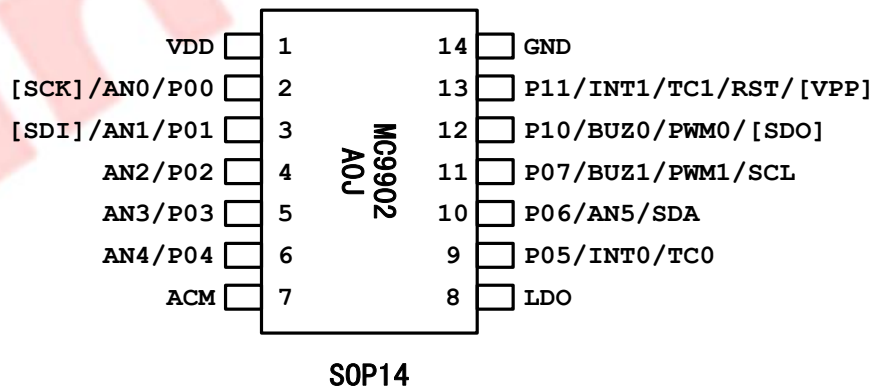
- 中断
  - ◇ 外部中断 (INT0~INT1)
  - ◇ 定时器中断 (T0~T1)
  - ◇ ADC 中断
  - ◇ IIC 中断
- 低电压复位 LVR: 2.0V/2.5V
- 低电压检测 LVD: 2.0V~3.4V (15 级)
- 工作电压
  - ◇ VLVR25 ~ 3.6V @ Fcpu = 0~4MHz
  - ◇ VLVR20 ~ 3.6V @ Fcpu = 0~2MHz
- 封装形式: SOP14/SOP8

## 1.2 订购信息

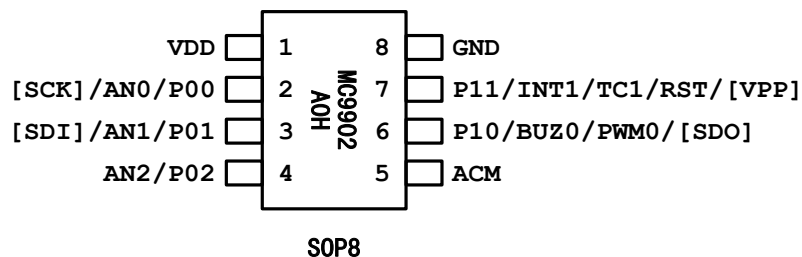
产品名称	封装形式	备注
MC9902A0J	SOP14	
MC9902A0H	SOP8	
MC9902A1H	SOP8	
MC9902A2H	SOP8	

## 1.3 引脚排列

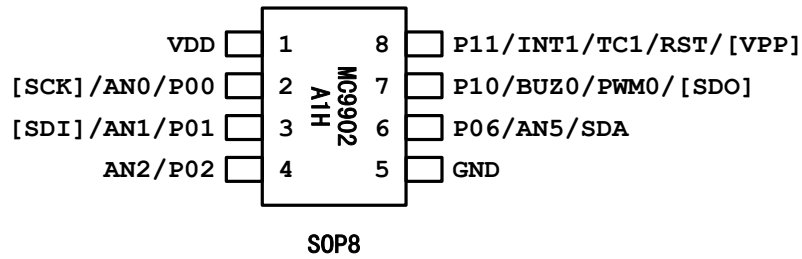
### MC9902A0J



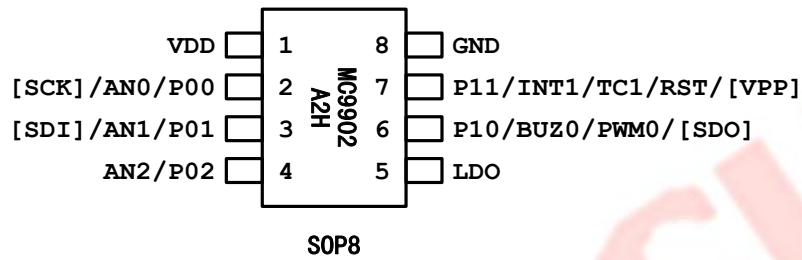
### MC9902A0H



MC9902A1H



MC9902A2H



1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P00~P07, P10	D	GPIO, 内部上拉
P11	D	开漏 IO, 内部上拉
INT0~INT1	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
BUZ0/PWM0	DO	定时器 T0 的 BUZ/PWM 输出
BUZ1/PWM1	DO	定时器 T1 的 BUZ/PWM 输出
AN0~AN5	AI	ADC 模拟输入通道
ACM	A	ADC 内部共模电压输出, 需外接 22~100nF 对地电容
LDO	A	LDO 输出, 需外接 1~10uF 对地电容
SCL, SDA	D	IIC 通讯时钟/数据接口, 开漏输出
RST	DI	外部复位输入
SCK, SDI, SDO	D	编程时钟/数据输入/数据输出接口
VPP	P	编程高压输入

注：P-电源；D-数字输入输出，DI-数字输入，DO-数字输出；A-模拟输入输出，AI-模拟输入，AO-模拟输出。

## 2 电气特性

### 2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~4.0	V
输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	°C
储存温度	Tstg	-65~150	°C

注：若芯片工作条件超过极限值，则会造成永久性损坏；若芯片长时间工作在极限条件下，则会影响其可靠性。

### 2.2 直流电气特性

VDD=3V, T=25°C

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=4MHz@FHIRC/2	VLVR25		3.6	V
			Fcpu=2MHz@FHIRC/4	VLVR20		3.6	
			Fcpu=16KHz@FLIRC/2	VLVR20		3.6	
输入漏电流	Ileak	所有输入脚		-1		1	uA
输入高电平	Vih	所有输入脚		0.8VDD			V
输入低电平	Vil	所有输入脚				0.2VDD	V
输出拉电流	Ioh	推挽输出脚	Voh=0.9VDD		10		mA
输出灌电流	Iol	所有输出脚	Vol=0.1VDD		10		mA
上拉电阻	Rpu1	P0, P1	Vin=0, IICRS=0, RESSEL 配置		150		KΩ
			Vin=0, IICRS=0, RESSEL 配置		60		KΩ
	Rpu2	SCL, SDA	Vin=0, IICRS=1		4.7		KΩ
运行模式功耗	Irun	VDD	Fcpu=4MHz@HIRC		1.3		mA
			Fcpu=2MHz@HIRC		1.1		mA
			Fcpu=32KHz/2@LIRC		200		uA
HOLD1 功耗	Ihold1	VDD	CPU 停, HIRC/LIRC 开		250		uA
HOLD2 功耗	Ihold2	VDD	CPU 停, HIRC 关, LIRC 开		1	5	uA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.1	3	
低压检测电压	VLVD	VDD		-5%		+5%	V
低压复位电压	VLVR25	VDD		-10%	2.5	+10%	V
	VLVR20			-10%	2.0	+10%	

注：功耗特性参数的条件说明中，诸如 HIRC/LIRC/WDT/LVR/LVD/ADC 等未注明模块，默认其为关闭状态。

## 2.3 交流电气特性

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	F <sub>HIRC</sub>	VDD=3V, T=25°C	-2%	8	+2%	MHz
		VDD=2.0V~3.6V, T=-20°C~70°C	-3%		+3%	MHz
		VDD=2.0V~3.6V, T=-40°C~85°C	-5%		+5%	MHz
LIRC 振荡频率	F <sub>LIRC</sub>	VDD=3V, T=25°C	-50%	32	+50%	KHz

## 2.4 ADC 特性参数

VDD=3V, T=25°C

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V <sub>ADC</sub>	选择 VDD	2.5		3.6	V
		选择 LDO 输出电压 V <sub>LDO</sub>	2.4	2.5	2.6	
			2.7	2.8	2.9	
			2.9	3.0	3.1	
LDO 有效工作电压		V <sub>LDO</sub> +0.5			V	
ADC 动态电流	I <sub>ADD</sub>	VDD=5V, AD 转换中		3.5		mA
ADC 静态电流	I <sub>ADS</sub>	VDD=5V, ADC 关闭			1	uA
ACM 电压	V <sub>ACM</sub>		0.8	1.1	1.4	V
ACM 参考电压温度系数			100			ppm/°C
模拟差分输入范围 (INP)-(INN)		通过 ADGAS 选择放大倍数			V <sub>ACM</sub> /(2×放大倍数)	V
共模输入电压					V <sub>ADC</sub> /2	V
差分输入阻抗	Z <sub>AIN</sub>	通过 ADGAS 选择放大倍数		256/放大倍数		KΩ
转换时钟	F <sub>ADC</sub>		250		500	KHz
分辨率		F <sub>ADC</sub> =250KHz, VLDOS[1:0]=11		15		Bits



### 3 CPU 及存储器

#### 3.1 指令集

芯片的指令集为精简指令集。除程序跳转类指令，其余指令均为单周期指令，即执行时间为 1 个指令周期；所有指令均为单字指令，即指令码只占用 1 个程序存储器地址空间。

#### 指令汇总表

助记符	说明	操作	周期	长度	标志
ADDAR R	R 和 ACC 相加，结果存入 ACC	R+ACC→ACC	1	1	C,DC,Z
ADDRA R	R 和 ACC 相加，结果存入 R	R+ACC→R	1	1	C,DC,Z
ADGAR R	R 和 ACC 相加 (带 C 标志)，结果存入 ACC	R+ACC+C→ACC	1	1	C,DC,Z
ADGRA R	R 和 ACC 相加 (带 C 标志)，结果存入 R	R+ACC+C→R	1	1	C,DC,Z
RSUBAR R	R 和 ACC 相减，结果存入 ACC	R-ACC→ACC	1	1	C,DC,Z
RSUBRA R	R 和 ACC 相减，结果存入 R	R-ACC→R	1	1	C,DC,Z
RSBCAR R	R 和 ACC 相减 (带 C 标志)，结果存入 ACC	R-ACC-/C→ACC	1	1	C,DC,Z
RSBCRA R	R 和 ACC 相减 (带 C 标志)，结果存入 R	R-ACC-/C→R	1	1	C,DC,Z
ASUBAR R	ACC 和 R 相减，结果存入 ACC	ACC-R→ACC	1	1	C,DC,Z
ASUBRA R	ACC 和 R 相减，结果存入 R	ACC-R→R	1	1	C,DC,Z
ASBCAR R	ACC 和 R 相减 (带 C 标志)，结果存入 ACC	ACC-R-/C→ACC	1	1	C,DC,Z
ASBCRA R	ACC 和 R 相减 (带 C 标志)，结果存入 R	ACC-R-/C→R	1	1	C,DC,Z
ANDAR R	R 和 ACC 与操作，结果存入 ACC	R and ACC→ACC	1	1	Z
ANDRA R	R 和 ACC 与操作，结果存入 R	R and ACC→R	1	1	Z
ORAR R	R 和 ACC 或操作，结果存入 ACC	R or ACC→ACC	1	1	Z
ORRA R	R 和 ACC 或操作，结果存入 R	R or ACC→R	1	1	Z
XORAR R	R 和 ACC 异或操作，结果存入 ACC	R xor ACC→ACC	1	1	Z
XORRA R	R 和 ACC 异或操作，结果存入 R	R xor ACC→R	1	1	Z
COMAR R	对 R 取反，结果存入 ACC	R 取反→ACC	1	1	Z
COMR R	对 R 取反，结果存入 R	R 取反→R	1	1	Z
RLA	ACC 循环左移 (带 C 标志)	ACC[7]→C ACC[6:0]→ACC[7:1] C→ACC[0]	1	1	C
RLAR R	R 循环左移 (带 C 标志)，结果存入 ACC	R[7]→C R[6:0]→ACC[7:1] C→ACC[0]	1	1	C
RLR R	R 循环左移 (带 C 标志)，结果存入 R	R[7]→C R[6:0]→R[7:1] C→R[0]	1	1	C
RRA	ACC 循环右移 (带 C 标志)	C→ACC[7] ACC[7:1]→ACC[6:0] ACC[0]→C	1	1	C

RRAR	R	R 循环右移 (带 C 标志), 结果存入 ACC	R[0]→C R[7:1]→ACC[6:0] C→ACC[7]	1	1	C
RRR	R	R 循环右移 (带 C 标志), 结果存入 R	R[0]→C R[7:1]→R[6:0] C→R[7]	1	1	C
SWAPAR	R	交换 R 的高低半字节, 结果存入 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	1	-
SWAPR	R	交换 R 的高低半字节, 结果存入 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	1	-
MOVRA	R	将 ACC 存入 R	ACC→R	1	1	-
MOVAR	R	将 R 存入 ACC	R→ACC	1	1	Z
MOVR	R	将 R 存入 R	R→R	1	1	Z
CLRA		将 ACC 清 0	0→ACC	1	1	Z
CLRR	R	将 R 清 0	0→R	1	1	Z
INCA		ACC 自加 1	ACC+1→ACC	1	1	-
INCR	R	R 自加 1	R+1→R	1	1	Z
INCAR	R	R 加 1, 结果存入 ACC	R+1→ACC	1	1	Z
DECA		ACC 自减 1	ACC-1→ACC	1	1	-
DECR	R	R 自减 1	R-1→R	1	1	Z
DECAR	R	R 减 1, 结果存入 ACC	R-1→ACC	1	1	Z
JZA		ACC 自加 1; 结果为 0 则跳过下一条指令	ACC+1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
JZR	R	R 自加 1; 结果为 0 则跳过下一条指令	R+1→R; 结果为 0 则 PC+2→PC	1/2	1	-
JZAR	R	R 加 1, 结果存入 ACC; 结果为 0 则跳过下一条指令	R+1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
DJZA		ACC 自减 1; 结果为 0 则跳过下一条指令	ACC-1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
DJZR	R	R 自减 1; 结果为 0 则跳过下一条指令	R-1→R; 结果为 0 则 PC+2→PC	1/2	1	-
DJZAR	R	R 减 1, 结果存入 ACC; 结果为 0 则跳过下一条指令	R-1→ACC; 结果为 0 则 PC+2→PC	1/2	1	-
BCLR	R, b	将 R 的第 b 位清 0	0→R[b]	1	1	-
BSET	R, b	将 R 的第 b 位置 1	1→R[b]	1	1	-
JBCLR	R, b	若 R 的第 b 位为 0, 则跳过下一条指令	若 R[b]=0, 则 PC+2→PC	1/2	1	-
JBSET	R, b	若 R 的第 b 位为 1, 则跳过下一条指令	若 R[b]=1, 则 PC+2→PC	1/2	1	-
ADDAI	K	K 和 ACC 相加, 结果存入 ACC	K+ACC→ACC	1	1	C,DC,Z
ADCAI	K	K 和 ACC 相加 (带 C 标志), 结果存入 ACC	K+ACC+C→ACC	1	1	C,DC,Z
ISUBAI	K	K 和 ACC 相减, 结果存入 ACC	K-ACC→ACC	1	1	C,DC,Z
ISBCAI	K	K 和 ACC 相减 (带 C 标志), 结果存入 ACC	K-ACC-/C→ACC	1	1	C,DC,Z
ASUBAI	K	ACC 和 K 相减, 结果存入 ACC	ACC-K→ACC	1	1	C,DC,Z
ASBCAI	K	ACC 和 K 相减 (带 C 标志), 结果存入 ACC	ACC-K-/C→ACC	1	1	C,DC,Z
ANDAI	K	K 和 ACC 与操作, 结果存入 ACC	K and ACC→ACC	1	1	Z
ORAI	K	K 和 ACC 或操作, 结果存入 ACC	K or ACC→ACC	1	1	Z
XORAI	K	K 和 ACC 异或操作, 结果存入 ACC	K xor ACC→ACC	1	1	Z
MOVAI	K	将 K 存入 ACC	K→ACC	1	1	-
CALL	K	子程序调用	PC+1→TOS K→PC[12:0]	2	1	-

GOTO	K	无条件跳转	K→PC[12:0]	2	1	-
RETURN		从子程序返回	TOS→PC	2	1	-
RETAI	K	从子程序返回, 并将 K 存入 ACC	TOS→PC K→ACC	2	1	-
RETIE		从中断返回	TOS→PC 1→GIE	2	1	-
NOP		空操作	空操作	1	1	-
DAA		BCD 码加法后, 将 ACC 的值调整为 BCD 码	ACC(HEX 码)→ACC(BCD 码)	1	1	C
DSA		BCD 码减法后, 将 ACC 的值调整为 BCD 码	ACC(HEX 码)→ACC(BCD 码)	1	1	-
CLRWDT		将看门狗定时器清 0	0→WDT	1	1	TO,PD
STOP		进入低功耗模式	0→WDT ; CPU 暂停	1	1	TO,PD
MULAR	R	R 和 ACC 相乘, 结果存入[HIBYTE:ACC]	R×ACC→[HIBYTE:ACC]	1	1	-
MULRA	R	R 和 ACC 相乘, 结果存入[HIBYTE:R]	R×ACC→[HIBYTE:R]	1	1	-

注:

1. ACC-算数逻辑单元累加器, R-数据存储寄存器, K-立即数;
2. 对于条件跳转类指令, 若跳转条件成立, 则指令需 2 个周期, 否则只需 1 个周期;

### 3.2 程序存储器

芯片的程序存储器为 OTP 型存储器, 可通过用户配置字 PAGE 选择存储器的地址空间范围。

2028×16 位的地址空间范围为 0000H~07EBH, 仅能烧录 1 次, 地址分配如下图所示:

复位起始地址 (0000H)
通用程序区 (0001H - 0007H)
中断入口地址 (0008H)
通用程序区 (0009H - 07EBH)

1014×16 位的地址空间范围为 0000H~03F5H, 可以烧录 2 次, 地址分配如下图所示:

复位起始地址 (0000H)
通用程序区 (0001H - 0007H)
中断入口地址 (0008H)
通用程序区 (0009H - 03F5H)

程序存储器支持间接寻址，可通过寄存器 INDF3 访问  $FSR1 \times 256 + FSR0$  指向的程序存储器地址中的内容。例：通过间接寻址读取程序存储器 0155H 地址中的内容，高 8 位存入数据存储器 11H 地址，低 8 位存入数据存储器 10H 地址

```

MOVAI    01H
MOVRA    FSR1           ; 将 01H 写入 FSR1
MOVAI    55H
MOVRA    FSR0           ; 将 55H 写入 FSR0
MOVAR    INDF3          ; 读取 FSR1 × 256 + FSR0 指向的程序存储器地址 (0155H)
                                ; 中的内容，高 8 位存入 HIBYTE，低 8 位存入 A 寄存器
MOVRA    10H           ; 低 8 位存入数据存储器 10H 地址
MOVAR    HIBYTE        ; 从 HIBYTE 读取高 8 位
MOVRA    11H           ; 高 8 位存入数据存储器 11H 地址
    
```

### 3.3 数据存储器

数据存储器包括通用数据存储器 GPR 和特殊功能寄存器 SFR，具体地址分配参照下表。GPR 可直接寻址或通过 INDF0/INDF2 间接寻址，SFR 可直接寻址或通过 INDF1/INDF2 间接寻址。

**数据存储器区地址映射表**

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-07FH	GPR	通用数据存储器区							
080H-17FH	保留	保留							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE	INTF	OSCMR	LVDCR		
190H-197H		IOP0	OEP0	PUP0	RUPSEL	IOP1	OEP1	PUP1	
198H-19FH									
1A0H-1A7H		T0CR	T0CNT	T0LOAD	T0DATA	T1CR	T1CNT	T1LOAD	T1DATA
1A8H-1AFH									
1B0H-1B7H		ADCR0	ADCR1	ADCR2		ADRH	ADRM	ADRL	
1B8H-1BFH		IICCR	IICAR	IICDR					
1C0H-1FFH	保留								

**注：上表中灰色部分地址为系统保留区，读出数据不确定，写入操作可能会影响芯片正常工作。**

**数据存储器寻址模式地址组成**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址模式
0	0	0	0	0	0	0	来自指令的 9 位地址								直接寻址模式	
0	0	0	0	0	0	0	0	FSR0								间接寻址模式 0
0	0	0	0	0	0	0	1	FSR1								间接寻址模式 1
FSR1							FSR0								间接寻址模式 2	

直接寻址模式，是以指令的低 9 位为数据存储器地址，通过指令访问，寻址范围为 000H~1FFH。

例：通过直接寻址模式将数据 55H 写入数据存储器 010H 地址

```
MOVAI    55H
MOVRA    10H           : 将数据 55H 写入数据存储器 10H 地址
```

间接寻址模式 0，是以 FSR0 为数据存储器地址指针，通过 INDF0 访问，寻址范围为 000H~0FFH。

例：通过间接寻址模式 0 将数据 55H 写入数据存储器 010H 地址

```
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF0        : 将数据 55H 写入 FSR0 指向的数据存储器中
```

间接寻址模式 1，是以 FSR1 为数据存储器地址指针，通过 INDF1 访问，寻址范围为 100H~1FFH。

例：通过间接寻址模式 1 将数据 55H 写入数据存储器 110H 地址

```
MOVAI    10H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF1        : 将数据 55H 写入 FSR1 指向的数据存储器中
```

间接寻址模式 2，是以 [FSR1:FSR0] 为数据存储器地址指针，通过 INDF2 访问，寻址范围为 0000H ~ FFFFH。例：通过间接寻址模式 2 将数据 55H 写入数据存储器 0110H 地址

```
MOVAI    01H
MOVRA    FSR1
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF2        : 将数据 55H 写入 FSR1×256+FSR0 指向的数据存储器中
```

**注：间接寻址模式 2 可寻址 0~FFFFH，但访问数据存储器中未定义的地址时，读出数据不确定，写入数据可能会更改其他地址中的数据。**

### 3.4 堆栈

6 级堆栈深度，当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压栈保存；当执行中断返回指令或子程序返回指令时，栈顶数据自动出栈载入 PC。

### 3.5 控制寄存器

#### 间接寻址寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF0</b>	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

INDF0: INDF0 不是物理寄存器，对 INDF0 寻址实际是对 FSR0 指向的数据存储器地址进行访问，从而实现间接寻址功能。

#### 间接寻址寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF1</b>	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

INDF1: INDF1 不是物理寄存器，对 INDF1 寻址实际是对 FSR1+256 指向的数据存储器地址进行访问，从而实现间接寻址功能。

#### 间接寻址寄存器 2

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF2</b>	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

INDF2: INDF2 不是物理寄存器，对 INDF2 寻址实际是对 FSR1×256+FSR0 指向的数据存储器地址进行访问，从而实现间接寻址功能。

#### 间接寻址寄存器 3

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INDF3</b>	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **INDF3[7:0]** – 间接寻址寄存器 3

INDF3: INDF3 不是物理寄存器，对 INDF3 寻址实际是对 FSR1×256+FSR0 指向的程序存储器地址进行访问，从而实现间接寻址功能。

**注：寄存器 INDF3 仅可使用读取指令 (MOVAR INDF3) 进行读取操作，读取内容高 8 位存入 HIBYTE 寄存器，低 8 位存入 A 寄存器。**

### 字操作高字节缓存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>HIBYTE</b>	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节缓存器

HIBYTE: 用于存放通过 INDF3 访问程序存储器时所读取内容的高 8 位数据。

### 数据指针寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR0</b>	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

FSR0: 间接寻址模式 0 的指针, 或间接寻址模式 2、3 的指针低 8 位。

### 数据指针寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>FSR1</b>	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

FSR1: 间接寻址模式 1 的指针, 或间接寻址模式 2、3 的指针高 8 位。

### 程序指针计数器低字节

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PCL</b>	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **PC[7:0]** – 程序指针计数器低 8 位

程序指针计数器 (PC) 有以下几种操作模式:

- ◇ 顺序运行指令:  $PC = PC + 1$ ;
- ◇ 分支指令 GOTO/CALL:  $PC =$  指令码低 13 位;
- ◇ 返回指令 RETIE/RETURN/RETAI:  $PC =$  堆栈栈顶 (TOS);

对 PCL 操作指令:

- ◇ 对 PCL 操作的加法指令:  $PC = (PC[12:0] + ALU[7:0])$ ;
- ◇ 对 PCL 操作的其它指令:  $PC = \{PC[12:8]; ALU[7:0](ALU \text{ 运算结果})\}$ ;

**CPU 状态寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PFLAG</b>	-	-	-	-	-	Z	DC	C
<b>R/W</b>	-	-	-	-	-	R/W	R/W	R/W
<b>初始值</b>	-	-	-	-	-	X	X	X

- BIT[2]     **Z** – 零标志位  
 0: 算术或逻辑运算的结果不为零;  
 1: 算术或逻辑运算的结果为零;
- BIT[1]     **DC** – 半字节进/借位标志位  
 0: 加法运算时半字节无进位; 减法运算时半字节有借位;  
 1: 加法运算时半字节有进位; 减法运算时半字节无借位;
- BIT[0]     **C** – 进/借位标志位  
 0: 加法运算时无进位; 减法运算时有借位; 移位后移出逻辑 0;  
 1: 加法运算时有进位; 减法运算时无借位; 移位后移出逻辑 1;

**杂项控制寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MCR</b>	GIE	-	TO	PD	INT1M1	INT1M0	INT0M1	INT0M0
<b>R/W</b>	R/W	-	R	R	R/W	R/W	R/W	R/W
<b>初始值</b>	0	-	0	0	0	0	0	0

- BIT[7]     **GIE** – 中断总使能位  
 0: 屏蔽所有中断;  
 1: 由相应的中断使能位决定 CPU 是否响应中断源所触发的中断;
- BIT[5]     **TO** – 看门狗溢出标志位  
 0: 上电复位, 或执行 CLRWDT/STOP 指令;  
 1: 发生 WDT 溢出;
- BIT[4]     **PD** – 进入低功耗模式标志位  
 0: 上电复位, 或执行 CLRWDT 指令;  
 1: 执行 STOP 指令;
- BIT[3:2]   **INT1M[1:0]** – 外部中断 INT1 触发方式选择位

INT1M[1:0]	INT1 触发方式
00	上升沿触发
01	下降沿触发
10	电平变化触发
11	





## 4 系统时钟

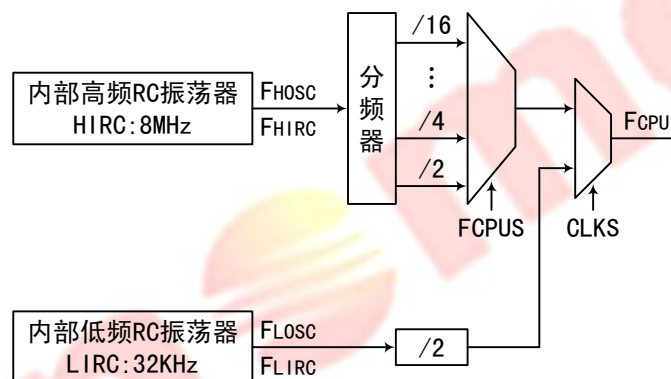
芯片为双时钟系统，内部电路均在系统高频时钟  $F_{HOSC}$  或系统低频时钟  $F_{LOSC}$  下工作，部分模块的时钟还可在  $F_{HOSC}$  和  $F_{LOSC}$  之间切换。

系统高频时钟  $F_{HOSC}$  固定为内部高频 RC 振荡器  $HIRC$  (8MHz) 时钟  $F_{HIRC}$ ；系统低频时钟  $F_{LOSC}$  固定为内部低频 RC 振荡器  $LIRC$  (32KHz) 时钟  $F_{LIRC}$ 。

CPU 的时钟源可在系统高频时钟  $F_{HOSC}$  和系统低频时钟  $F_{LOSC}$  之间切换。选择  $F_{HOSC}$  时，CPU 的时钟频率  $F_{CPU}$  通过用户配置字  $FCPUS$  选择；选择  $F_{LOSC}$  时， $F_{CPU}$  固定为  $F_{LOSC}$  的 2 分频。

WDT（看门狗）电路的时钟源固定为内部低频 RC 振荡器。

### 系统时钟示意图



### 4.1 内部高频 RC 振荡器

芯片内置 1 个振荡频率为 8MHz 的高精度  $HIRC$  振荡器，可用作系统高频时钟源。

### 4.2 内部低频 RC 振荡器

芯片内置 1 个振荡频率典型值为 32KHz 的  $LIRC$  振荡器，可用作系统低频时钟源，也用于系统上电延时控制、WDT 定时器等电路。

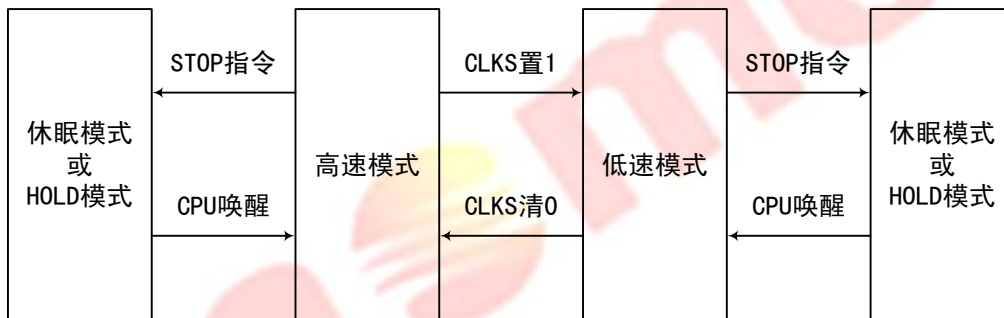
### 4.3 系统工作模式

芯片支持高速模式、低速模式、HOLD1 模式、HOLD2 模式和休眠模式等多种系统工作模式。

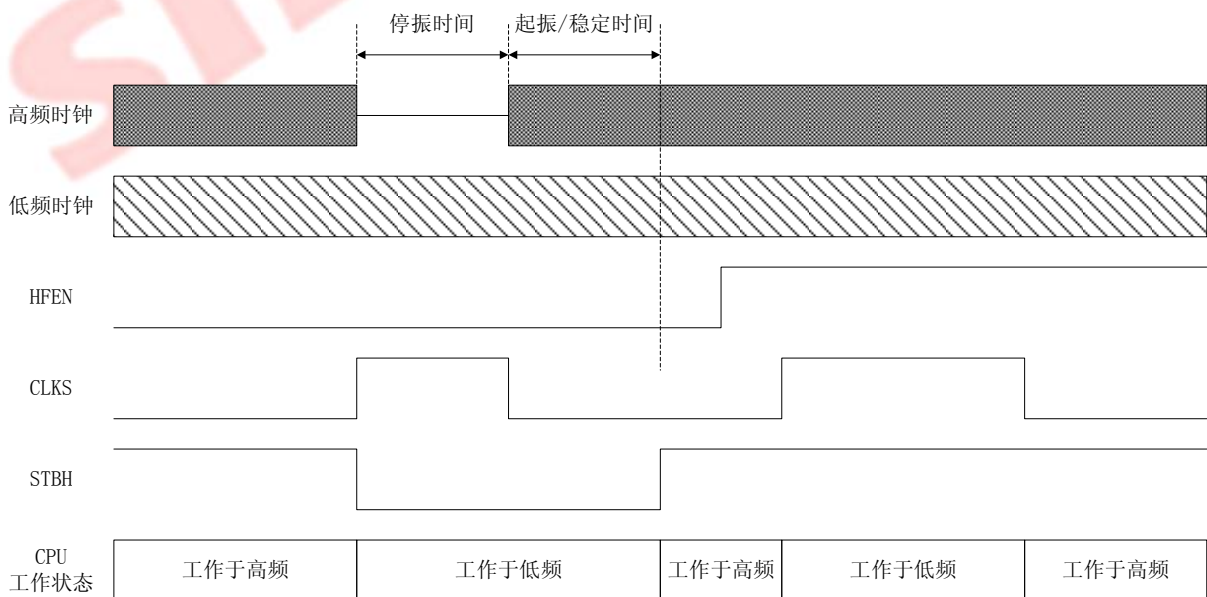
工作模式	切入条件	系统状态
高速	任意模式下, 系统复位	CPU 高速运行, 高/低频时钟源均工作
	低速模式下, CLKS 清 0	
	HOLD1/HOLD2/休眠模式下, CPU 唤醒	
低速	高速模式下, CLKS 置 1	CPU 低速运行, 高频时钟源由 HFEN 决定
	HOLD1/HOLD2/休眠模式下, CPU 唤醒	
HOLD1	高/低速模式下, HFEN 置 1, 执行 STOP 指令	CPU 暂停, 高频时钟源工作, 低频时钟源由 LFEN 决定
HOLD2	高/低速模式下, HFEN 清 0, LFEN 置 1, 执行 STOP 指令	CPU 暂停, 高频时钟源停止, 低频时钟源工作
休眠	高/低速模式下, HFEN 清 0, LFEN 清 0, 执行 STOP 指令	CPU 暂停, 高/低频时钟源均停止

注：WDT 时钟源为 LIRC，WDT 开启时 LIRC 将一直工作而不受系统工作模式影响。

#### 工作模式切换示意图



#### 高低频时钟切换时序图



**工作模式寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OSCMR</b>	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
<b>R/W</b>	-	-	R	R	-	R/W	R/W	R/W
<b>初始值</b>	-	-	X	1	-	0	0	0

- BIT[5]      **STBL** – 低频时钟源振荡状态标志位  
 0: 低频时钟源停振或未稳定;  
 1: 低频时钟源已稳定振荡;
- BIT[4]      **STBH** – 高频时钟源振荡状态标志位  
 0: 高频时钟源停振或未稳定;  
 1: 高频时钟源已稳定振荡;
- BIT[2]      **CLKS** – CPU 时钟源选择位  
 0: 系统高频时钟作为 CPU 时钟源;  
 1: 系统低频时钟作为 CPU 时钟源;
- BIT[1]      **LFEN** – 低频时钟源使能位  
 0: 在休眠/HOLD 模式下, 低频时钟源停止工作;  
 1: 低频时钟源始终工作;
- BIT[0]      **HFEN** – 高频时钟源使能位  
 0: 在低速/休眠/HOLD 模式下, 高频时钟源停止工作;  
 1: 高频时钟源始终工作;

#### 4.4 低功耗模式

芯片的高速模式、低速模式为运行模式, 而休眠模式、HOLD1 模式、HOLD2 模式则为低功耗模式。

执行 STOP 指令可使系统进入低功耗模式, 同时对系统会产生以下影响:

- ◇ CPU 停止运行;
- ◇ 根据不同模式停止相应时钟源的振荡;
- ◇ RAM 内容保持不变;
- ◇ 所有的输入输出端口保持原态不变;
- ◇ 定时器若其时钟源未停止, 则可以保持继续工作;

以下情况可使系统退出低功耗模式:

- ◇ 上电复位;
- ◇ 外部复位 (若有外部复位功能);
- ◇ 有 WDT 溢出 (若低功耗模式下 WDT 保持继续工作);
- ◇ 有外部中断请求发生 (若有外部中断功能);

- ◇ 定时器溢出中断发生（若低功耗模式下定时器保持继续工作）；
- ◇ 有键盘中断请求发生（若有键盘中断功能）；

**注：低功耗模式下触发中断请求时，若对应的中断使能位关闭，则不会退出低功耗模式；若对应的中断使能位开启而中断总使能位关闭，则仅唤醒 CPU 执行下一条指令；若对应的中断使能位和中断总使能位均开启，则唤醒 CPU 后执行中断服务程序。**

## 5 复位

### 5.1 复位条件

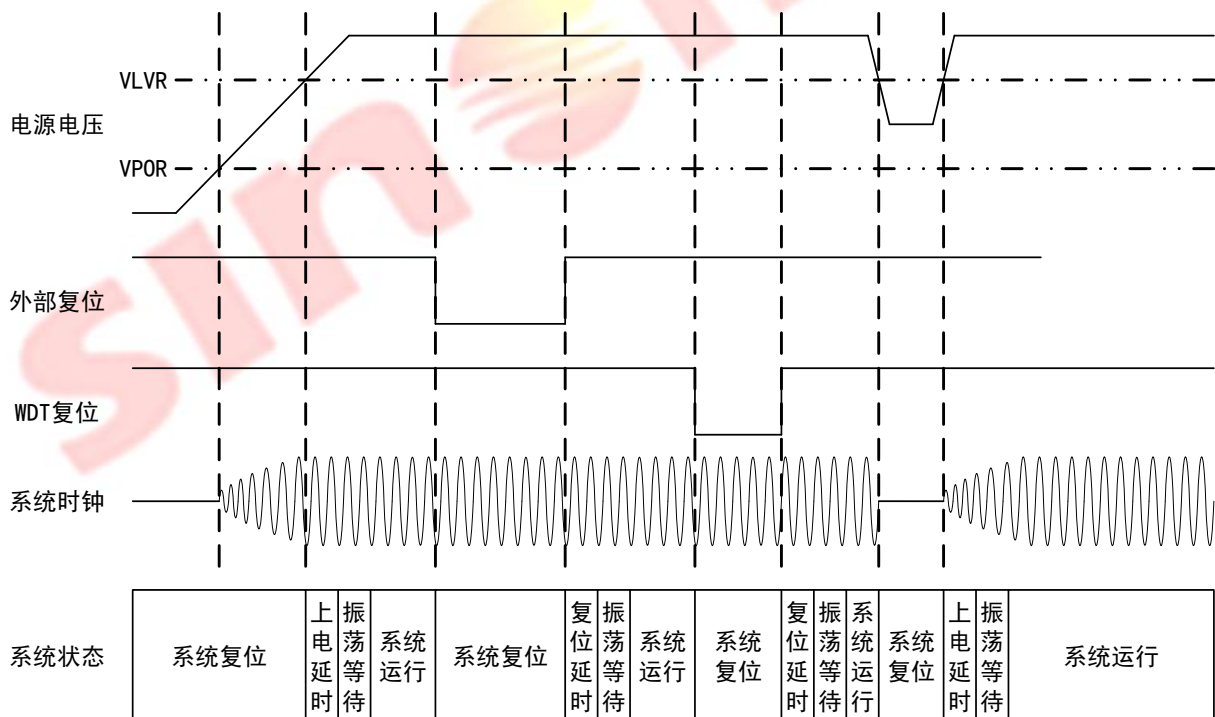
芯片共有如下几种复位方式：

- ◇ 上电复位 POR；
- ◇ 外部复位；
- ◇ 低电压复位 LVR；
- ◇ WDT 看门狗复位；

任何一种复位发生时，系统均会重新从 0000H 地址处开始执行指令，且系统还会将特殊功能寄存器 SFR 重置为复位初始值。

上电复位 POR 和低电压复位 LVR 会关闭系统主时钟振荡器，复位解除后才重新打开振荡器，因为振荡器起振和稳定需要一定的时间，所以系统会保持一定时间的上电延时和振荡等待后才开始工作；而外部复位和 WDT 复位不会关闭主时钟振荡器，复位解除时系统会在较短的复位延时和振荡等待后即开始工作。

下图是复位产生和系统工作状态之间时序关系的示意图：



## 5.2 上电复位

芯片的上电复位电路可以适应快速、慢速上电的情况，且当芯片上电过程中出现电源电压抖动时均能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤：

- (1) 检测系统工作电压，等待电压高于上电复位电压  $V_{POR}$  并保持稳定；
- (2) 若有 LVR 功能，则需等待电压高于  $V_{LVR}$  并保持稳定；
- (3) 若有外部复位功能，则需等待复位引脚电压高于  $V_{ih}$ ；
- (4) 初始化所有寄存器；
- (5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
- (6) 上电结束，系统开始执行指令。

## 5.3 外部复位

芯片的外部复位功能可通过用户配置字 RSTEN 开启，引脚设为外部复位脚即开启外部复位功能，端口的内部上拉电阻自动有效。外部复位输入端口 RST 为施密特结构，低电平有效。当端口输入为高电平时，系统正常运行；输入为低电平时，系统复位。

## 5.4 低电压复位

芯片的低电压复位电压可通过用户配置字 VLVR 选择。电压检测电路有一定的回滞特性，通常回滞电压为 0.1V 左右，当电源电压下降至 LVR 电压时 LVR 复位有效，反之则电源电压需上升至 LVR 电压+0.1V 后 LVR 复位才解除。

## 5.5 看门狗复位

看门狗 (WDT) 复位是一种对程序正常运行的保护机制。正常情况下，用户程序需定时对 WDT 定时器进行清零操作，以保证 WDT 不溢出。若出现异常情况，程序未按时对 WDT 定时器清零，则芯片会因 WDT 溢出而产生看门狗复位，系统重新初始化，返回受控状态。

**注：低功耗模式下 CPU 暂停工作，若此时有 WDT 溢出，则仅唤醒 CPU 而不产生复位。**

## 6 I/O 端口

### 6.1 通用 I/O 功能

芯片的输入/输出端口包括一组 8 位端口 P0 和一组 2 位端口 P1。所有 I/O 端口均支持施密特输入，除 P11 外的 I/O 端口均支持推挽输出。

除用作通用数字 I/O 端口外，部分端口还具有外部中断输入、PWM 输出、或 ADC 模拟输入等复用功能。

#### 端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP0</b>	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0] **P0nD** – P0n 端口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IOP1</b>	-	-	-	-	-	-	P11D	P10D
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	X	X

BIT[1:0] **P1nD** – P1n 端口数据位 (n=1-0)

#### 端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEP0</b>	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **P0nOE** – P0n 端口输出使能位 (n=7-0)

- 0: 端口作为输入口，读端口操作将读取端口的输入电平状态；
- 1: 端口作为输出口，读端口操作将读取端口的数据寄存器值；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OEP1</b>	-	-	-	-	-	-	P11OE	P10OE
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

BIT[1:0] **P1nOE** – P1n 端口输出使能位 (n=1-0)

- 0: 端口作为输入口，读端口操作将读取端口的输入电平状态；
- 1: 端口作为输出口，读端口操作将读取端口的数据寄存器值；



## 6.2 内部上拉电阻

所有端口均有内部上拉电阻，且均有独立的寄存器位控制其上拉电阻在端口处于输入状态时是否有效。端口处于输出状态时，上拉电阻及其控制位无效。

### 上拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP0</b>	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPU** – P0n 端口上拉电阻控制位 (n=7-0)

0: 端口内部上拉电阻无效;

1: 端口内部上拉电阻有效;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP1</b>	-	-	-	-	-	-	P11PU	P10PU
<b>R/W</b>	-	-	-	-	-	-	R/W	R/W
<b>初始值</b>	-	-	-	-	-	-	0	0

BIT[1:0] **P1nPU** – P1n 端口上拉电阻控制位 (n=1-0)

0: 端口内部上拉电阻无效;

1: 端口内部上拉电阻有效;

### 上拉电阻选择寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>RUPSEL</b>	IICRS	-	-	-	-	-	-	-
<b>R/W</b>	R/W	-	-	-	-	-	-	-
<b>初始值</b>	0	-	-	-	-	-	-	-

BIT[7] **IICRS** – SCL/SDA 端口上拉电阻选择位

0: SCL/SDA 对应端口上拉电阻为 60 KΩ/150 KΩ;

1: SCL/SDA 对应端口上拉电阻为 4.7KΩ;

**注：该位仅用于选择 SCL/SDA 所对应端口的内部上拉电阻阻值，并不影响端口的工作模式，即该位为 1 时端口也可用作通用 I/O。**

## 7 定时器 TIMER

### 7.1 看门狗定时器 WDT

看门狗定时器 WDT 的时钟源为内部低频 RC 振荡器，WDT 计数器溢出将复位芯片或唤醒 CPU。

可通过用户配置字 WDTM 设置 WDT 工作模式。若选择始终开启，则 WDT 一直工作，高速/低速模式下 WDT 溢出将复位芯片，休眠/HOLD 模式下 WDT 溢出将唤醒 CPU；若选择低功耗模式下关闭，则 WDT 在休眠/HOLD 模式下自动停止、在其他方式唤醒 CPU 后恢复工作。

执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器。

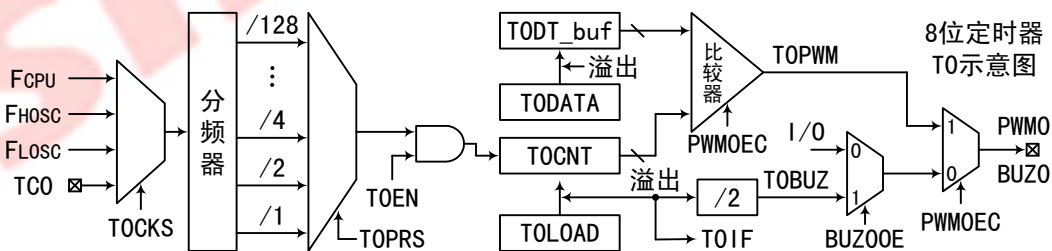
WDT 溢出时间可配置为 16ms/64ms/256ms/1024ms/2048ms/4096ms。

**注：WDT 溢出时间为典型值，实际值偏差大，必须保证清 WDT 时间小于典型值的 1/4。**

### 7.2 定时器 T0

定时器 T0 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器及比较寄存器。

- ◇ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ◇ 支持 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ◇ 支持 BUZ 输出；
- ◇ 支持溢出中断和溢出唤醒功能；



定时器 T0，可通过寄存器位 T0CKS 选择时钟源，通过 T0PRS 选择预分频比，所选时钟源通过预分频器后产生 T0 计数器 TOCNT 的计数时钟。预分频比可选择 1~128 分频，对 TOCNT 的写操作将清零预分频计数器，而预分频比保持不变。

当 TOEN=0 时，TOCNT 保持不变，写重载寄存器 TOLOAD 将立即载入 TOCNT 中；当 TOEN=1 时，TOCNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 TOIF 将被置 1，同时 T0 自动将当前 TOLOAD 值载入 TOCNT 中重新开始计数。

如图所示，定时器 T0 可实现 BUZ 功能，当 BUZ0OE=1 时，端口输出蜂鸣器驱动信号，频率为 T0 溢出频率的 2 分频。

如图所示，定时器 T0 可实现 PWM 功能，PWM0EC 置 1 将使能 PWM0 功能并允许端口输出 PWM 波形。每个 PWM 周期内，计数器 T0CNT 从重载值开始递减计数：当计数到与比较寄存器 T0DATA 相等时，TOPWM 信号变为高电平；当计数溢出时，TOPWM 信号变为低电平。

T0DATA 配有 1 个 8 位的比较缓冲器 (T0DT\_buf) 用于与 T0CNT 比较，PWM0 关闭时写 T0DATA 将立即载入比较缓冲器中，而 PWM0 使能后写 T0DATA 则将在 T0 溢出时才载入比较缓冲器中。若要首个 PWM 周期和占空比准确，需先写重载寄存器和比较寄存器，再开启定时器和 PWM 功能。

T0PWM 信号的占空比计算如下：

- ◇ 高电平时间 = (T0DATA) × T0CNT 计数时钟周期
- ◇ 周期 (T0 的溢出周期) = (T0LOAD+1) × T0CNT 计数时钟周期
- ◇ 占空比 = 高电平时间 / 周期 = (T0DATA) / (T0LOAD+1)

### 定时器 T0 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0CR</b>	TOEN	PWM0EC	BUZ0OE	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **TOEN** – 定时器 T0 使能位

- 0: 关闭定时器 T0;
- 1: 开启定时器 T0;

BIT[6] **PWM0EC** – PWM0 使能位及输出控制位

- 0: 关闭 PWM0 功能，并禁止端口输出 PWM 波形;
- 1: 使能 PWM0 功能，并允许端口输出 PWM 波形;

BIT[5] **BUZ0OE** – BUZ0 端口输出使能位

- 0: 禁止端口输出 BUZ 波形;
- 1: 允许端口输出 BUZ 波形 (仅 PWM0EC=0 时有效);

BIT[4:3] **T0CKS[1:0]** – T0 时钟源选择位

T0CKS[1:0]	T0 时钟源
00	F <sub>CPU</sub>
01	F <sub>HOSC</sub>
10	F <sub>LOSC</sub>
11	TC0 上升沿

BIT[2:0] **T0PRS[2:0]** – T0 预分频比选择位

T0PRS[2:0]	T0 时钟预分频比
000	1 : 1
001	1 : 2

010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

### 定时器 T0 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0CNT</b>	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T0CNT[7:0]** – T0 计数器，为可读写的递减计数器

### 定时器 T0 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0LOAD</b>	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T0LOAD[7:0]** – T0 重载寄存器，用于设置 T0 的计数周期

**注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。**

### 定时器 T0 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0DATA</b>	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

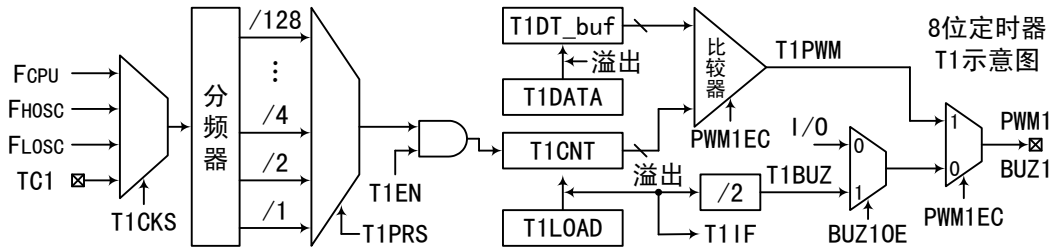
BIT[7:0] **T0DATA[7:0]** – T0 比较寄存器，用于设置 PWM0 的占空比

## 7.3 定时器 T1

定时器 T1 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器及比较寄存器。

- ◇ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ◇ 支持 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ◇ 支持 BUZ 输出；
- ◇ 支持溢出中断和溢出唤醒功能；

定时器 T1，其定时、外部计数、BUZ 和 PWM 功能与 T0 完全相同。



### 定时器 T1 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1CR</b>	T1EN	PWM1EC	BUZ1OE	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **T1EN** – 定时器 T1 使能位

- 0: 关闭定时器 T1;
- 1: 开启定时器 T1;

BIT[6] **PWM1EC** – PWM1 使能位及输出控制位

- 0: 关闭 PWM1 功能，并禁止端口输出 PWM 波形;
- 1: 使能 PWM1 功能，并允许端口输出 PWM 波形;

BIT[5] **BUZ1OE** – BUZ1 端口输出使能位

- 0: 禁止端口输出 BUZ 波形;
- 1: 允许端口输出 BUZ 波形（仅 PWM1EC=0 时有效）;

BIT[4:3] **T1CKS[1:0]** – T1 时钟源选择位

T1CKS[1:0]	T1 时钟源
00	F <sub>CPU</sub>
01	F <sub>HOSC</sub>
10	F <sub>LOSC</sub>
11	TC1 上升沿

BIT[2:0] **T1PRS[2:0]** – T1 预分频比选择位

T1PRS[2:0]	T1 时钟预分频比
000	1 : 1
001	1 : 2
010	1 : 4
011	1 : 8
100	1 : 16
101	1 : 32
110	1 : 64
111	1 : 128

**定时器 T1 计数器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1CNT</b>	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T1CNT[7:0]** – T1 计数器，为可读写的递减计数器

**定时器 T1 重载寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1LOAD</b>	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	1	1	1	1	1	1	1	1

BIT[7:0] **T1LOAD[7:0]** – T1 重载寄存器，用于设置 T1 的计数周期

**注：定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作。**

**定时器 T1 比较寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T1DATA</b>	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[7:0]** – T1 比较寄存器，用于设置 PWM1 的占空比

## 8 模数转换器 ADC

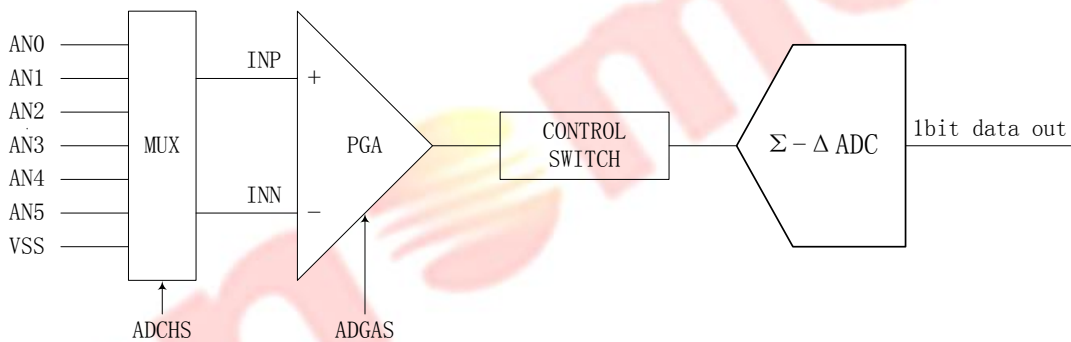
### 8.1 ADC 概述

芯片内置 1 个 24 位高精度差分输入  $\Sigma-\Delta$  型的模数转换器 ADC。

- ◇ 3 对外部差分输入通道：AN0/AN1、AN2/AN3、AN4/AN5，或 6 路负端内接 GND 的单输入通道：AN0~AN5
- ◇ 内置可编程增益放大器 PGA：放大倍数 1/2/4/8/16/32/64/128/256/512 倍可选
- ◇ 工作电压可选：VDD、内部 LDO 电压  $V_{LDO}$  (2.5V/2.8V/3.0V)
- ◇ ADC 共模电压可通过端口 ACM 输出

ADC 模块可通过寄存器位 ADEN 开启，通过 ADCKS 选择转换时钟，通过 ADCHS 选择转换的模拟通道，通过 ADSPS 选择降采样率，并可通过 ADGAS 选择 PGA 放大倍数；转换完成后结果存入 ADRH/ADRM/ADRL 中，同时中断标志 ADIF 置 1 触发 ADC 中断。

#### 系统时钟示意图



### 8.2 ADC 相关寄存器

#### ADC 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADCRO</b>	ADEN	CHOPEN	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7] **ADEN** – ADC 使能位

- 0: 关闭 ADC;
- 1: 开启 ADC;

BIT[6] **CHOPEN** – BANDGAP-CHOPCLK 使能位

- 0: 关闭 CHOP;
- 1: 开启 CHOP (仅 LFEN=1 时有效);

BIT[5:4] **ADCKS[1:0]** – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 F <sub>ADC</sub>
00	F <sub>HIRC</sub> /4
01	F <sub>HIRC</sub> /8
10	F <sub>HIRC</sub> /16
11	F <sub>HIRC</sub> /32

BIT[3:0] **ADCHS[3:0]** – ADC 输入通道选择位

ADCHS[3:0]	ADC 输入通道	
	PGA 正端 ( INP )	PGA 负端 ( INN )
0000	AN1	AN0
0001	AN0	AN1
0010	AN1	VSS
0011	AN0	VSS
0100	AN3	AN2
0101	AN2	AN3
0110	AN3	VSS
0111	AN2	VSS
1000	AN5	AN4
1001	AN4	AN5
1010	AN5	VSS
1011	AN4	VSS
1100	保留	保留
1101	保留	保留
1110	保留	保留
1111	保留	保留

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADCR1</b>	ADSPS3	ADSPS2	ADSPS1	ADSPS0	ADGAS3	ADGAS2	ADGAS1	ADGAS0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:4] **ADSPS[3:0]** – ADC 降采样率选择位

ADSPS[3:0]	ADC 降采样率
0000	512
0001	1024
0010	2048
0011	4096
0100	8192
0101	16384
0110	32768
0111	65536
1000~1111	256



BIT[3:0] ADGAS[3:0] – PGA 放大倍数选择位

ADGAS[3:0]	PGA 放大倍数
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001~1111	512

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR2	-	AD0EN	AD1EN	AD2EN	VLDOS0	VLDOS1	LDOEN	ADVRS
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	0	0	0	0	0	0	0

BIT[6] AD0EN – 第 0 组差分输入模拟通道使能位

- 0: 端口用于其他功能;
- 1: 端口复用为模拟输入 AN0/AN1;

BIT[5] AD1EN – 第 1 组差分输入模拟通道使能位

- 0: 端口用于其他功能;
- 1: 端口复用为模拟输入 AN2/AN3;

BIT[4] AD2EN – 第 2 组差分输入模拟通道使能位

- 0: 端口用于其他功能;
- 1: 端口复用为模拟输入 AN4/AN5;

注:

1. 每组差分输入模拟通道的 2 个端口只能同时使能/禁止用作模拟输入;
2. 端口用作模拟输入时, 数字输入功能自动关闭, 但数字输出功能需通过设置端口方向寄存器来关闭;

BIT[3:2] VLDOS[1:0] – 内部 LDO 输出电压选择位

VLDOS[1:0]	LDO 输出电压 (V)
00	禁用
01	2.5
10	2.8
11	3.0

BIT[1] LDOEN – LDO 使能位

- 0: 关闭 LDO;
- 1: 开启 LDO;

BIT[0]      **ADVRS** – ADC 工作电压选择位

0: ADC 工作电压为内部 LDO 电压  $V_{LDO}$  (仅 LDOEN=1 时有效);

1: ADC 工作电压为 VDD;

**注: 若 LDO 引脚外部无对地电容或未封出引脚, 则 LDO 电压不稳定, 此时 ADC 工作电压必须选择 VDD。**

### ADC 转换结果寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRH</b>	ADR23	ADR22	ADR21	ADR20	ADR19	ADR18	ADR17	ADR16
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0]      **ADR[23:16]** – ADC 转换结果高 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRM</b>	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR9	ADR8
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0]      **ADR[15:8]** – ADC 转换结果中 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>ADRL</b>	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
<b>R/W</b>	R	R	R	R	R	R	R	R
<b>初始值</b>	X	X	X	X	X	X	X	X

BIT[7:0]      **ADR[7:0]** – ADC 转换结果低 8 位

## 8.3 ADC 操作步骤

模数转换操作步骤:

- (1) 设置相应端口为输入端口, 关闭上/下拉电阻;
- (2) 设置 ADC 相关参数;
- (3) ADEN 置 1, 使能 ADC 模块;
- (4) 设置 ADCHS, 选择 ADC 转换通道;
- (5) 等待 ADIF 硬件置 1 (或利用 ADC 中断);
- (6) 读取 ADC 转换结果 (ADRH、ADRM、ADRL);
- (7) 重复对不同的通道进行转换或对同一通道进行多次转换;

## 9 低电压检测 LVD

芯片内置低电压检测模块 LVD，可通过寄存器位 LVDEN 开启，通过 VLVD3 选择电压比较值。当 VDD 电压低于比较电压时检测状态标志位 LVDF 将被置 1，反之 LVDF 将被清 0。

### LVD 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>LVDCR</b>	LVDEN	VLVD3	VLVD2	VLVD1	VLVD0	-	-	LVDF
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	-	-	R
<b>初始值</b>	0	0	0	0	0	-	-	0

BIT[7] **LVDEN** – LVD 使能位

0: 关闭 LVD;

1: 开启 LVD;

BIT[6:3] **VLVD3[3:0]** – LVD 比较电压选择位

VLVD3[3:0]	LVD 比较电压
0000	保留
0001	2.0V
0010	2.1V
0011	2.2V
0100	2.3V
0101	2.4V
0110	2.5V
0111	2.6V
1000	2.7V
1001	2.8V
1010	2.9V
1011	3.0V
1100	3.1V
1101	3.2V
1110	3.3V
1111	3.4V

BIT[0] **LVDF** – LVD 检测状态标志位

0: VDD 电压高于比较电压，或 LVD 关闭;

1: VDD 电压低于比较电压;

## 10 IIC 总线从机接口

### 10.1 IIC 概述

芯片内置 IIC 通讯模块，支持 IIC 总线从机接口。IIC 总线是双向两线结构，数据线 SDA 和时钟线 SCL 与 IO 端口复用，当 IIC 模块使能时，端口用作 SDA/SCL，此时为开漏输出，可选择内置或外接合适的上拉电阻，以匹配选定的通讯速率。

芯片 IIC 通讯模块仅支持 IIC 总线 7 位地址从机模式，总线时钟信号始终由 SCL 端口输入。当从总线上检测到 IIC 起始信号后开始接收主机发送的 7 位地址编码信息与 1 位读/写命令信息，若与本机预设地址匹配，则发送应答 ACK 以响应主机，并根据主机命令准备接收或发送数据，完成后检测到总线上的停止信号后结束通讯。

### 10.2 IIC 相关寄存器

#### IIC 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IICCR</b>	IICEOF	IICAAS	IICBUSY	IICMOD	SACK	MSTCMD	IICEN	RACK
<b>R/W</b>	R	R	R	R/W	R/W	R	R/W	R
<b>初始值</b>	0	0	0	0	0	0	0	1

BIT[7] **IICEOF** – IIC 通讯一帧数据收发结束标志位

- 0: 收发数据中；
- 1: 一帧数据收发结束；

BIT[6] **IICAAS** – IIC 地址匹配标志位

- 0: 地址不匹配；
- 1: 地址匹配；

BIT[5] **IICBUSY** – IIC 总线状态标志位

- 0: IIC 总线空闲；
- 1: IIC 总线被占用；

BIT[4] **IICMOD** – IIC 通讯发送/接收模式控制位

- 0: 接收模式，接收主机发送的数据；
- 1: 发送模式，向主机发送数据；

BIT[3] **SACK** – 发送应答信号控制位（仅 IICMOD=0 时有效）

- 0: 接收模式下收到数据后应答；
- 1: 接收模式下收到数据后不应答；

- BIT[2] **MSTCMD** – 主机发送的命令信息标志位  
 0: 主机发送写命令, 从机将接收数据;  
 1: 主机发送读命令, 从机需发送数据;
- BIT[1] **IICEN** – IIC 接口使能位  
 0: 关闭 IIC 接口;  
 1: 开启 IIC 接口;
- BIT[0] **RACK** – 接收到的应答信号标志位 (仅 IICMOD=1 时有效)  
 0: 向主机发送数据后收到应答;  
 1: 向主机发送数据后未收到应答;

### IIC 地址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IICAR</b>	IICAD6	IICAD5	IICAD4	IICAD3	IICAD2	IICAD1	IICAD0	-
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
<b>初始值</b>	0	0	0	0	0	0	0	-

BIT[7:1] **IICAD[6:0]** – 预设的 7 位从机地址信息, 用作本机在总线中的地址编码

### IIC 数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IICDR</b>	IICDT7	IICDT6	IICDT5	IICDT4	IICDT3	IICDT2	IICDT1	IICDT0
<b>R/W</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	0	0	0	0	0	0	0	0

BIT[7:0] **IICDT[7:0]** – IIC 接口接收到的 8 位数据, 或准备发送的 8 位数据, 通讯时高位在前

## 10.3 IIC 通讯流程

IIC 控制寄存器中, IICEN 位控制 IIC 模块的关闭/使能, IICMOD 位设置 IIC 模块当前工作在发送模式还是接收模式。

接收模式下, 通过 SACK 位设置应答信号, 当接收到主机发送的 8 位数据后, 第 9 个时钟 IIC 模块将 SACK 位内容发送至 IIC 总线, 若需继续接收下一个数据, 则在接收数据前 SACK 位需清 0。

IICEOF 位在开始传送数据时被清 0, 在一帧数据收发结束后被置 1。

IICAAS 位当从器件地址匹配时被置 1, 同时 IIC 中断标志 (IICIF) 被置 1; 如果地址不匹配, IICAAS 被清 0。

IICBUSY=1 时表示 IIC 总线忙, 即检测到 START 信号; HBB=0 表示 IIC 总线空闲, 即检测到“STOP”信号, 此时 IIC 总线空闲。

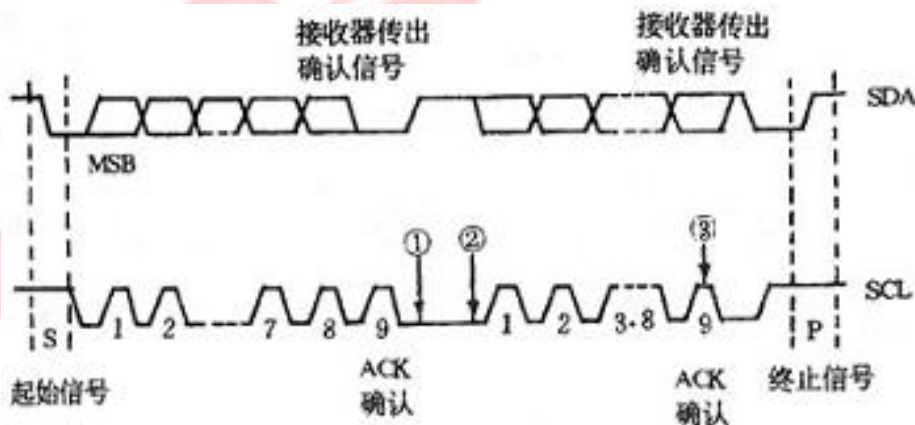
MSTCMD 表示当地址匹配时, 主机同时发送过来的读/写命令。当 IICAAS 被置 1 后, 可通过 MSTCMD 位来确定芯片是准备发送数据还是接收数据。当 MSTCMD=1, 表示主机将从 IIC 总线读数据,

从机需将数据写到 IIC 总线上，即从机为发送模式；当 MSTCMD=0，表示主机将写数据到 IIC 总线，从机需从总线读取数据，即从机为接收模式。

RACK=0，表示接收到一个应答信号，在发送模式，从机通过检测 RACK 以确定主机是否要接收下一个数据。从机将一直写数据到 IIC 总线直到 RACK=1，同时将释放 SDA 线，从而主机可以发送 STOP 信号来释放总线。

IICAR 寄存器的第 7~1 位定义从机地址，开始数据传送时，主机通过发送从机地址来指定通讯对象。一旦 IIC 总线上出现起始信号，则所有的从机都会接收到连续的 8 位数，该数据的前 7 位是从机地址，高位在前，低位在后。如果从机地址匹配，系统会置位 IICAAS，同产生 IIC 中断。进入中断服务程序后，系统要检测 IICAAS 位，以确定 IIC 总线中断是因从机地址匹配而产生的，还是来自 8 位数据传送完毕而产生的。8 位数据的最后一位是主机发送的读/写控制位，该位会反映到 MSTCMD。从机通过检测 MSTCMD 位以确定主机是要发送数据还是接收数据，并确定本机是作发送器还是接收器。

- (1) 将本机分配的 IIC 总线地址编码预先写入 IICAR 中；
- (2) IICEN 置 1，使能 IIC 模块；
- (3) IICIE 置 1，允许 IIC 中断；
- (4) 当从机地址匹配且 SACK=0 时，图中①处 IICAAS 将置 1 并产生中断，且会将 SCL 线拉低；
- (5) 通过检测 MSTCMD 来确定本机是准备发送数据还是接收数据。若将发送数据，则将 IICMOD 置 1，再将数据写入 IICDR 中；若将接收数据，则将 IICMOD 清 0，并设置 SACK 位以决定下一帧是否应答，如图中③处所示；
- (6) IICIF 清 0，并释放 SCL 线，如图中②处所示；
- (7) 当下一帧第 9 个时钟结束时，IICEOF=1 且产生中断。如果芯片处于接收模式且 SACK 为 0，或芯片处于发送模式且 RACK=0，将拉低时钟线 SCL；否则芯片会释放时钟线 SCL 和数据线 SDA 等待主机发 STOP 命令。



## 11 中断

芯片的中断源包括外部中断（INT0~INT1）、定时器中断（T0~T1）、ADC 中断和 IIC 中断等。可通过中断总使能位 GIE 屏蔽所有中断。

CPU 响应中断的过程如下：

- ◇ CPU 响应中断源触发的中断请求时，自动将当前指令的下一条要执行指令的地址压栈保存，自动清 0 中断总使能位 GIE 以暂停响应后续中断。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ◇ CPU 响应中断后，程序跳到中断入口地址（0008H）开始执行中断服务程序，中断服务程序应先保存累加器 A 和状态寄存器 PFLAG，然后处理被触发的中断。
- ◇ 中断服务程序处理完中断后，应先恢复累加器 A 和状态寄存器 PFLAG，然后执行 RETIE 返回主程序。此时芯片将自动恢复 GIE 为 1，然后从堆栈取出 PC 值，从中断产生时当前指令的下一条指令继续执行。

### 11.1 外部中断

芯片有 2 路外部中断源 INT0/INT1，可选择上升沿、下降沿或电平变化等触发方式。外部中断触发时，中断标志 INTnIF（n=0-1）将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位 INTnIE（n=0-1）为 1，则产生外部中断。

**注：要使用外部中断功能，需将相应端口设为输入状态。**

### 11.2 定时器中断

定时器 Tn（n=0-1）在计数溢出时触发定时器中断，中断标志 TnIF（n=0-1）将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位 TnIE（n=0-1）为 1，则产生定时器中断。

### 11.3 ADC 中断

ADC 转换完成后触发 ADC 中断，中断标志 ADIF 将被置 1，若中断总使能位 GIE 为 1 且 ADC 中断使能位 ADIE 为 1，则产生 ADC 中断。

## 11.4 IIC 中断

IIC 通讯时，当器件地址匹配、数据接收或发送完成时触发 IIC 中断，中断标志 IICIF 将被置 1，若中断总使能位 GIE 为 1 且 IIC 中断使能位 IICIE 为 1，则产生 IIC 中断。

## 11.5 中断相关寄存器

### 中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INTE</b>	-	ADIE	IICIE	-	INT1IE	INT0IE	T1IE	TOIE
<b>R/W</b>	-	R/W	R/W	-	R/W	R/W	R/W	R/W
<b>初始值</b>	-	0	0	-	0	0	0	0

BIT[6] **ADIE** – ADC 中断使能位

- 0: 屏蔽 ADC 中断;
- 1: 使能 ADC 中断;

BIT[5] **IICIE** – IIC 中断使能位

- 0: 屏蔽 IIC 中断;
- 1: 使能 IIC 中断;

BIT[3] **INT1IE** – INT1 中断使能位

- 0: 屏蔽 INT1 中断;
- 1: 使能 INT1 中断;

BIT[2] **INT0IE** – INT0 中断使能位

- 0: 屏蔽 INT0 中断;
- 1: 使能 INT0 中断;

BIT[1] **T1IE** – 定时器 T1 中断使能位

- 0: 屏蔽定时器 T1 中断;
- 1: 使能定时器 T1 中断;

BIT[0] **TOIE** – 定时器 T0 中断使能位

- 0: 屏蔽定时器 T0 中断;
- 1: 使能定时器 T0 中断;

### 中断标志寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>INTF</b>	-	ADIF	IICIF	-	INT1IF	INT0IF	T1IF	TOIF
<b>R/W</b>	-	R/W	R/W	-	R/W	R/W	R/W	R/W
<b>初始值</b>	-	0	0	-	0	0	0	0



- 
- BIT[6]      **ADIF** – ADC 中断标志位  
0: 未触发 ADC 中断;  
1: 已触发 ADC 中断, 需软件清零;
- BIT[5]      **IICIF** – IIC 中断标志位  
0: 未触发 IIC 中断;  
1: 已触发 IIC 中断, 需软件清零;
- BIT[3]      **INT1IF** – INT1 中断标志位  
0: 未触发 INT1 中断;  
1: 已触发 INT1 中断, 需软件清零;
- BIT[2]      **INT0IF** – INT0 中断标志位  
0: 未触发 INT0 中断;  
1: 已触发 INT0 中断, 需软件清零;
- BIT[1]      **T1IF** – 定时器 T1 中断标志位  
0: 未触发定时器 T1 中断;  
1: 已触发定时器 T1 中断, 需软件清零;
- BIT[0]      **T0IF** – 定时器 T0 中断标志位  
0: 未触发定时器 T0 中断;  
1: 已触发定时器 T0 中断, 需软件清零;

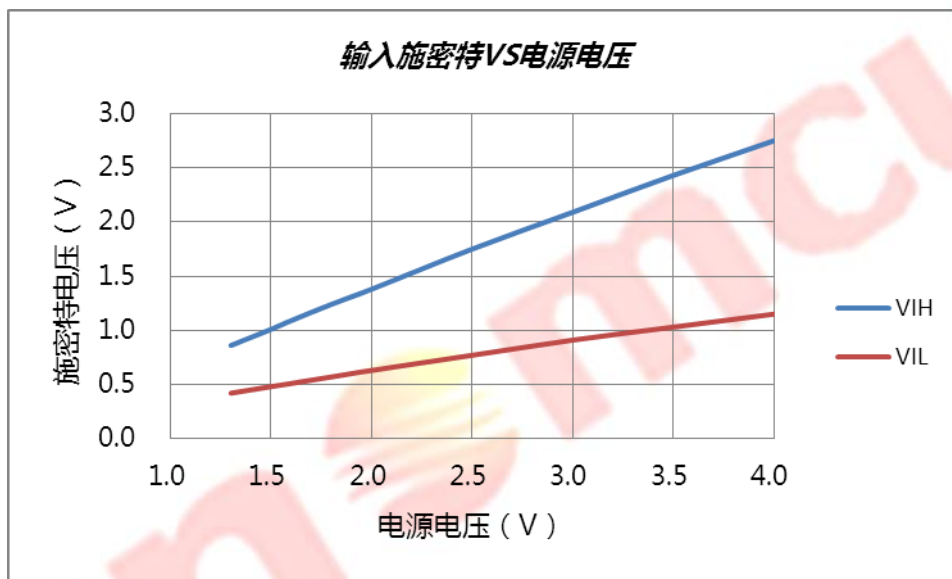
## 12 特性曲线

注：

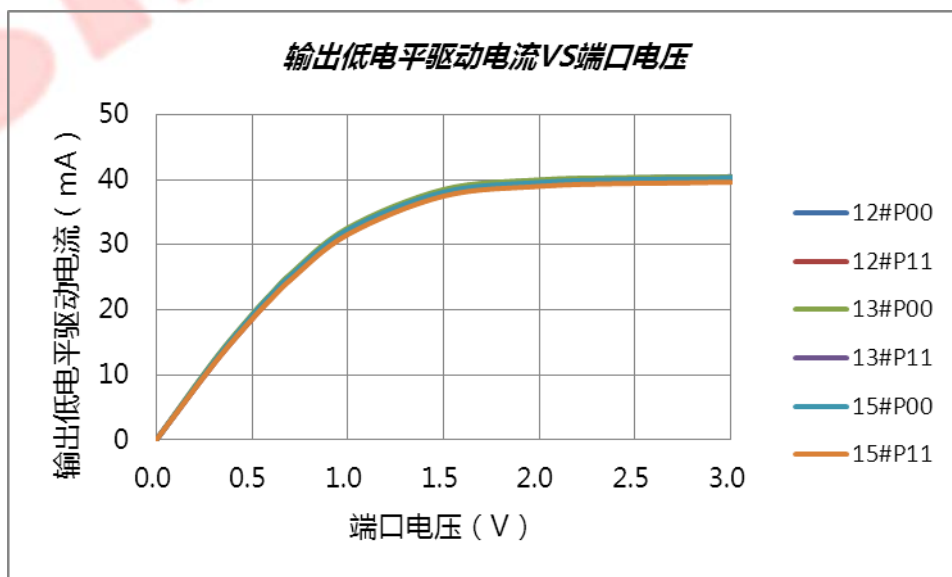
1. 特性曲线图中数据均来自抽样实测，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明；
2. 若图文中无特别说明，则电压特性曲线的温度条件为  $T=25^{\circ}\text{C}$ ，温度特性曲线的电压条件为  $VDD=3\text{V}$ ；

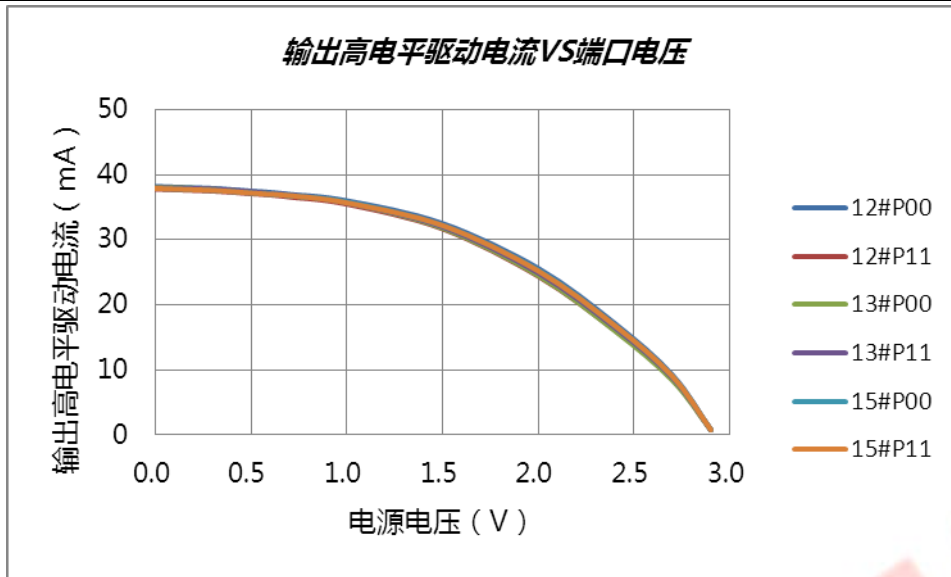
### 12.1 I/O 特性

#### 输入 SMT 阈值电压 VS 电源电压

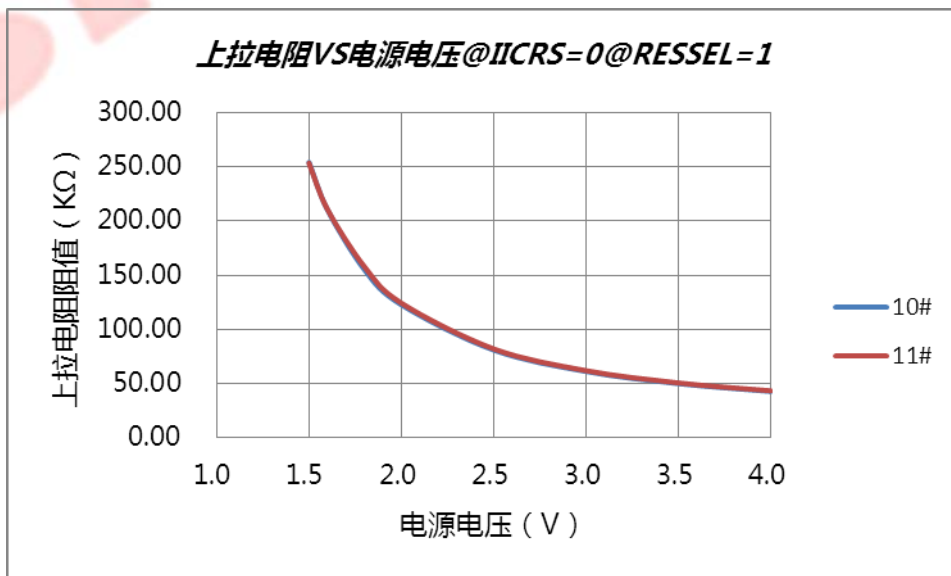
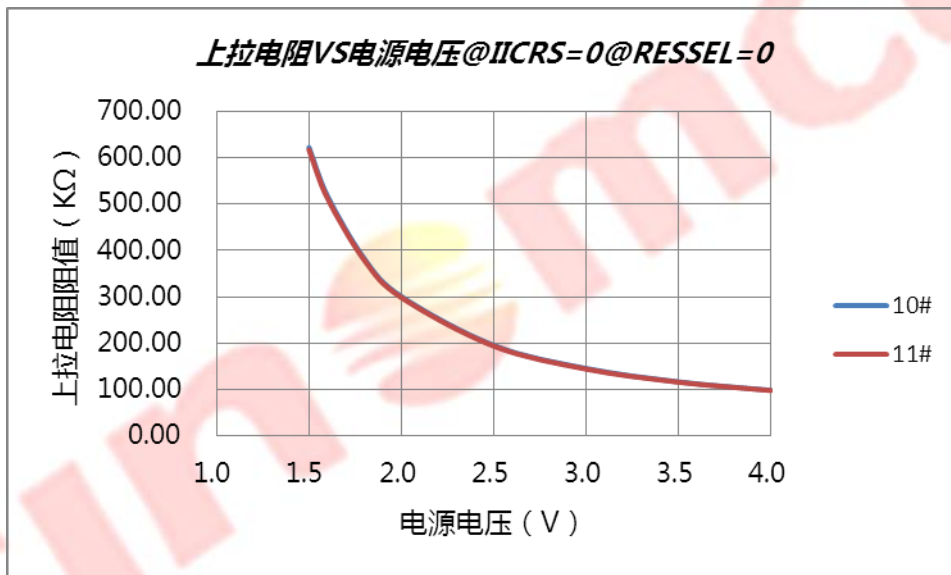


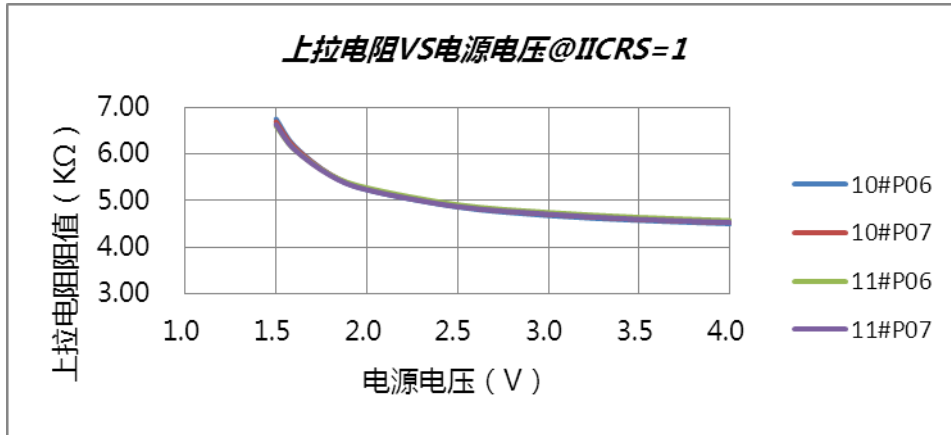
#### I/O 输出 驱动电流 VS 端口电压 (VDD=3V)





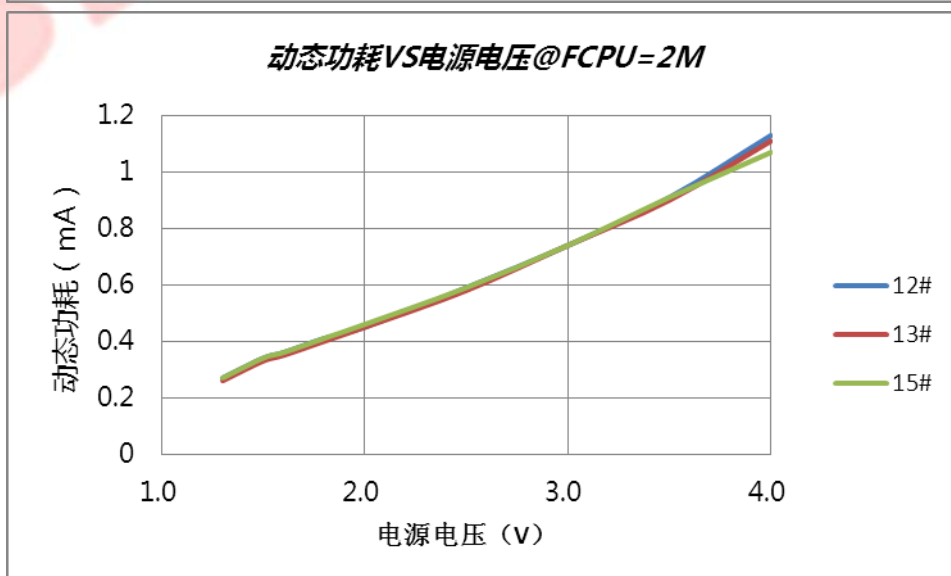
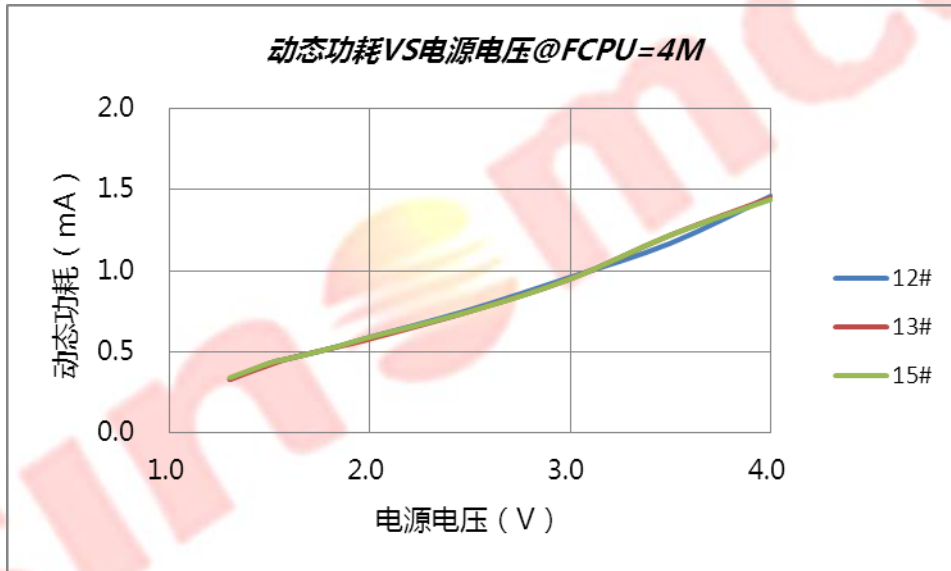
**上/下拉电阻值 VS 电源电压**



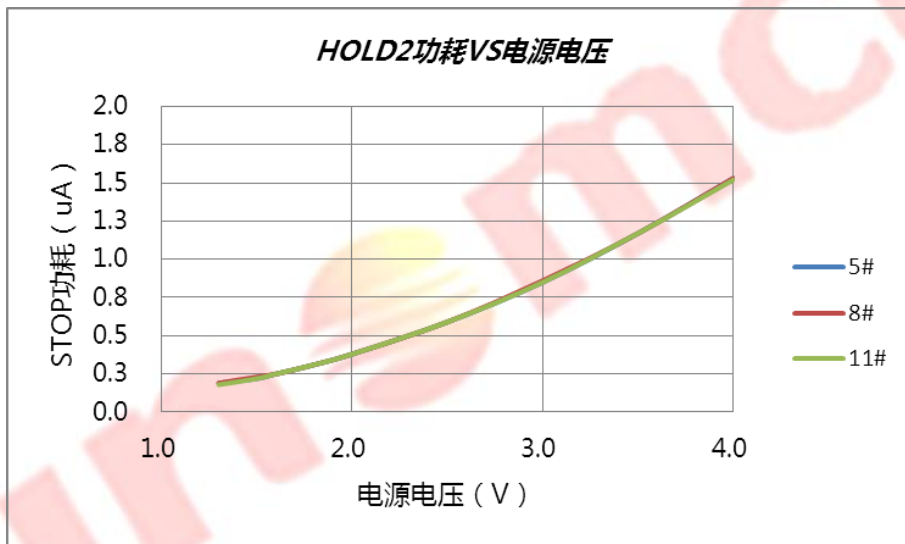
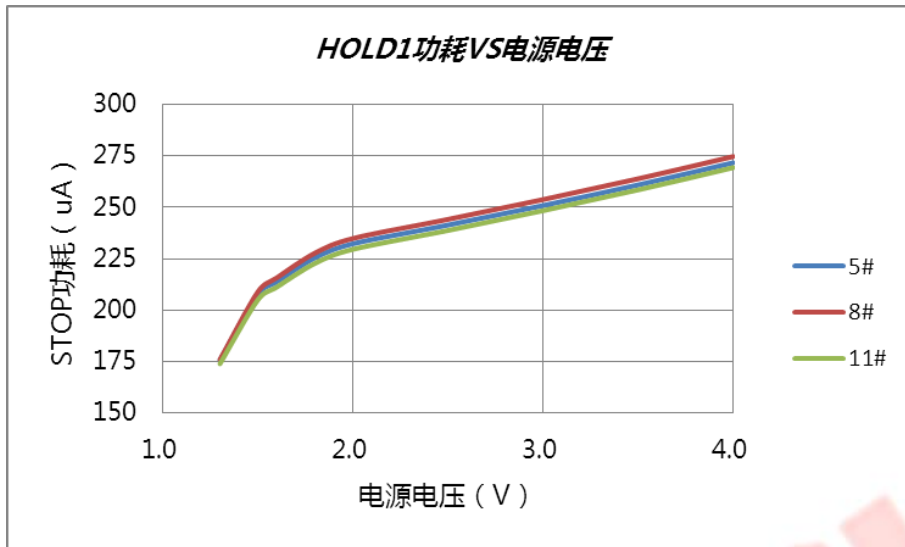


## 12.2 功耗特性

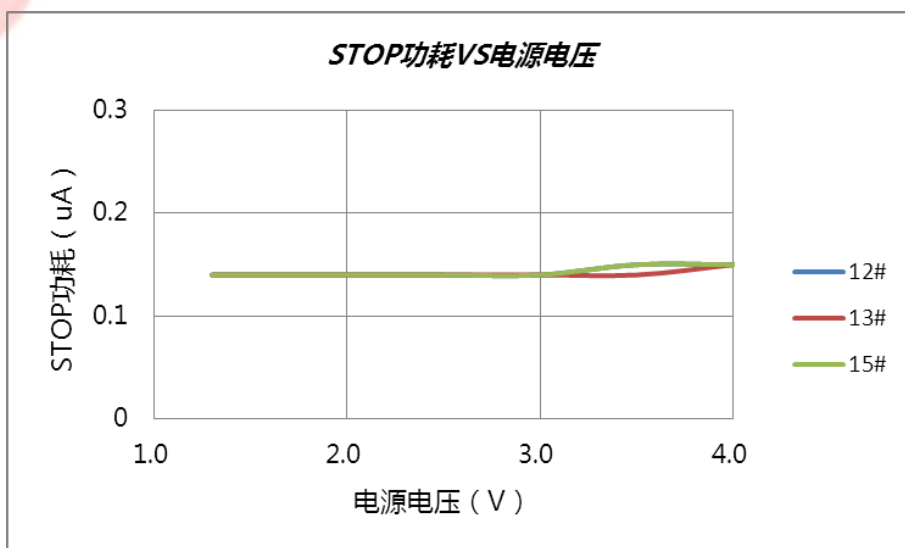
### 运行模式 功耗 VS 电源电压



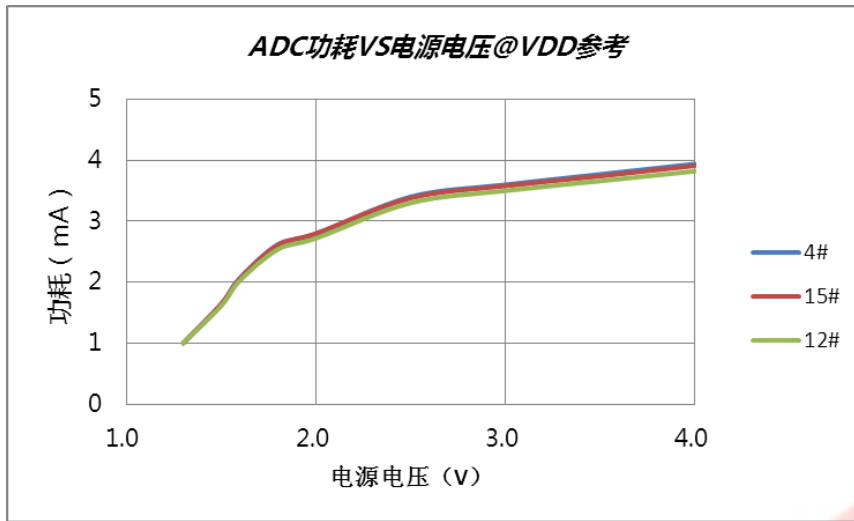
**HOLD 模式 功耗 VS 电源电压**



**休眠模式 功耗 VS 电源电压**

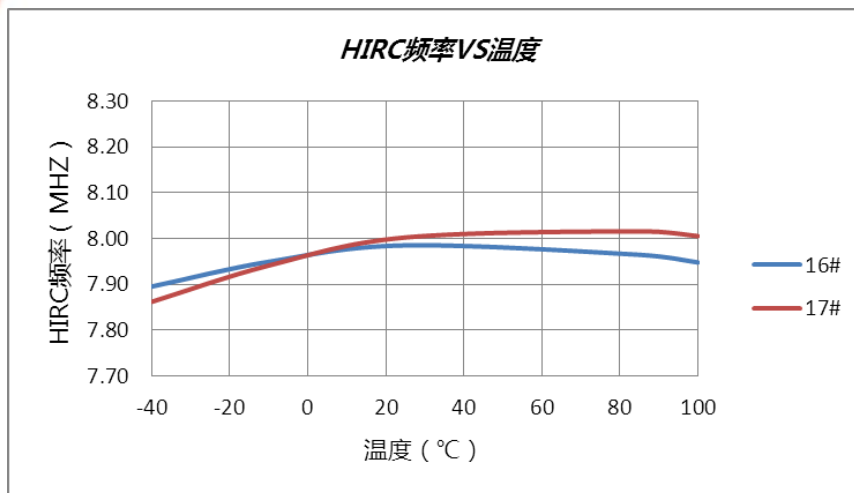
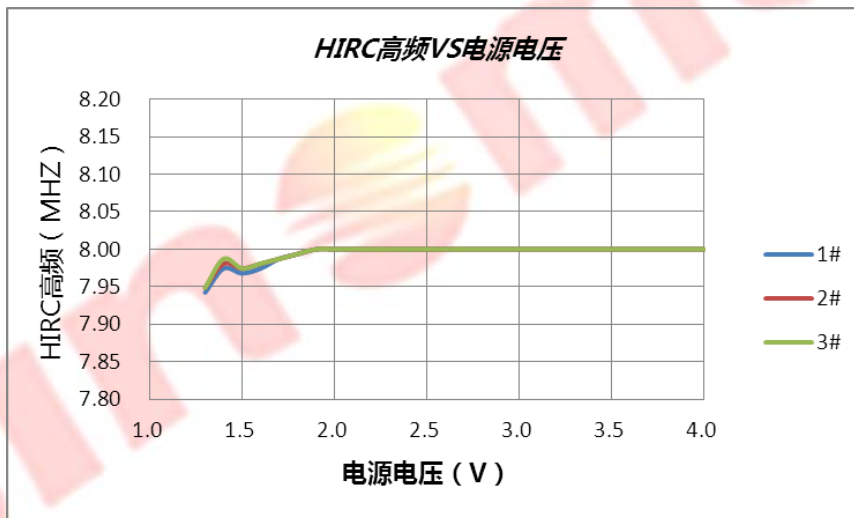


**ADC 功耗 VS 电源电压**

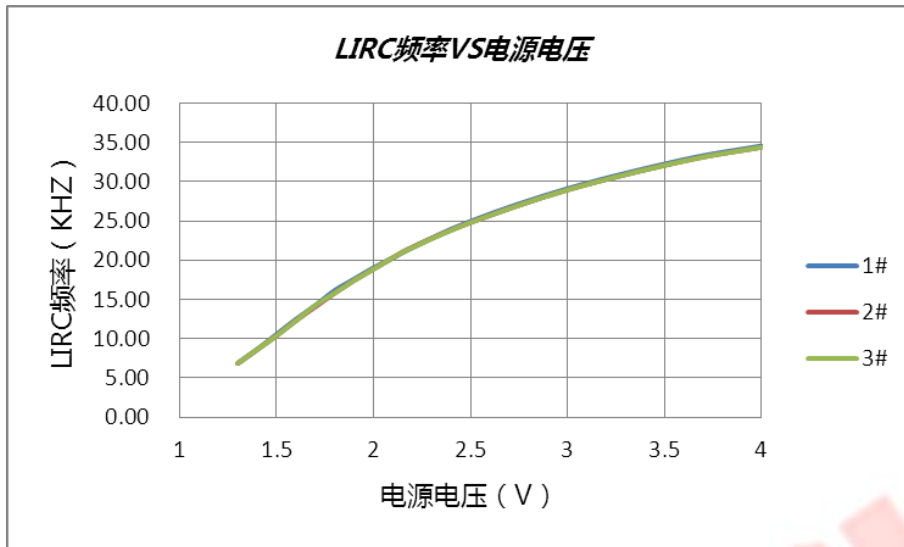


**12.3 模拟电路特性**

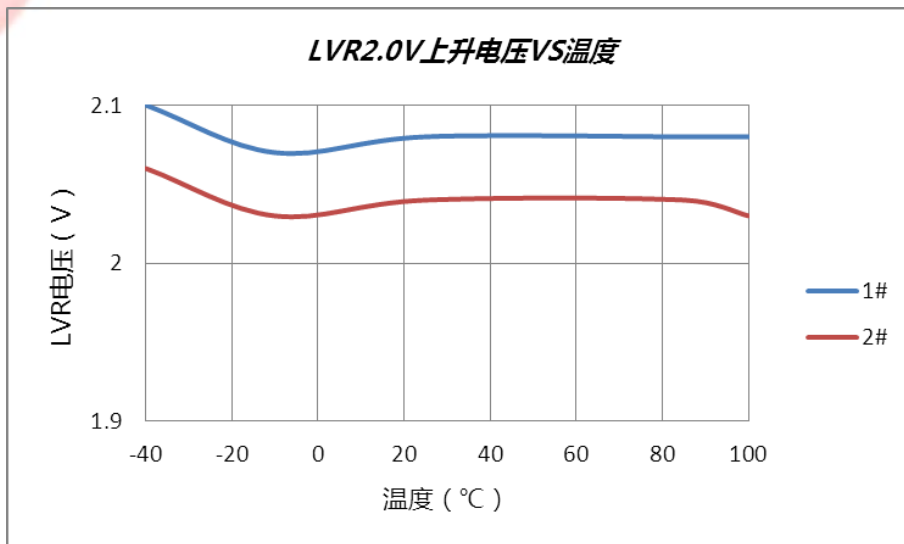
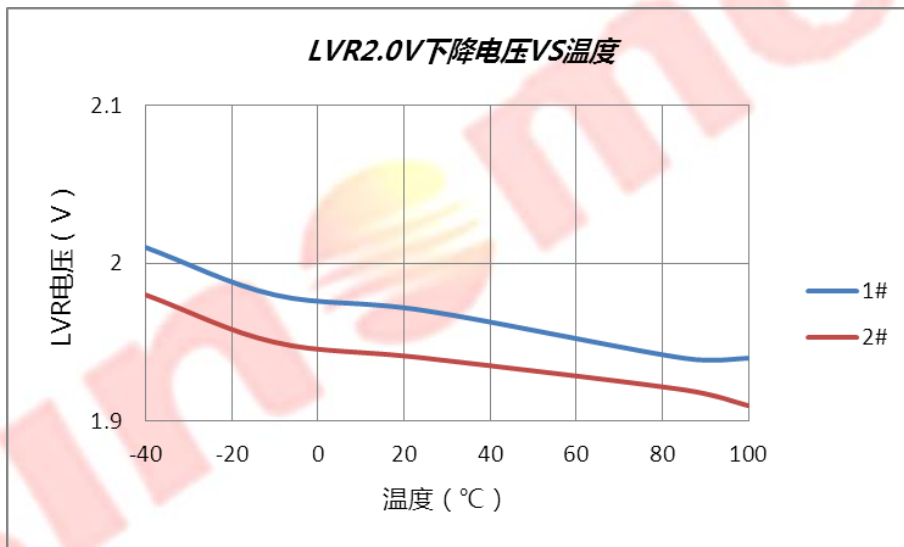
**HIRC 频率 VS 电源电压/温度**



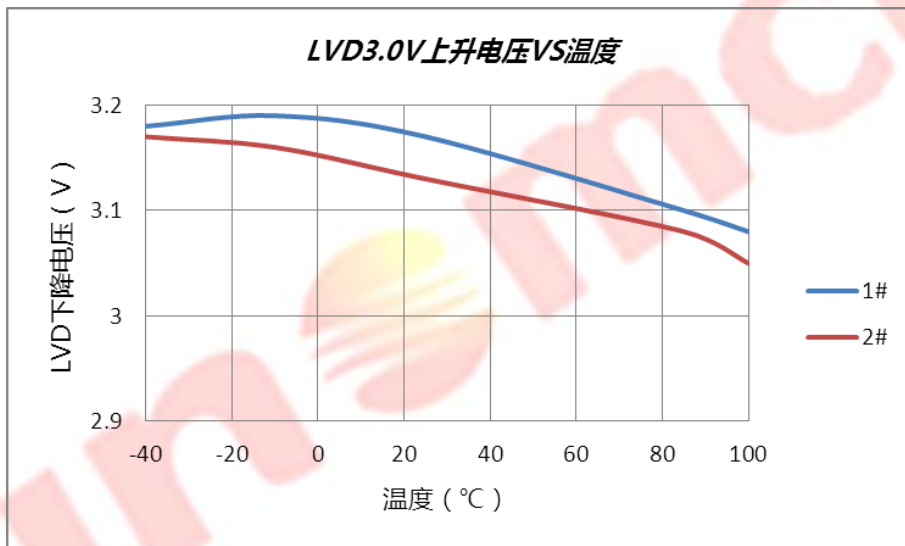
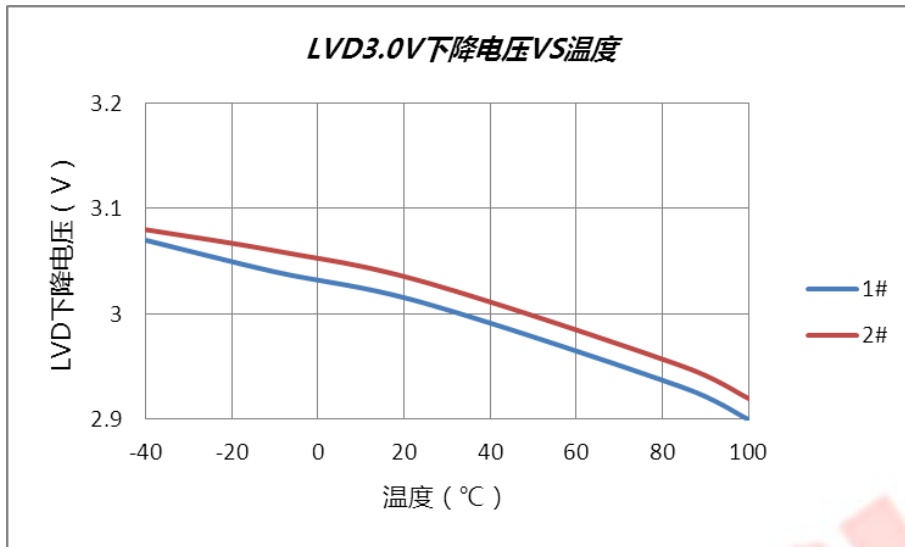
**LIRC 频率 VS 电源电压**



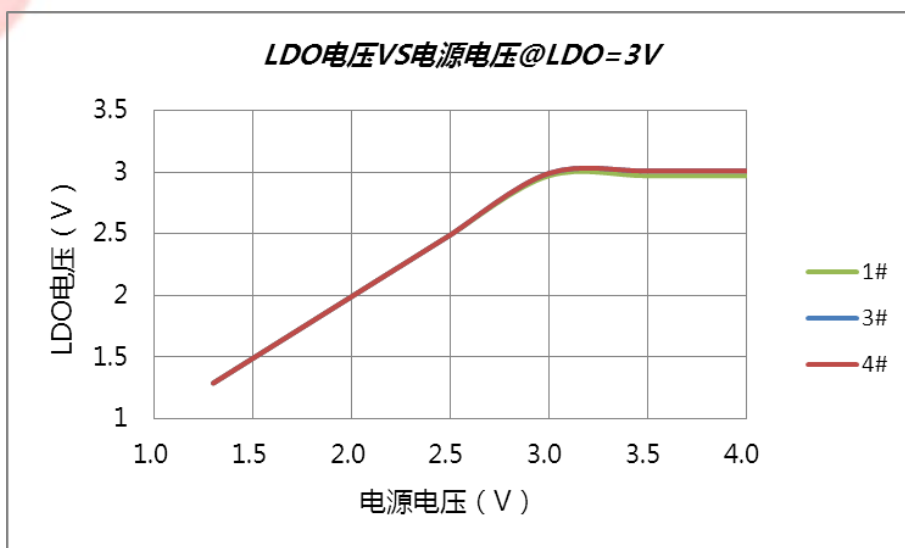
**LVR 电压 VS 温度**



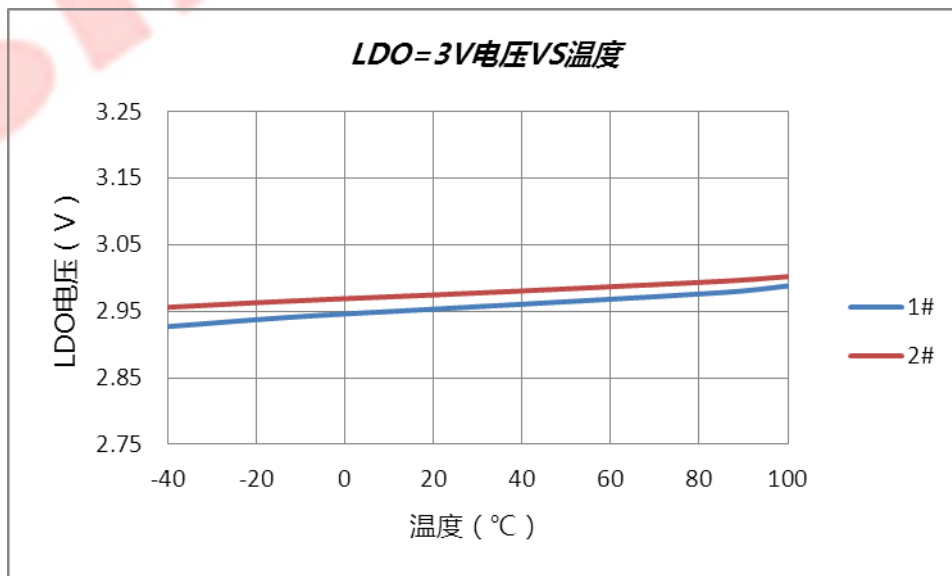
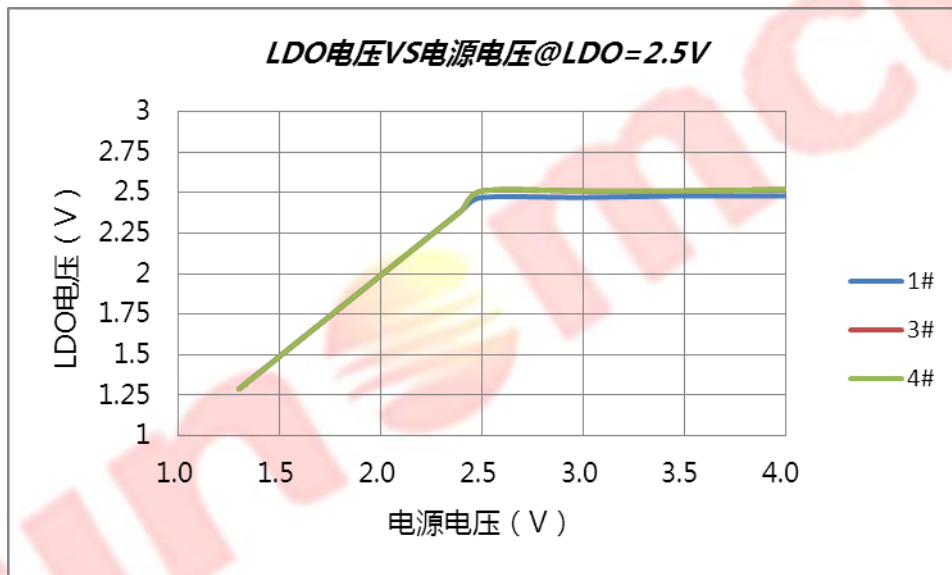
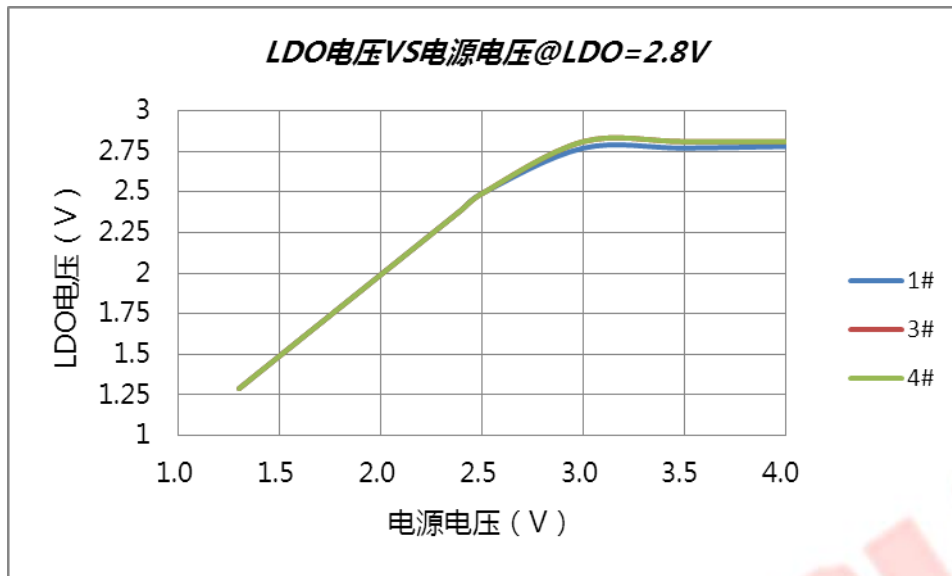
**LVD 电压 VS 温度**

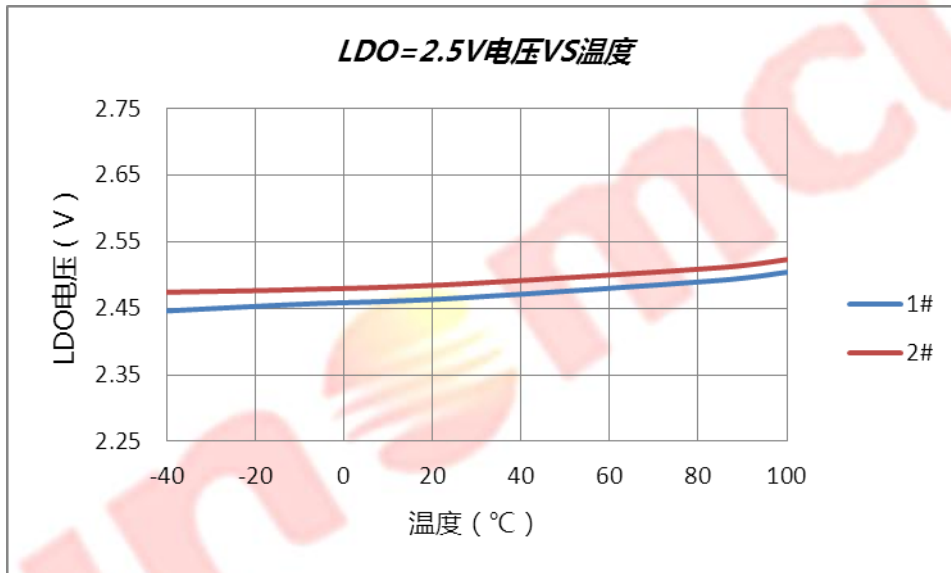
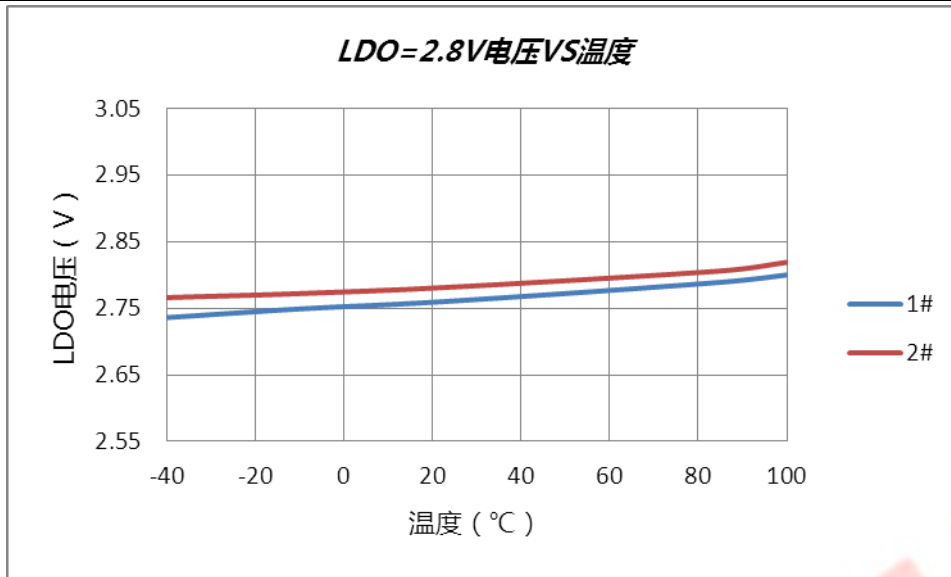


**LDO 电压 VS 电源电压/温度**

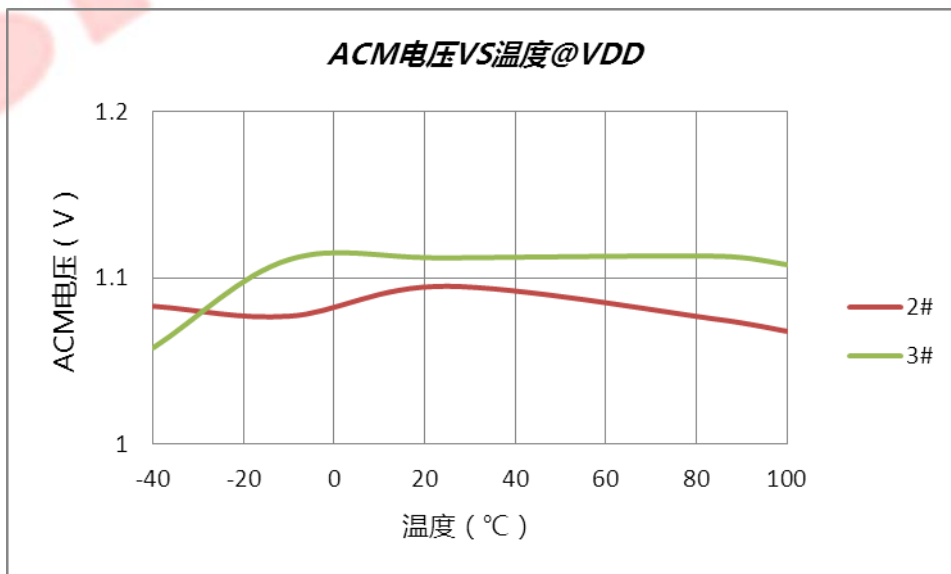


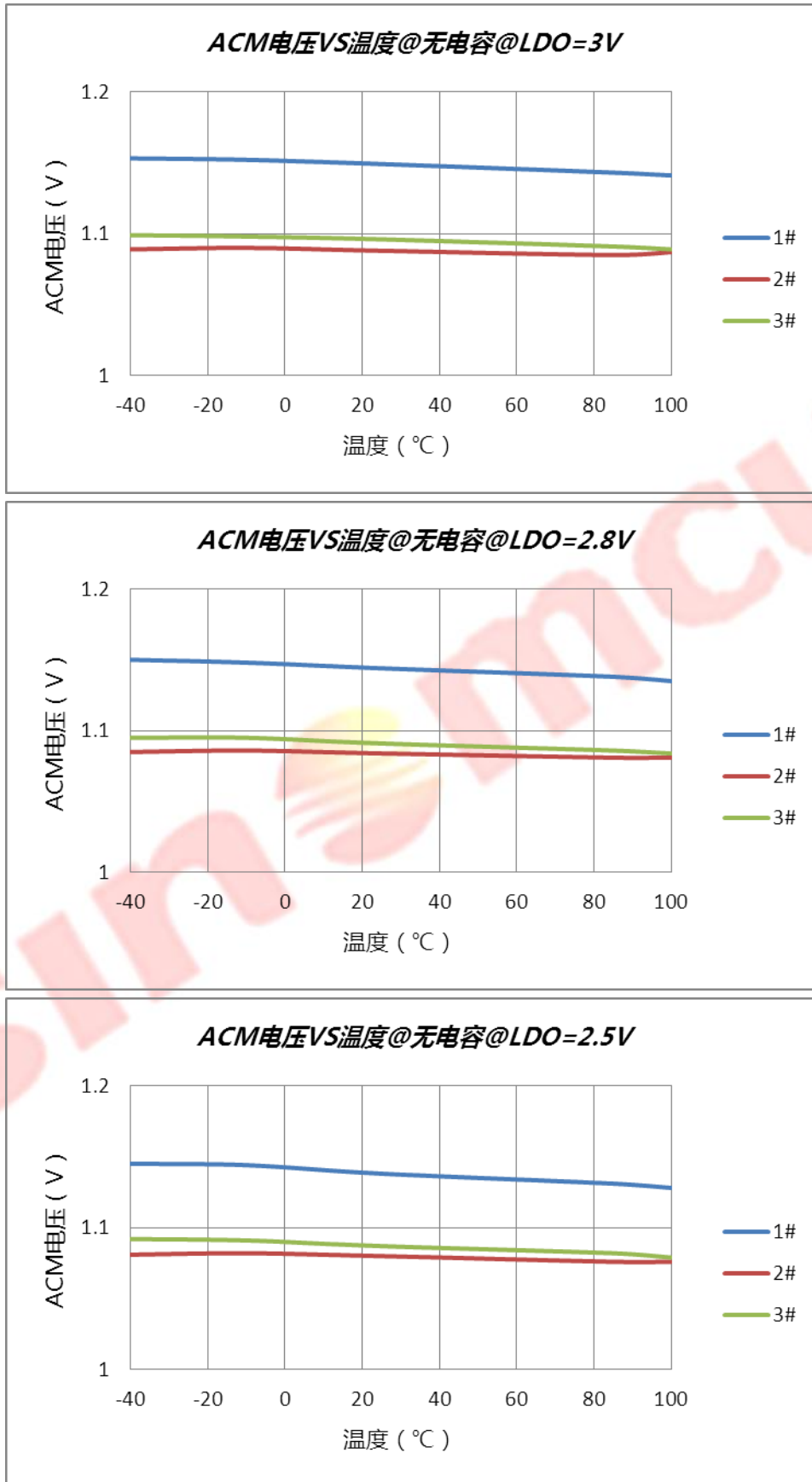




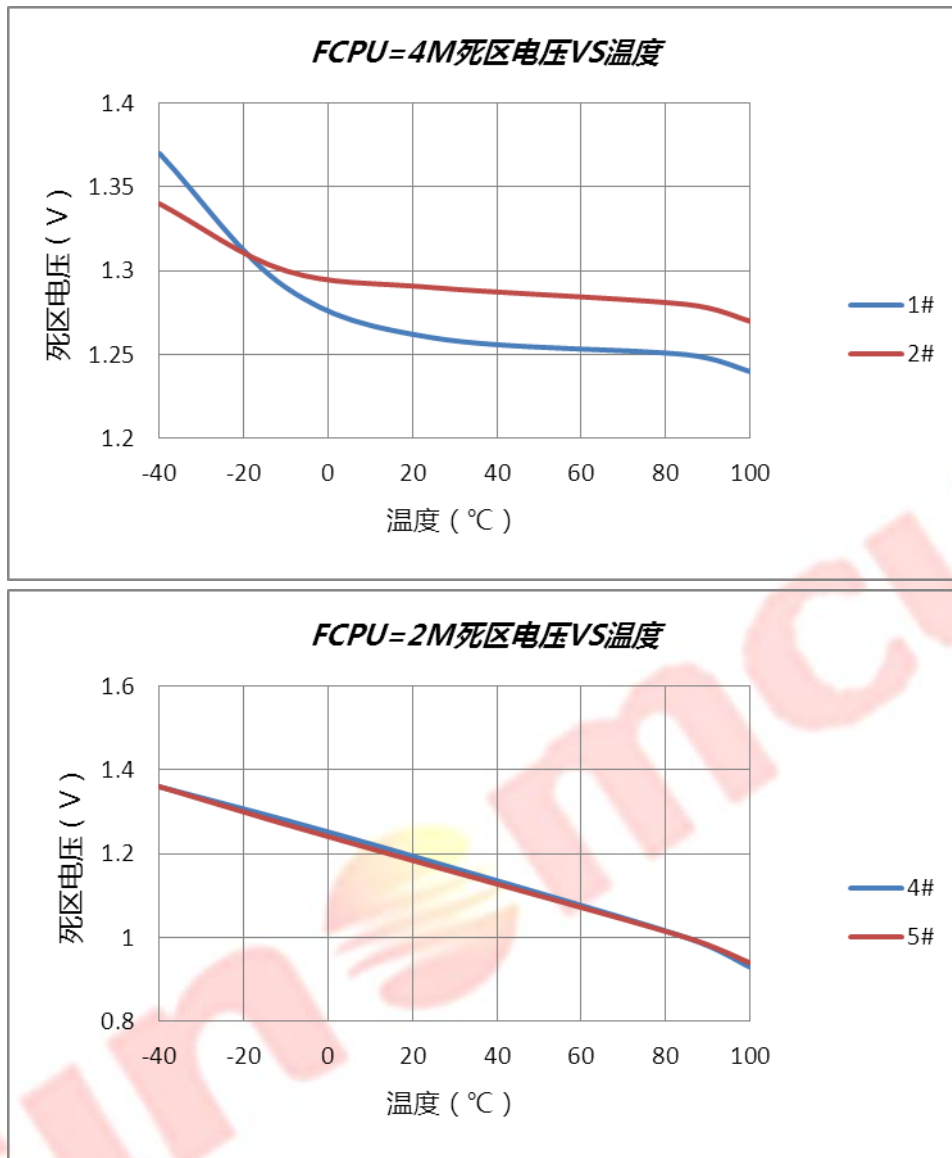


**ACM 电压 VS 温度**



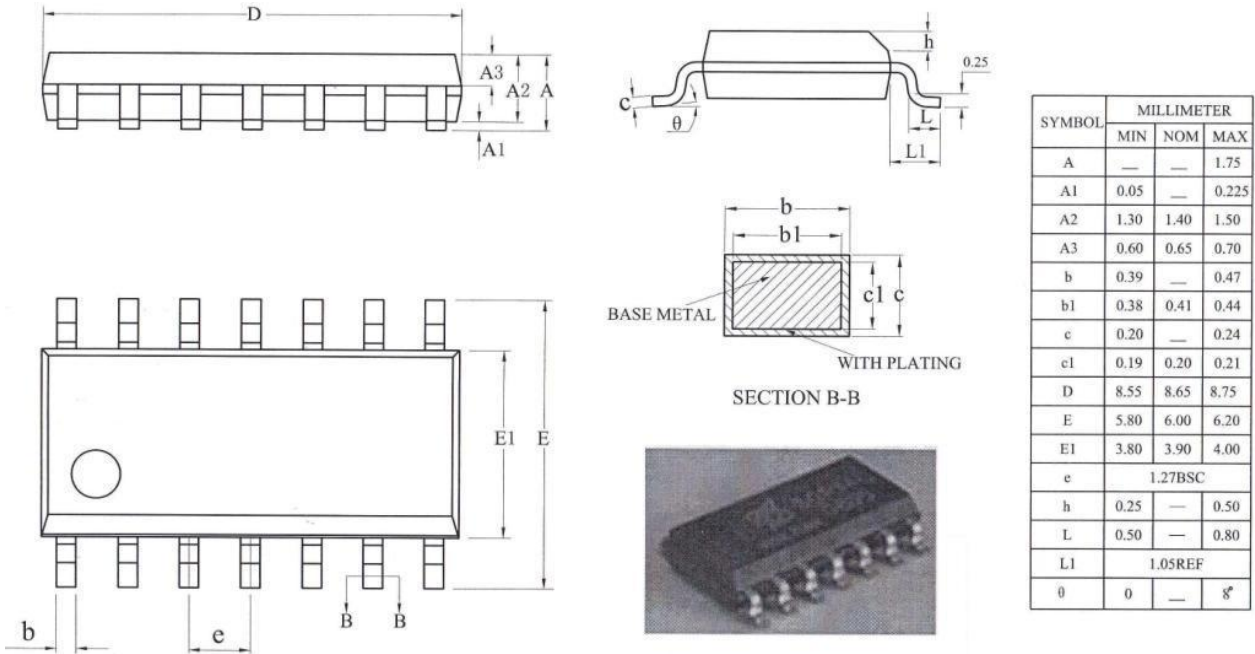


最低工作电压 VS 温度

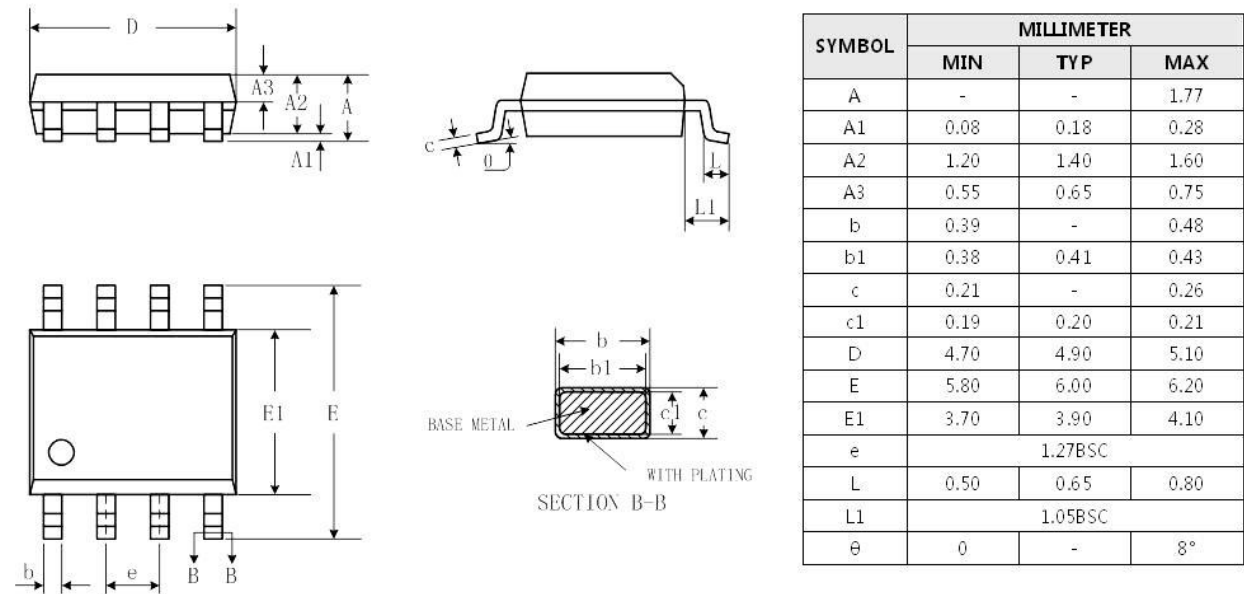


### 13 封装尺寸

#### 13.1 SOP14



#### 13.2 SOP8



## 14 修订记录

版本	修订日期	修订内容
V1.0	2019-03-20	初版发布；
V1.1	2019-05-06	调整 ADC 特性参数；