

晟矽微电 8 位单片机

MC32P5140

用户手册

V1.2





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	7
2	电气特性	8
2.1	极限参数	8
2.2	直流电气特性	8
2.3	交流电气特性	9
3	CPU 与存储器	10
3.1	指令集	10
3.2	程序存储器	12
3.3	数据存储器	13
3.4	堆栈	14
3.5	控制寄存器	14
3.6	用户配置字	17
4	系统时钟	19
4.1	内部高频 RC 振荡器	19
4.2	内部低频 RC 振荡器	19
4.3	系统工作模式	20
4.4	低功耗模式	21
5	复位	23
5.1	复位条件	23
5.2	上电复位	24
5.3	外部复位	24
5.4	低电压复位	24
5.5	看门狗复位	24
6	I/O 端口	25
6.1	通用 I/O 功能	25
6.2	内部上/下拉电阻	26
7	定时器 TIMER	27
7.1	看门狗定时器 WDT	27
8	键盘扫描电路 DKS	28
8.1	DKS 概述	28
8.2	DKS 相关寄存器	29
8.3	DKS 应用示例	30
9	低电压检测 LVD	32
10	中断	33
10.1	外部中断	33
10.2	LVD 中断	33
10.3	中断相关寄存器	33



11	特性曲线.....	35
11.1	I/O 特性.....	35
11.2	功耗特性.....	40
11.3	模拟电路特性.....	43
12	封装尺寸.....	46
12.1	SOP16.....	46
12.2	SOP14.....	46
12.3	SOP8.....	47
13	修订记录.....	48

Sinomcu.com



1 产品概要

1.1 产品特性

- 8 位 CPU 内核
 - ◇ 精简指令集，4 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_{CPU} 可配置为 F_{HOSC} 的 2/4/8/16 分频
 - ◇ 系统低频时钟下 F_{CPU} 固定为 F_{LOSC} 的 2 分频
- 程序存储器
 - ◇ 1K×16 位 OTP 型程序存储器，可通过间接寻址读取程序存储器内容
- 数据存储器
 - ◇ 64 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 14 个 I/O
 - ◇ P0 (P00~P06), P1 (P10~P16)
 - ◇ 所有端口均支持施密特输入，均支持推挽输出
 - ◇ P16 可复用为外部复位 RST 输入，编程时为高压 VPP 输入
 - ◇ P06 (IROUT) 可设置为输入/推挽输出口或高阻/开漏输出口，且开漏输出时输出灌电流 4 级可配置 (125mA/250mA/375mA/500mA)
 - ◇ 所有端口均内置上拉和下拉电阻，均可单独使能
 - ◇ P05 输出源电流 4 级 (1mA/2mA/4mA/8mA) 可选
 - ◇ P00/P15 可复用为外部中断输入，支持外部中断唤醒功能
 - ◇ 所有端口均支持键盘扫描及唤醒功能，并可单独使能
- 系统时钟源
 - ◇ 内置高频 RC 振荡器 (8.06MHz)，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
- 系统工作模式
 - ◇ 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 (低功耗模式)：CPU 暂停，高频时钟源工作，低频时钟源可选停止或工作
 - ◇ HOLD2 模式 (低功耗模式)：CPU 暂停，高频时钟源停止，低频时钟源工作
 - ◇ 休眠模式 (低功耗模式)：CPU 暂停，高/低频时钟源均停止
- 内部自振式看门狗计数器 (WDT)
 - ◇ 溢出时间可配置：16ms/64ms/256ms/1024ms
 - ◇ 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 键盘扫描电路
 - ◇ 14 路键盘扫描端口，均可单独使能
 - ◇ 可组合成 14 路 T 型键盘电路，支持 T 型键扫唤醒
 - ◇ 键盘扫描周期固定为 32ms
- 中断
 - ◇ 外部中断 (INT0~INT1)，LVD 中断
- 低电压检测 LVD
 - ◇ 2.0V/2.4V



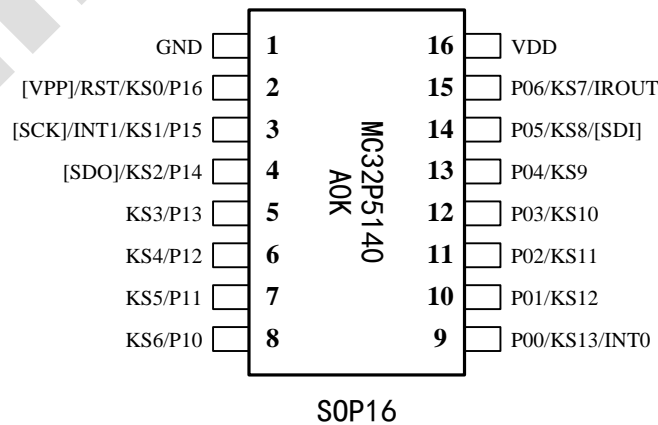
- 低电压复位 LVR
 - ◇ 关闭/1.4V/1.6V/1.8V/2.2V
- 工作电压
 - ◇ VLVR22 ~ 3.6V @ Fcpu = 0~4MHz
 - ◇ VLVR18 ~ 3.6V @ Fcpu = 0~2MHz
 - ◇ VLVR14 ~ 3.6V @ Fcpu = 0~1MHz
 - ◇ VPOR ~ 3.6V @ Fcpu = 0~500KHz
- 封装形式
 - ◇ SOP16/SOP14/SOP8

1.2 订购信息

产品名称	封装形式	备注
MC32P5140A0K	SOP16	
MC32P5140A1K	SOP16	
MC32P5140A0J	SOP14	
MC32P5140A0H	SOP8	

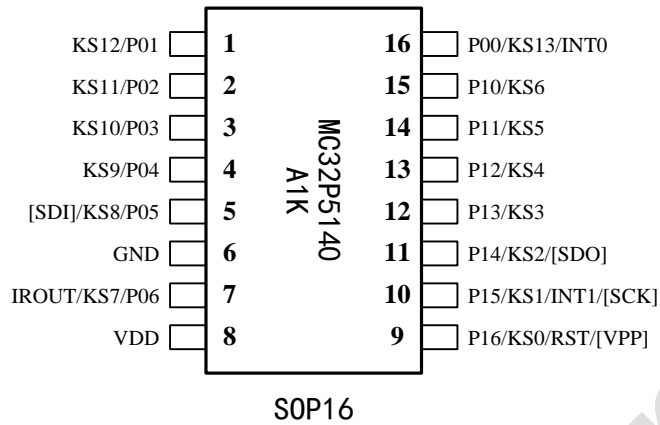
1.3 引脚排列

MC32P5140A0K

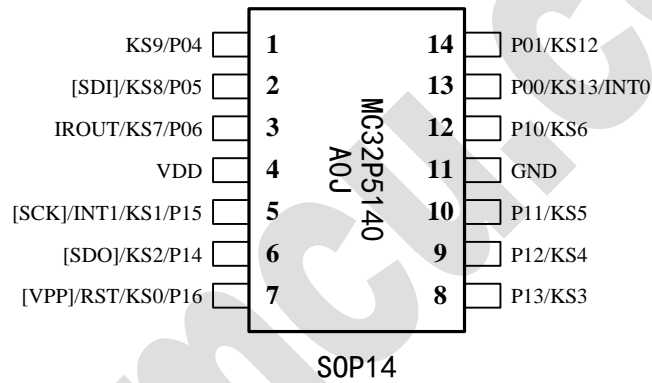




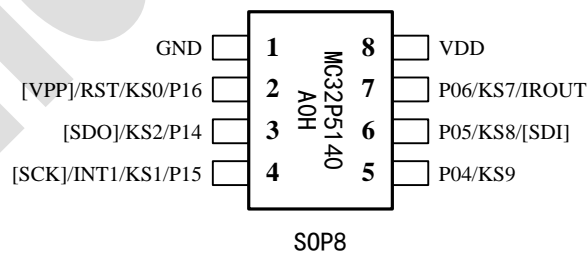
MC32P5140A1K



MC32P5140A0J



MC32P5140A0H





1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P00~P05, P10~P16	D	GPIO (推挽输出), 内部上/下拉
P06	D	GPIO (输入/推挽输出或高阻/开漏输出), 内部上/下拉
INT0~INT1	DI	外部中断输入
KS0~KS13	D	键盘扫描输入/输出
IROUT	AO	红外发射驱动输出
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-电源端口; D-数字端口, DI-数字输入, DO-数字输出; A-模拟端口, 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
I/O 输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-20~70	°C
储存温度	Tstg	-65~150	°C
焊接温度及时间	Tsld	260°C持续 10s	

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

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	VLVR22	3.0	3.6	V
			Fcpu=2MHz@FHIRC/4	VLVR18	3.0	3.6	
			Fcpu=1MHz@FHIRC/8	VLVR14	3.0	3.6	
			Fcpu=500KHz@FHIRC/16	VPOR	3.0	3.6	
RAM 保持电压	Vramhold	VDD		0.9			V
输入漏电流	Ileak	所有输入脚		-1		1	μA
输入高电平	Vih	所有输入脚	IROUT 不支持外接按键	0.8VDD			V
			IROUT 支持外接按键		VDD-0.5		
输入低电平	Vil	所有输入脚	IROUT 不支持外接按键			0.2VDD	V
			IROUT 支持外接按键		VDD-0.5		
上拉电阻	Rpu	P0, P1	Vin=0, RSEL=0, RUSEL 配置		150		KΩ
			Vin=0, RSEL=0, RUSEL 配置		60		KΩ
			Vin=0, RSEL=1		10		KΩ
下拉电阻	Rpd	P0, P1	Vin=VDD		10		KΩ
输出源电流	Ioh1	P0(除 P05), P1	Voh=2.4V		8		mA
	Ioh2	P05	Voh=2.4V, LSEL=00		1		
			Voh=2.4V, LSEL=01		2		
			Voh=2.4V, LSEL=10		4		
			Voh=2.4V, LSEL=11		8		



输出灌电流	Io11	P0, P1	Vol=0.6V, IREN=0		8		mA
	Io12	IROUT(P06)	Vol=0.6V, IREN=1, IRIOLS 配置		125		mA
			Vol=0.6V, IREN=1, IRIOLS 配置		250		
			Vol=0.6V, IREN=1, IRIOLS 配置		375		
			Vol=0.6V, IREN=1, IRIOLS 配置		500		
运行模式功耗	Irun	VDD	Fcpu=4MHz@HIRC		1000		μA
			Fcpu=2MHz@HIRC		700		μA
			Fcpu=1MHz@HIRC		500		μA
			Fcpu=500KHz@HIRC		350		μA
HOLD1 功耗	Ihold1	VDD	CPU 停, HIRC/LIRC 开		800		μA
HOLD2 功耗	Ihold2	VDD	CPU 停, HIRC 关, LIRC 开		0.8	3	μA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.2	1	μA
			休眠模式, WDT 开, LVR 关		0.8	3	μA
			休眠模式, WDT 关, LVR 开		5	10	μA
			休眠模式, LVR 关, DKSE/DKSM=1		0.8	3	μA
低压检测电压	VLVD	VDD	LVDVS 选择	-5%		+5%	V
LVD 响应时间	TLVD			1	50	200	μs
低压复位电压	VLVR	VDD	LVRVS 配置	-10%		+10%	V
上电复位电压	VPOR	VDD	LVR 关闭	-30%	1.2	+30%	V
LVD/LVR 回滞电压		VDD			6%	12%	

注：条件项中，无关模块默认关闭，无关端口设为低电平无负载输出或内部上/下拉电阻无效且外接 GND 的输入。

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	FHIRC	VDD=1.8V~3.6V, T=25°C	-1.5%	8.06	+1.5%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-3%		+3%	
LIRC 振荡频率	FLIRC	VDD=3V, T=25°C	-50%	32	+50%	KHz



3 CPU 与存储器

3.1 指令集

芯片的指令集为精简指令集。

除程序跳转类指令外，其他指令均为单周期指令，即执行时间为 1 个指令周期（CPU 时钟周期）；所有指令均为单字指令，即指令码仅占用 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 A 相加，结果存入 A	$R+A \rightarrow A$	1	1	C, DC, Z
ADDRA R	R 和 A 相加，结果存入 R	$R+A \rightarrow R$	1	1	C, DC, Z
ADCAR R	R 和 A 相加（带 C 标志），结果存入 A	$R+A+C \rightarrow A$	1	1	C, DC, Z
ADCRA R	R 和 A 相加（带 C 标志），结果存入 R	$R+A+C \rightarrow R$	1	1	C, DC, Z
RSUBAR R	R 和 A 相减，结果存入 A	$R-A \rightarrow A$	1	1	C, DC, Z
RSUBRA R	R 和 A 相减，结果存入 R	$R-A \rightarrow R$	1	1	C, DC, Z
RSBCAR R	R 和 A 相减（带 C 标志），结果存入 A	$R-A-C \rightarrow A$	1	1	C, DC, Z
RSBCRA R	R 和 A 相减（带 C 标志），结果存入 R	$R-A-C \rightarrow R$	1	1	C, DC, Z
ASUBAR R	A 和 R 相减，结果存入 A	$A-R \rightarrow A$	1	1	C, DC, Z
ASUBRA R	A 和 R 相减，结果存入 R	$A-R \rightarrow R$	1	1	C, DC, Z
ASBCAR R	A 和 R 相减（带 C 标志），结果存入 A	$A-R-C \rightarrow A$	1	1	C, DC, Z
ASBCRA R	A 和 R 相减（带 C 标志），结果存入 R	$A-R-C \rightarrow R$	1	1	C, DC, Z
ANDAR R	R 和 A 与操作，结果存入 A	$R \text{ and } A \rightarrow A$	1	1	Z
ANDRA R	R 和 A 与操作，结果存入 R	$R \text{ and } A \rightarrow R$	1	1	Z
ORAR R	R 和 A 或操作，结果存入 A	$R \text{ or } A \rightarrow A$	1	1	Z
ORRA R	R 和 A 或操作，结果存入 R	$R \text{ or } A \rightarrow R$	1	1	Z
XORAR R	R 和 A 异或操作，结果存入 A	$R \text{ xor } A \rightarrow A$	1	1	Z
XORRA R	R 和 A 异或操作，结果存入 R	$R \text{ xor } A \rightarrow R$	1	1	Z
COMAR R	对 R 取反，结果存入 A	$R \text{ 取反} \rightarrow A$	1	1	Z
COMR R	对 R 取反，结果存入 R	$R \text{ 取反} \rightarrow R$	1	1	Z
RLA	A 循环左移（带 C 标志）	$A[7] \rightarrow C; A[6:0] \rightarrow A[7:1]; C \rightarrow A[0]$	1	1	C
RLAR R	R 循环左移（带 C 标志），结果存入 A	$R[7] \rightarrow C; R[6:0] \rightarrow A[7:1]; C \rightarrow A[0]$	1	1	C
RLR R	R 循环左移（带 C 标志），结果存入 R	$R[7] \rightarrow C; R[6:0] \rightarrow R[7:1]; C \rightarrow R[0]$	1	1	C
RRA	A 循环右移（带 C 标志）	$A[0] \rightarrow C; A[7:1] \rightarrow A[6:0]; C \rightarrow A[7]$	1	1	C
RRAR R	R 循环右移（带 C 标志），结果存入 A	$R[0] \rightarrow C; R[7:1] \rightarrow A[6:0]; C \rightarrow A[7]$	1	1	C
RRR R	R 循环右移（带 C 标志），结果存入 R	$R[0] \rightarrow C; R[7:1] \rightarrow R[6:0]; C \rightarrow R[7]$	1	1	C
SWAPAR R	交换 R 的高低半字节，结果存入 A	$R[7:4] \rightarrow A[3:0]; R[3:0] \rightarrow A[7:4]$	1	1	-
SWAPR R	交换 R 的高低半字节，结果存入 R	$R[7:4] \rightarrow R[3:0]; R[3:0] \rightarrow R[7:4]$	1	1	-



MOVRA	R	将 A 存入 R	$A \rightarrow R$	1	1	-
MOVAR	R	将 R 存入 A	$R \rightarrow A$	1	1	Z
MOVRR	R	将 R 存入 R	$R \rightarrow R$	1	1	Z
CLRA		将 A 清零	$0 \rightarrow A$	1	1	Z
CLRR	R	将 R 清零	$0 \rightarrow R$	1	1	Z
INCA		A 自加 1	$A+1 \rightarrow A$	1	1	-
INCR	R	R 自加 1	$R+1 \rightarrow R$	1	1	Z
INCAR	R	R 加 1, 结果存入 A	$R+1 \rightarrow A$	1	1	Z
DECA		A 自减 1	$A-1 \rightarrow A$	1	1	-
DECR	R	R 自减 1	$R-1 \rightarrow R$	1	1	Z
DECAR	R	R 减 1, 结果存入 A	$R-1 \rightarrow A$	1	1	Z
JZA		A 自加 1: 结果为 0 则跳过下一条指令	$A+1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
JZR	R	R 自加 1: 结果为 0 则跳过下一条指令	$R+1 \rightarrow R$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
JZAR	R	R 加 1, 结果存入 A: 结果为 0 则跳过下一条指令	$R+1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZA		A 自减 1: 结果为 0 则跳过下一条指令	$A-1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZR	R	R 自减 1: 结果为 0 则跳过下一条指令	$R-1 \rightarrow R$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
DJZAR	R	R 减 1, 结果存入 A: 结果为 0 则跳过下一条指令	$R-1 \rightarrow A$: 结果为 0 则 $PC+2 \rightarrow PC$	1/2	1	-
BCLR	R, b	将 R 的第 b 位清 0	$0 \rightarrow R[b]$	1	1	-
BSET	R, b	将 R 的第 b 位置 1	$1 \rightarrow R[b]$	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 $PC+2 \rightarrow PC$	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 $PC+2 \rightarrow PC$	1/2	1	-
ADDAI	I	I 和 A 相加, 结果存入 A	$I+A \rightarrow A$	1	1	C, DC, Z
ADCAI	I	I 和 A 相加 (带 C 标志), 结果存入 A	$I+A+C \rightarrow A$	1	1	C, DC, Z
ISUBAI	I	I 和 A 相减, 结果存入 A	$I-A \rightarrow A$	1	1	C, DC, Z
ISBCAI	I	I 和 A 相减 (带 C 标志), 结果存入 A	$I-A-/C \rightarrow A$	1	1	C, DC, Z
ASUBAI	I	A 和 I 相减, 结果存入 A	$A-I \rightarrow A$	1	1	C, DC, Z
ASBCAI	I	A 和 I 相减 (带 C 标志), 结果存入 A	$A-I-/C \rightarrow A$	1	1	C, DC, Z
ANDAI	I	I 和 A 与操作, 结果存入 A	$I \text{ and } A \rightarrow A$	1	1	Z
ORAI	I	I 和 A 或操作, 结果存入 A	$I \text{ or } A \rightarrow A$	1	1	Z
XORAI	I	I 和 A 异或操作, 结果存入 A	$I \text{ xor } A \rightarrow A$	1	1	Z
MOVAI	I	将 I 存入 A	$I \rightarrow A$	1	1	-
CALL	K	子程序调用	$PC+1 \rightarrow TOS$; $K \rightarrow PC[12:0]$	2	1	-
GOTO	K	无条件跳转	$K \rightarrow PC[12:0]$	2	1	-
RETURN		从子程序返回	$TOS \rightarrow PC$	2	1	-
RETAI	I	从子程序返回, 并将 I 存入 A	$TOS \rightarrow PC$; $I \rightarrow A$	2	1	-
RETIE		从中断返回	$TOS \rightarrow PC$; $1 \rightarrow GIE$	2	1	-
NOP		空操作	空操作	1	1	-
DAA		BCD 码加法操作后, 将 A 的值调整为 BCD 码	$A(\text{HEX 码}) \rightarrow A(\text{BCD 码})$	1	1	C
DSA		BCD 码减法操作后, 将 A 的值调整为 BCD 码	$A(\text{HEX 码}) \rightarrow A(\text{BCD 码})$	1	1	-



CLRWDT	将看门狗计数器清零	0→WDTCNT	1	1	TO, PD
STOP	进入低功耗模式	0→WDTCNT; CPU 暂停	1	1	TO, PD

注:

- 1、A-算术逻辑单元累加器 ALU, R-数据存储器, I-立即数, K-程序存储器地址, TOS-堆栈栈顶;
- 2、对于条件跳转类指令, 若跳转条件成立, 则执行时间需 2 个指令周期, 否则仅需 1 个指令周期;
- 3、禁止采用对 C, DC, Z 标志有影响的指令访问寄存器 PFLAG;

3.2 程序存储器

芯片的程序存储器为 OTP 型存储器, 1K×16 位的地址空间范围为 0000H~03FFH。程序存储器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 - 03FFH)

程序存储器支持间接寻址, 可通过寄存器 INDF3 访问地址为 (FSR1×256+FSR0) 的程序存储器内容, 高 8 位将缓存于寄存器 HIBYTE, 低 8 位将缓存于寄存器 A。

例如, 采用间接寻址读取程序存储器 0155H 地址中内容, 高 8 位存入通用数据存储器 11H 地址中, 低 8 位存入通用数据存储器 10H 地址中:

```

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) 所指地址的程序存储器中内容
                                ; 高 8 位缓存于 HIBYTE, 低 8 位缓存于 A
MOVRA    10H           ; 将 A 中缓存的低 8 位存入通用数据存储器 10H 地址中
MOVAR    HIBYTE        ; 读取 HIBYTE 中缓存的高 8 位
MOVRA    11H           ; 高 8 位存入通用数据存储器 11H 地址中

```



3.3 数据存储器

芯片的数据存储器包括通用数据存储器 GPR（64 字节）和特殊功能寄存器 SFR，地址映射如下表所示。其中 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-03FH	GPR	通用数据存储器区							
040H-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	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		DKSCR	PODKCR	P1DKCR					
1A0H-1FFH	保留	保留							

注：上表中灰色部分的存储器地址为系统保留区，禁止对其中未定义的地址进行读写操作。

数据存储器寻址方式地址组成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址方式
/	/	/	/	/	/	/	取自指令的 9 位地址								直接寻址方式	
/	/	/	/	/	/	/	0	FSR0								间接寻址方式 0
/	/	/	/	/	/	/	1	FSR1								间接寻址方式 1
FSR1							FSR0								间接寻址方式 2	

直接寻址方式，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 0~1FFH。例如，采用直接寻址方式将数据 55H 写入数据存储器 010H 地址中：

```
MOVAI    55H
MOVRA    10H           ; 将 55H 写入数据存储器 10H 地址中
```

间接寻址方式 0，是以 FSR0 为数据存储器地址指针，通过 INDF0 访问，寻址范围 0~0FFH。例如，采用间接寻址方式 0 将数据 55H 写入数据存储器 010H 地址中：

```
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF0       ; 将 55H 写入 FSR0 所指地址的数据存储器中
```

间接寻址方式 1，是以 FSR1 为数据存储器地址指针，通过 INDF1 访问，寻址范围 100H~1FFH。例如，采用间接寻址方式 1 将数据 55H 写入数据存储器 110H 地址中：

```
MOVAI    10H
MOVRA    FSR1
```



MOVAI 55H
 MOVRA INDF1 ; 将 55H 写入 (FSR1+256) 所指地址的数据存储器中

间接寻址方式 2, 是以[FSR1:FSR0]为数据存储器地址指针, 通过 INDF2 访问, 寻址范围 0~FFFFH。
 例如, 采用间接寻址方式 2 将数据 55H 写入数据存储器 0010H 地址中:

MOVAI 00H
 MOVRA FSR1
 MOVAI 10H
 MOVRA FSR0
 MOVAI 55H
 MOVRA INDF2 ; 将 55H 写入 (FSR1×256+FSR0) 所指地址的数据存储器中

注: 间接寻址方式 2 最大可寻址 FFFFH, 但访问数据存储器中未定义的地址时, 读出数据不确定, 写入操作可能会更改其他地址中的内容。

3.4 堆栈

芯片的堆栈为 4 级深度的硬件堆栈。当 CPU 响应中断或执行子程序调用指令时, 会自动将下一条指令的 PC 值压栈保存; 当 CPU 执行中断返回或子程序返回指令时, 会自动将栈顶内容出栈载入 PC。

3.5 控制寄存器

数据指针寄存器 0

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

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

FSR0: 间接寻址方式 0 的指针, 或间接寻址方式 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]
FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

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



间接寻址寄存器 0

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	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 操作实际是对 FSR0 所指向地址的数据存储器进行操作, 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]
INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	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 操作实际是对 (FSR1+256) 所指向地址的数据存储器进行操作, 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]
INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	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 操作实际是对 (FSR1×256+FSR0) 所指向地址的数据存储器进行操作, 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]
INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	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 操作实际是对 (FSR1×256+FSR0) 所指向地址的程序存储器进行操作, 从而实现间接寻址功能。

注: 对寄存器 INDF3 仅可执行读取操作, 且仅可使用读取指令 (MOVAR INDF3), 所读程序存储器内容的高 8 位存入寄存器 HIBYTE, 低 8 位存入寄存器 A。



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]
HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节缓存器

HIBYTE: 用于缓存通过 INDF3 访问程序存储器时所读取内容的高 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]
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	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	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: $PC = PC + 1$;
- ◇ 程序跳转指令 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 操作的其他指令: $PC = (PC[12:8]:ALU[7:0](ALU \text{ 运算结果}))$;

CPU 状态寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PFLAG	-	-	-	-	-	Z	DC	C
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	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** – 半字节进位/借位标志位

- 0: 加法运算中半字节无进位; 减法运算中半字节有借位;
- 1: 加法运算中半字节有进位; 减法运算中半字节无借位;

BIT[0] **C** – 进位/借位标志位

- 0: 加法运算中无进位; 减法运算中有借位; 移位操作中移出位为 0;
- 1: 加法运算中有进位; 减法运算中无借位; 移位操作中移出位为 1;



杂项控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MCR	GIE	-	TO	PD	INT1M1	INT1M0	INT0M1	INT0M0
R/W	R/W	-	R	R	R/W	R/W	R/W	R/W
初始值	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, 或已执行 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, 或已执行 CLRWDT 指令;

1: 已执行 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
1X	电平变化触发

BIT[1:0] **INT0M[1:0]** – 外部中断 INT0 触发方式选择位

INT0M[1:0]	INT0 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

3.6 用户配置字

芯片为保证系统正常工作, 会将关键模块的配置信息预先存储于单独的存储器区域内, 在上电或其他复位发生后将配置信息载入寄存器中, 通过寄存器控制关键模块的工作状态。该部分存储器中用户可选的内容即为用户配置字, 可在烧录用户程序代码时进行配置与烧录。



4 系统时钟

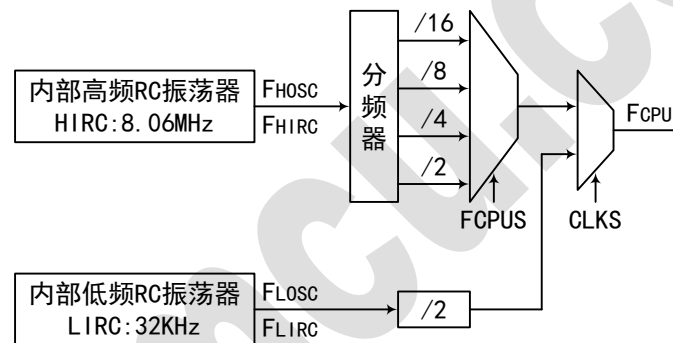
芯片内部电路均在系统高频时钟 F_{HOSC} 或系统低频时钟 F_{LOSC} 下工作，系统及部分外设模块的时钟源还可在 F_{HOSC} 和 F_{LOSC} 之间切换。

系统高频时钟 F_{HOSC} 固定为内部高频 RC 振荡器 $HIRC$ (8.06MHz) 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。 F_{HOSC} 下 CPU 的时钟频率 F_{CPU} 通过配置字 $FCPUS$ 选择； F_{LOSC} 下 F_{CPU} 则固定为 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 $LIRC$ 。

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 8.06MHz 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、看门狗定时器 (WDT) 等电路。



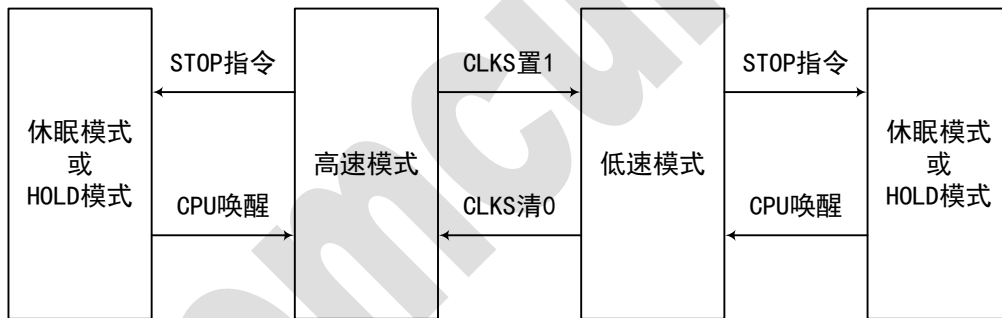
4.3 系统工作模式

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。

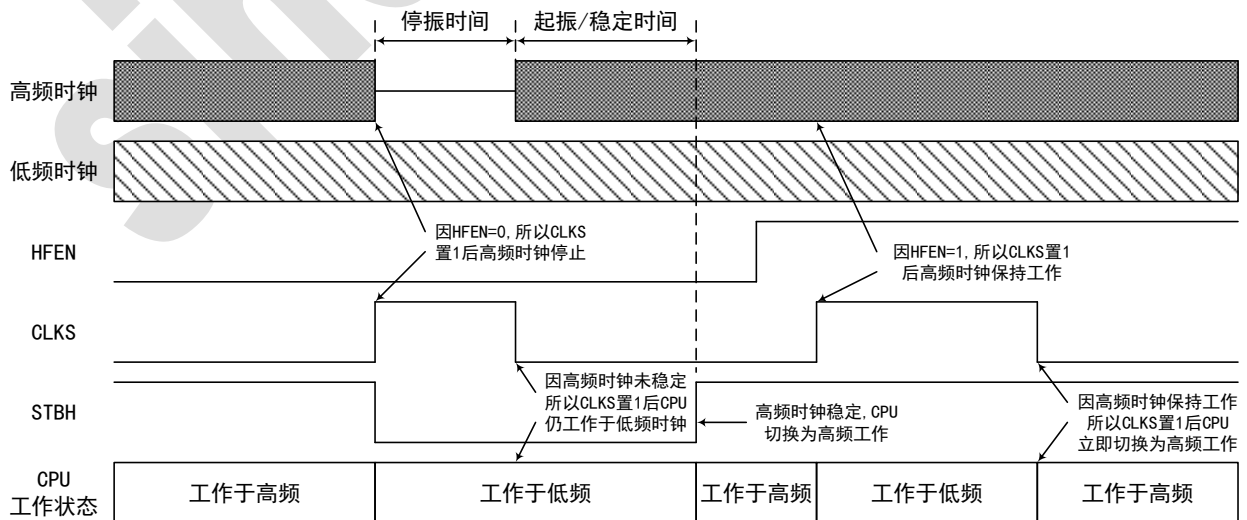
工作模式	模式切换条件	系统工作状态
高速	任意模式下，系统复位	CPU 高速运行，高/低频时钟源均工作
	低速模式下，CLKS 清 0	
	HOLD1/HOLD2/休眠模式下，CPU 唤醒 (@CLKS=0)	
低速	高速模式下，CLKS 置 1	CPU 低速运行，低频时钟源工作，高频时钟源由使能位 HFEN 决定
	HOLD1/HOLD2/休眠模式下，CPU 唤醒 (@CLKS=1)	
HOLD1	高/低速模式下，执行 STOP 指令 (@HFEN=1)	CPU 暂停，高频时钟源工作，低频时钟源由使能位 LFEN 决定
HOLD2	高/低速模式下，执行 STOP 指令 (@HFEN=0, LFEN=1)	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，执行 STOP 指令 (@HFEN=0, LFEN=0)	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





振荡器模式寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OSCMR	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
R/W	-	-	R	R	-	R/W	R/W	R/W
初始值	-	-	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, 低频时钟源暂停工作;
- 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, 高频时钟源暂停工作;
- 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/输出端口保持原有状态;

以下情况可使系统退出低功耗模式:

- ◇ 芯片复位;
- ◇ WDT 溢出 (若低功耗模式下 WDT 及其时钟源保持继续工作);
- ◇ 外部中断请求发生 (若有外部中断功能并有效);
- ◇ LVD 中断请求发生 (若低功耗模式下 LVD 保持继续工作);
- ◇ 键盘扫描端口有电平变化发生 (若有键盘扫描功能并有效);



注:

- 1、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 后将执行中断服务程序；
- 2、未使用或未封出的引脚，应将其对应的 I/O 端口设置为输出、输入上拉或输入下拉等稳定状态，以免因引脚浮空而产生漏电流或非预期的中断唤醒；

Sinomcu.com



5 复位

5.1 复位条件

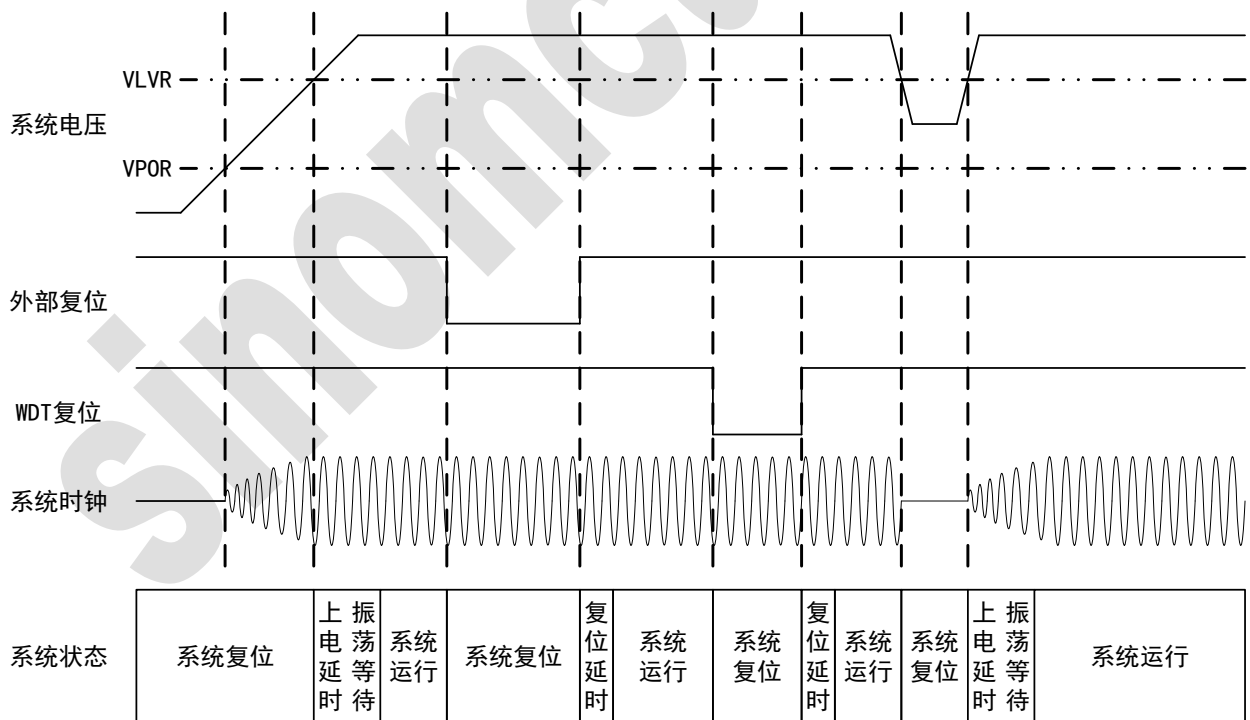
芯片共有如下几种复位方式：

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

任何一种复位发生后，系统进入复位状态，执行初始化操作并重置 SFR 为复位初始值；复位条件解除后，系统退出复位状态，CPU 重新从程序存储器 0000H 地址处开始运行。

上电复位 POR 和低电压复位 LVR 会关闭系统主时钟振荡器，复位解除后才重新开启振荡器，因为振荡器起振和稳定需要一定的时间，所以系统将保持一定时间的上电延时（典型值为 16ms）以待振荡器稳定振荡后才开始工作；而外部复位、WDT 复位则不会关闭主时钟振荡器，复位解除后系统将在较短的复位延时后即开始工作。

下图是复位产生和系统工作状态之间的时序关系示意图：



注：若应用系统在上电或掉电回升时芯片的 VDD 电压上升较慢，则应在复位后 CPU 开始工作时先进行软件延时，以确保芯片开始工作时 VDD 电压已稳定在 F_{CPU} 对应的工作电压范围内。



5.2 上电复位

芯片的上电复位电路可以适应系统快速上电或慢速上电等情况，即使上电过程中发生电源电压抖动的情况也能保证系统可靠的复位。

上电复位过程主要包括以下几个步骤：

- (1) 检测系统电源电压，等待电压高于上电复位电压 V_{POR} 并保持稳定；
- (2) 若 LVR 功能开启，则需等待电压高于低电压复位电压 V_{LVR} 并保持稳定；
- (3) 若有外部复位功能并已开启，则需等待外部复位引脚电压高于 V_{ih} ；
- (4) 初始化所有初始值确定的寄存器；
- (5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
- (6) 上电复位结束，CPU 开始执行指令；

5.3 外部复位

芯片的外部复位功能可通过配置字 RSTEN 开启，引脚设为外部复位脚即为开启外部复位功能，端口内部上拉电阻将自动使能。外部复位输入端口 RST 为施密特结构，低电平有效，即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 V_{LVR} 可通过配置字 LVRVS 选择。LVR 检测电路具有一定的回滞特性，回滞电压约为 6%（典型值），当电源电压下降至 V_{LVR} 时发生 LVR 复位，反之电源电压需上升至 $V_{LVR}+6\%$ 后 LVR 复位才解除。

5.5 看门狗复位

芯片的看门狗定时器（WDT）复位是一种对系统运行程序的保护机制。正常情况下，用户程序需定期对 WDT 执行清零操作，以避免 WDT 溢出。若发生异常情况，程序未及时清零 WDT，则芯片将因 WDT 溢出而产生看门狗复位，系统初始化后重新运行程序，从而返回受控状态。

注：低功耗模式下 CPU 暂停工作，若此时发生 WDT 溢出，则仅唤醒 CPU 而不复位芯片。



6 I/O 端口

6.1 通用 I/O 功能

芯片的输入/输出端口包括两组 7 位端口 P0、P1。所有端口均支持施密特输入，均支持推挽输出。除用作通用数字 I/O 端口外，部分端口还可复用为外部中断输入、键盘扫描输入/输出、或红外发射驱动输出等功能。

端口数据寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP0	-	P06D	P05D	P04D	P03D	P02D	P01D	P00D
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	X	X	X	X	X	X	X

BIT[6:0] **P0nD** – P0n 端口数据位 (n=6-0)

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
IOP1	-	P16D	P15D	P14D	P13D	P12D	P11D	P10D
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	X	X	X	X	X	X	X

BIT[6:0] **P1nD** – P1n 端口数据位 (n=6-0)

端口方向寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP0	-	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
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[6:0] **P0nOE** – P0n 端口输出使能位 (n=6-0)

0: 端口作为输入口，读端口操作将读取端口的电平状态；

1: 端口作为输出口，读端口操作将读取端口的数据位值；

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
OEP1	-	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
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[6:0] **P1nOE** – P1n 端口输出使能位 (n=6-0)

0: 端口作为输入口，读端口操作将读取端口的电平状态；

1: 端口作为输出口，读端口操作将读取端口的数据位值；



6.2 内部上/下拉电阻

所有端口均具有内部上拉和下拉电阻，且均可单独控制其上/下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/下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]
PUP0	-	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
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[6:0] **P0nPU** – P0n 端口上拉电阻控制位 (n=6-0)

- 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]
PUP1	-	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
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[6:0] **P1nPU** – P1n 端口上拉电阻控制位 (n=6-0)

- 0: 端口内部上拉电阻无效;
- 1: 端口内部上拉电阻有效;

下拉电阻控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP0	-	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
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[6:0] **P0nPD** – P0n 端口下拉电阻控制位 (n=6-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PDP1	-	P16PD	P15PD	P14PD	P13PD	P12PD	P11PD	P10PD
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[6:0] **P1nPD** – P1n 端口下拉电阻控制位 (n=6-0)

- 0: 端口内部下拉电阻无效;
- 1: 端口内部下拉电阻有效;



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 LIRC，WDT 溢出将复位芯片或唤醒 CPU。

可通过配置字 WDTM 设置 WDT 工作模式：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；选择低功耗模式下关闭，则 WDT 在休眠/HOLD 模式下自动关闭、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。

注：WDT 溢出时间为典型值，而实际值偏差较大，必须保证清 WDT 的间隔时间小于 WDT 溢出时间的 1/4。



8 键盘扫描电路 DKS

8.1 DKS 概述

芯片内置键盘扫描电路 DKS 模块，可通过寄存器位 DKSE 开启键盘扫描功能，在系统进入低功耗模式 CPU 暂停工作后，开启了键扫及唤醒功能的端口将自动输出键盘扫描波形，并可在输入电平下降沿时唤醒 CPU。所有端口均可通过端口键扫控制寄存器单独开启端口的键扫及唤醒功能，从而复用为键扫端口。

键扫端口支持三种键盘扫描模式：

静态键盘模式

系统进入低功耗模式后，所有键扫端口均自动固定为输入上拉状态（上拉电阻需先通过软件设为有效），当输入电平产生下降沿时，唤醒 CPU。静态键盘模式适用于键扫端口和其他通用端口（固定输出低电平）组成的矩型键盘电路。

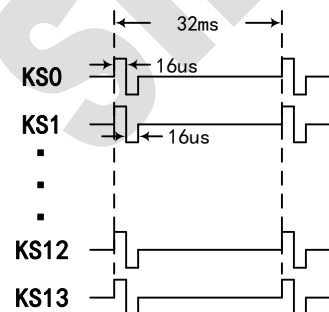
动态同步扫描模式

系统进入低功耗模式后，所有键扫端口每 32ms 同步输出 1 帧键扫波形（端口先输出 16 μ s 高电平再输出 16 μ s 低电平），其他时间端口均固定为高阻态。同步扫描模式适用于键扫端口和外部中断端口 INT0/INT1 组成的矩型键盘电路，通过外部中断输入电平的上升/下降沿触发中断，从而唤醒 CPU。

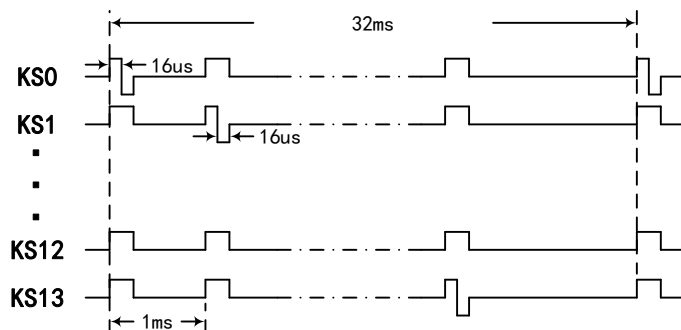
动态交替扫描模式

系统进入低功耗模式后，每个键扫端口交替输出 1 帧键扫波形（同时其他键扫端口切换为输入上拉状态，上拉电阻需先通过软件设为有效），其他时间均固定为高阻态。未开启键扫功能的端口不输出键扫波形且自动切换输入/输出状态，但在动态交替扫描时序中依然占用对应的扫描时间。交替扫描模式适用键扫端口组成的 T 型键盘电路，在键扫端口处于输入上拉状态时若输入电平产生下降沿则将唤醒 CPU。

DKSM=1/DKSS=0, 动态同步扫描模式



DKSM=1/DKSS=1, 动态交替扫描模式



端口 IROUT、VDD、GND 也可参与组成键盘电路。

可通过配置字 IRDKSET 选择支持 IROUT 外接按键，此时所有键扫端口的输入 SMT 阈值将变为



(VDD-0.5V)。若 IROUT 与其他键扫端口之间的按键按下，则在键扫端口电平变低时，IROUT 的输入电平将从 VDD 降为 (VDD-红外发射管压降)，从而唤醒 CPU。

可通过配置字 VDDDKSET 选择支持 VDD 外接按键，在键盘扫描时键扫端口的低电平成方式将由输出强低变为输入下拉，若 VDD 与其他键扫端口之间的按下，则键扫端口的电平将从低电平变为高电平，从而唤醒 CPU。

8.2 DKS 相关寄存器

键盘扫描控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
DKSCR	LSEL1	LSEL0	RSEL	IREN	DKSE	DKSM	DKSS	DKSI
R/W	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	0

BIT[7:6] LSEL[1:0] – P05 端口输出源电流选择位

LSEL[1:0]	P05 输出源电流
00	1mA
01	2mA
10	4mA
11	8mA

BIT[5] RSEL – 端口内部上拉电阻选择位

- 0: 端口内部上拉电阻为 150KΩ 或 60KΩ (由配置字 RUSEL 配置);
- 1: 端口内部上拉电阻为 10KΩ;

BIT[4] IREN – IROUT 端口红外发射驱动输出使能位

- 0: 端口为通用输入/推挽输出端口 (P06)，不支持红外发射驱动输出;
- 1: 端口为高阻/开漏输出端口 (IROUT)，支持红外发射增强灌电流输出;

BIT[3] DKSE – 键盘扫描功能使能位

- 0: 关闭 DKS 键盘扫描功能;
- 1: 开启 DKS 键盘扫描功能;

BIT[2,1] DKSM,DKSS – 键盘扫描模式选择位

DKSM:DKSS	键盘扫描模式
0X	静态键盘模式，键扫端口固定为输入上拉状态，支持输入电平下降沿唤醒
10	动态同步扫描模式，所有键扫端口同步输出 1 帧键扫波形
11	动态交替扫描模式，所有键扫端口交替输出 1 帧键扫波形

注: DKSE 和 DKSM 均为 1 时，系统执行 STOP 指令后 LIRC 将自动开启。



- BIT[0] **DKSI** – 键扫端口内部上拉电阻选择位
- 0: 键扫端口, 内部上拉电阻由寄存器位 RSEL 和配置字 RUSEL 决定;
 - 1: 键扫端口, 内部上拉电阻由配置字 RUSEL 决定, 不受寄存器位 RSEL 影响;

端口键扫控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
PODKCR	-	P06KSE	P05KSE	P04KSE	P03KSE	P02KSE	P01KSE	P00KSE
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[6:0] **P0nKSE** – P0n 端口键扫唤醒功能控制位 (n=6-0)
- 0: 关闭端口的键扫及唤醒功能;
 - 1: 开启端口的键扫及唤醒功能;

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
P1DKCR	-	P16KSE	P15KSE	P14KSE	P13KSE	P12KSE	P11KSE	P10KSE
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

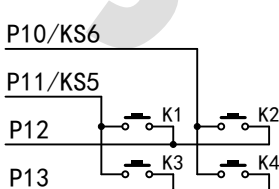
- BIT[6:0] **P1nKSE** – P1n 端口键扫唤醒功能控制位 (n=6-0)
- 0: 关闭端口的键扫及唤醒功能;
 - 1: 开启端口的键扫及唤醒功能;

注:

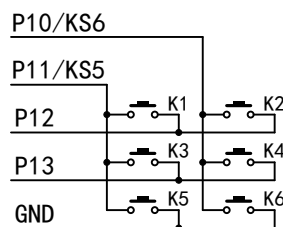
- 1、开启键扫唤醒功能的端口, 低功耗模式下将自动与键盘扫描周期同步切换输入/输出状态;
- 2、开启键扫唤醒功能的端口, 需先将端口内部上拉电阻设为有效;

8.3 DKS 应用示例

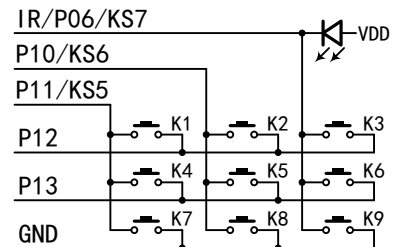
- (1) 静态键盘模式 (DKSM=0): 键扫端口内部上拉, 其他端口输出低。



①



②



③

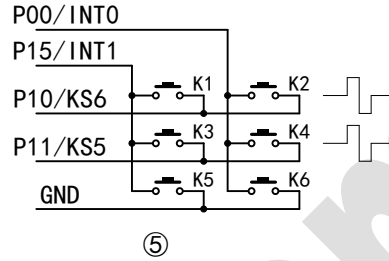
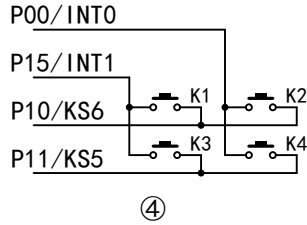
示例①②, P10/P11 设为键扫端口 (P10KSE/P11KSE=1), 并使能端口内部上拉电阻; P12/P13 用作通用 I/O 端口并输出低电平;

示例③, P10/P11/P06 设为键扫端口 (P10KSE/P11KSE/P06KSE=1), 并使能端口内部上拉电阻;

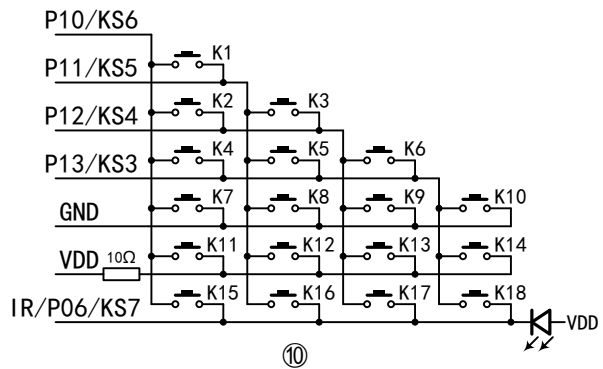
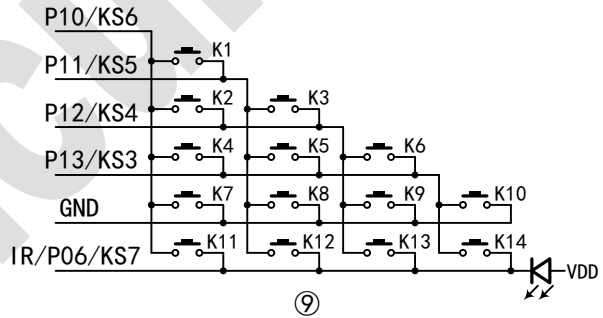
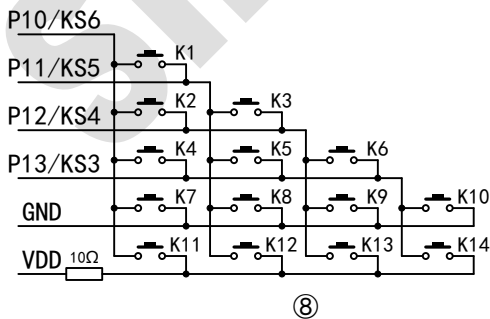
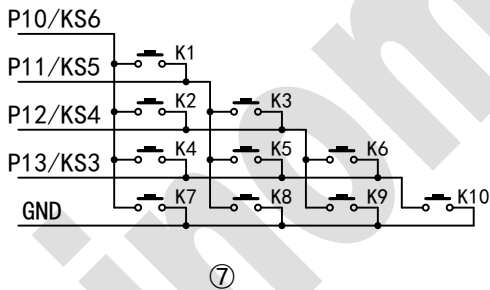
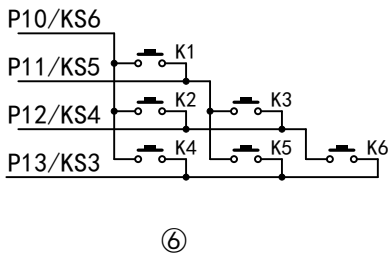


P12/P13 用作通用 I/O 端口并输出低电平；配置字 IRDKSET 配置为“IROUT 支持外接按键”，且建议去除按键 K9。

- (2) 动态同步扫描模式 (DKSM=1,DKSS=0)：外部中断 INT0/INT1 使能，选择下降沿触发，并开启内部上拉。



- (3) 动态交替扫描模式 (DKSM=1,DKSS=1)：





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 LVDVS 选择电压检测阈值。当 VDD 电压降至电压检测阈值以下时检测状态标志位 LVDF 将被置 1，同时中断标志 LVDIF 置 1 触发 LVD 中断；因 LVD 电路的回滞特性（回滞电压典型值为 6%），VDD 电压需恢复至电压检测阈值+6%后 LVDF 才被清 0（但中断标志不会自动清 0）。

注：开启 LVD 或切换电压检测阈值等操作，需待电路稳定（时间 > 2ms）后 LVD 输出才有效。

LVD 控制寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
LVDCR	LVDEN	-	-	-	LVDVS	-	-	LVDF
R/W	R/W	-	-	-	R/W	-	-	R
初始值	0	-	-	-	0	-	-	X

BIT[7] **LVDEN** – 低电压检测 LVD 使能位

0: 关闭 LVD;

1: 开启 LVD;

BIT[3] **LVDVS** – LVD 电压检测阈值选择位

0: LVD 电压检测阈值为 2.0V;

1: LVD 电压检测阈值为 2.4V;

BIT[0] **LVDF** – LVD 检测状态标志位

0: VDD 电压高于电压检测阈值，或 LVD 关闭;

1: VDD 电压低于电压检测阈值;



10 中断

芯片的中断源包括外部中断（INT0~INT1）和 LVD 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，自动将当前指令之后将要执行的下一条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，而是暂时挂起中断继续执行当前指令，完成后再处理中断。
- ◇ CPU 响应中断后，程序跳至中断入口地址（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，再执行 RETIE 指令以返回主程序。系统将自动恢复 GIE 为 1，然后从堆栈取出此前保存的 PC 值，CPU 从响应中断时正在执行指令的下一条指令的地址处开始继续运行。

注：应用外部中断功能或键盘中断功能，需将相应端口设为输入状态。

10.1 外部中断

芯片具有 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 GIE 为 1 且相应的外部中断使能位 INTnIE (n=0-1) 为 1，则产生外部中断。

10.2 LVD 中断

当 VDD 电压降至电压检测阈值以下时，将触发 LVD 中断，中断标志 LVDIF 将被置 1，若 GIE 为 1 且 LVD 中断使能位 LVDIE 为 1，则产生 LVD 中断。

10.3 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]
INTE	-	LVDIE	-	-	INT1IE	INT0IE	-	-
R/W	-	R/W	-	-	R/W	R/W	-	-



初始值	-	0	-	-	0	0	-	-
-----	---	---	---	---	---	---	---	---

BIT[6] **LVDIE** – LVD 中断使能位

- 0: 屏蔽 LVD 中断;
- 1: 使能 LVD 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;

中断标志寄存器

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
INTF	-	LVDIF	-	-	INT1IF	INT0IF	-	-
R/W	-	R/W	-	-	R/W	R/W	-	-
初始值	-	0	-	-	0	0	-	-

BIT[6] **LVDIF** – LVD 中断标志位

- 0: 未触发 LVD 中断;
- 1: 已触发 LVD 中断, 需软件清 0;

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 0;

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 0;



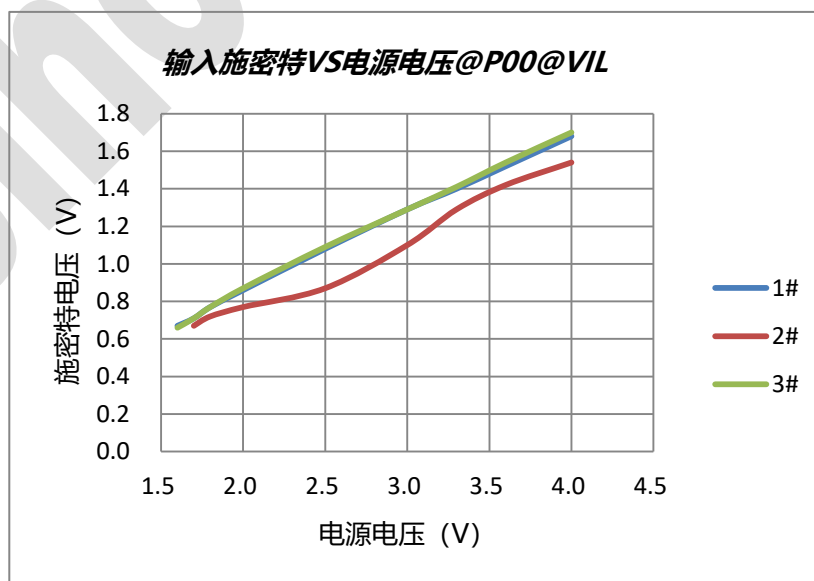
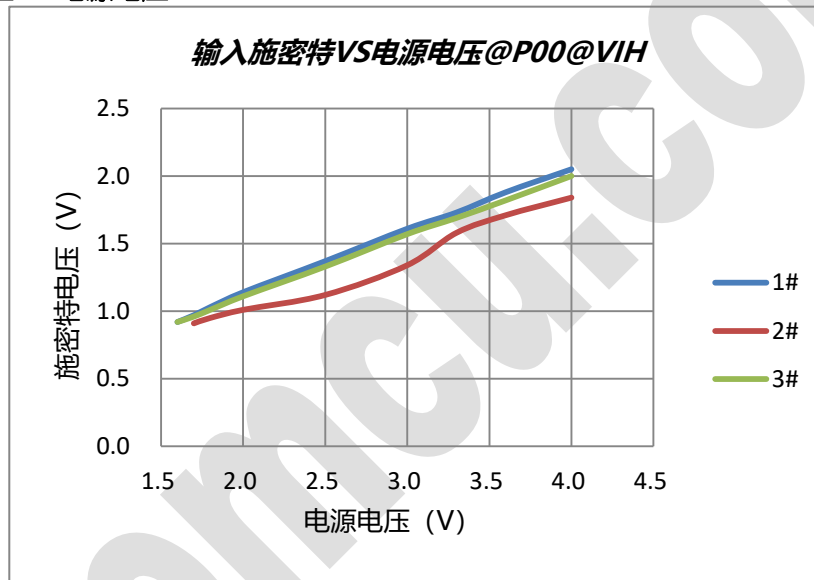
11 特性曲线

注:

- 1、特性曲线图中数据均源自抽样实测, 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、图文中若无特别说明, 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 $V_{DD}=3\text{V}$;

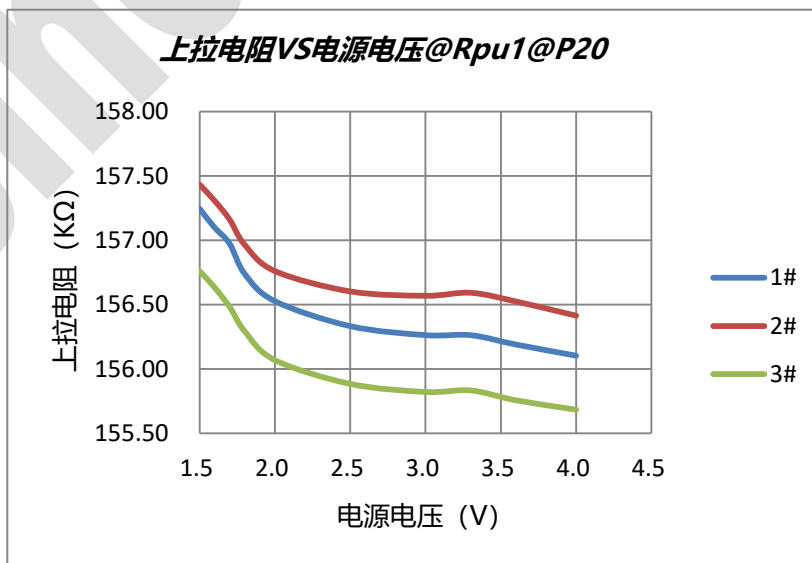
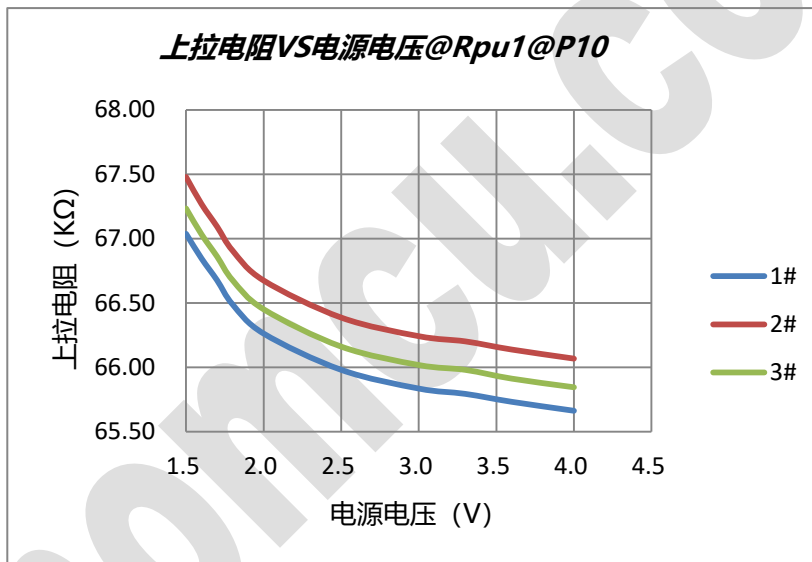
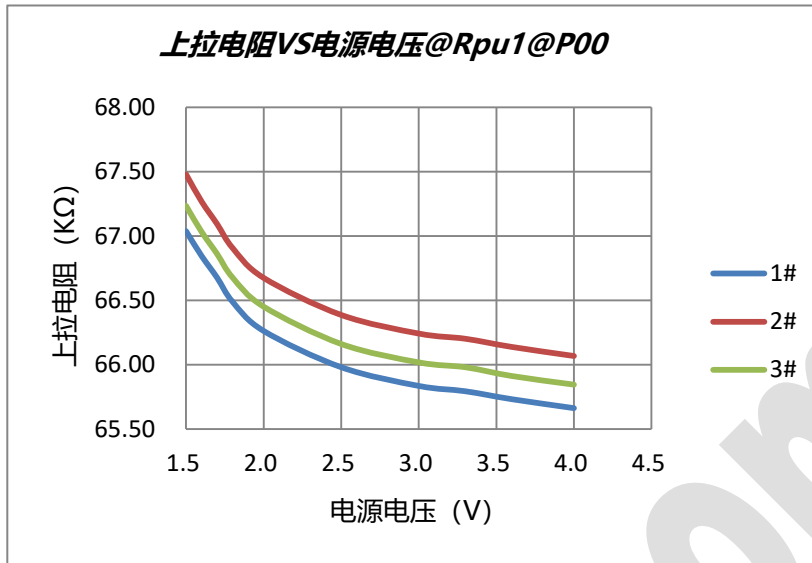
11.1 I/O 特性

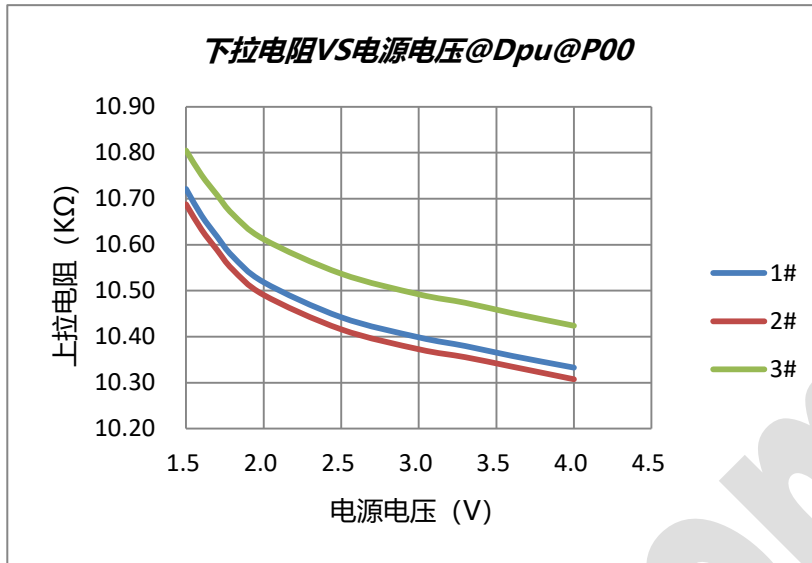
输入 SMT 阈值电压 VS 电源电压



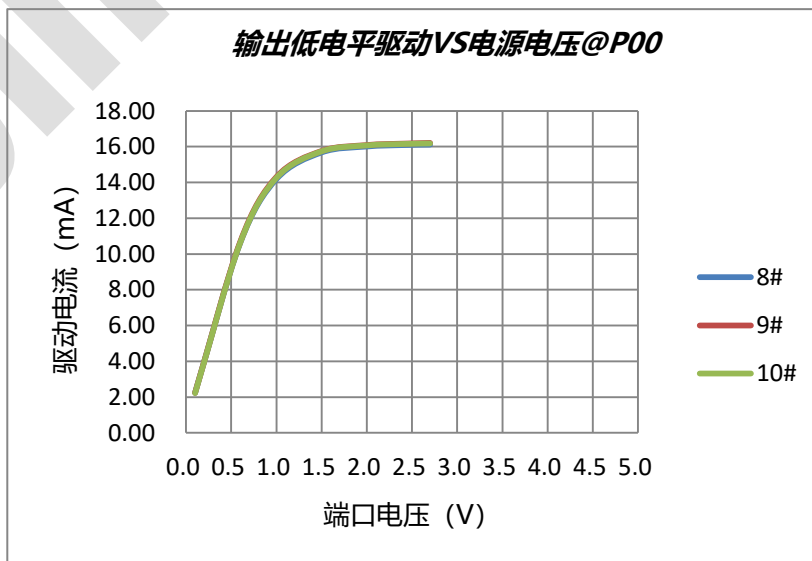
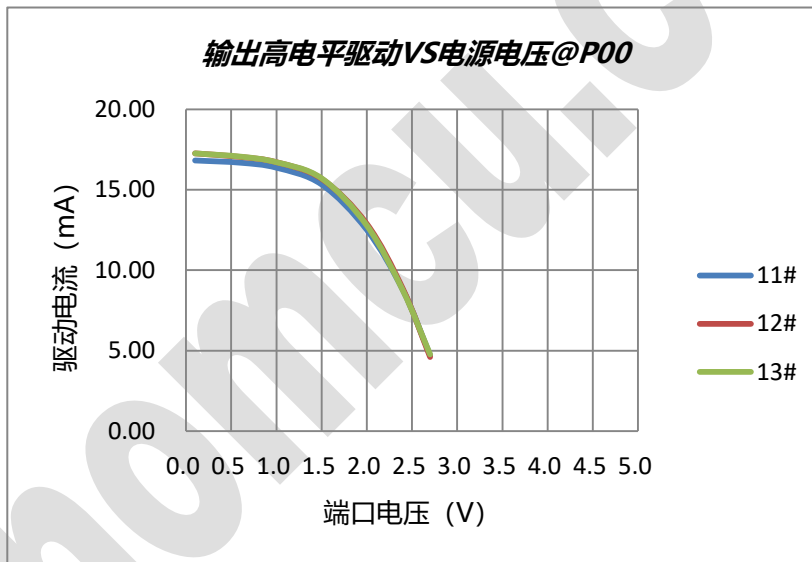


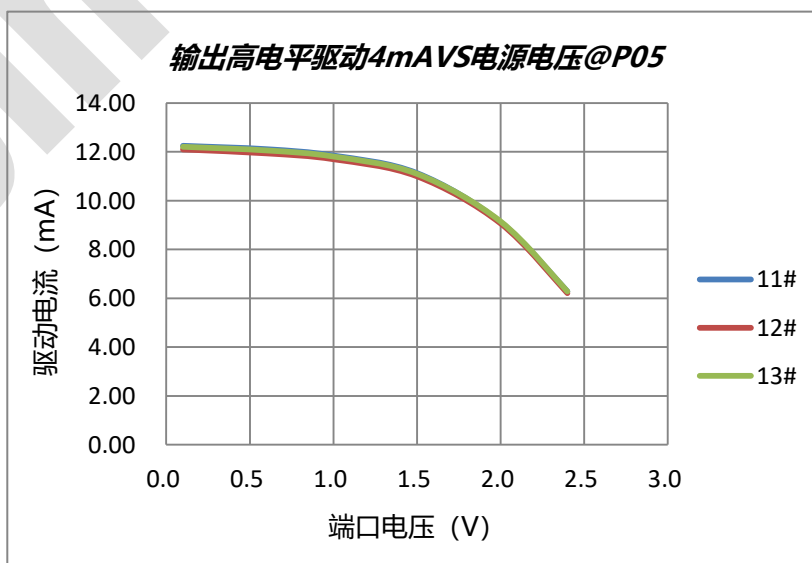
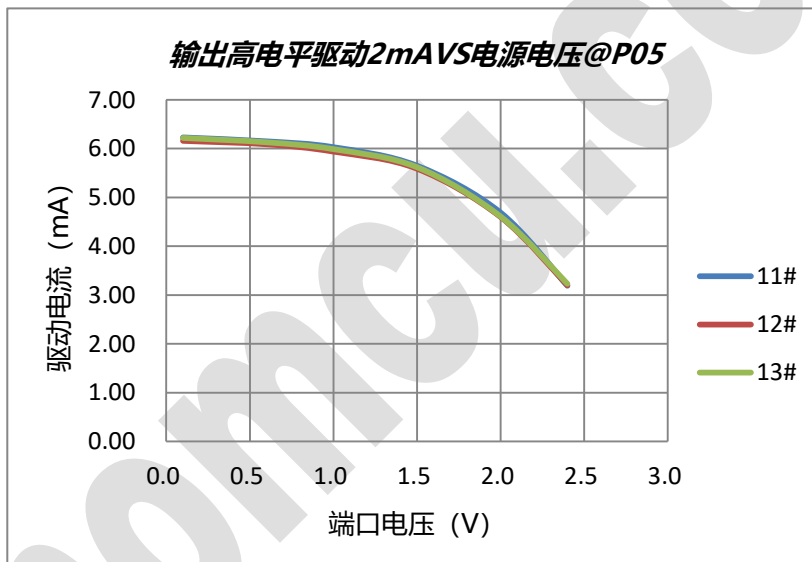
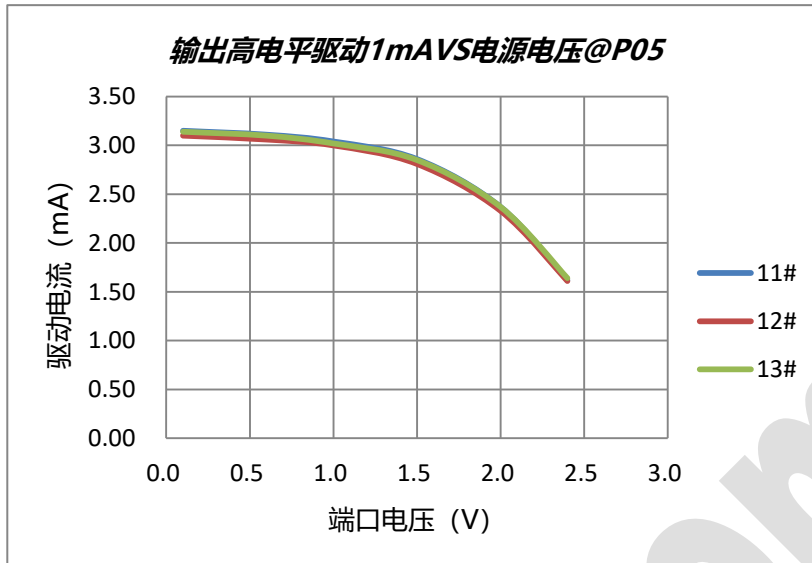
上/下拉电阻值 VS 电源电压

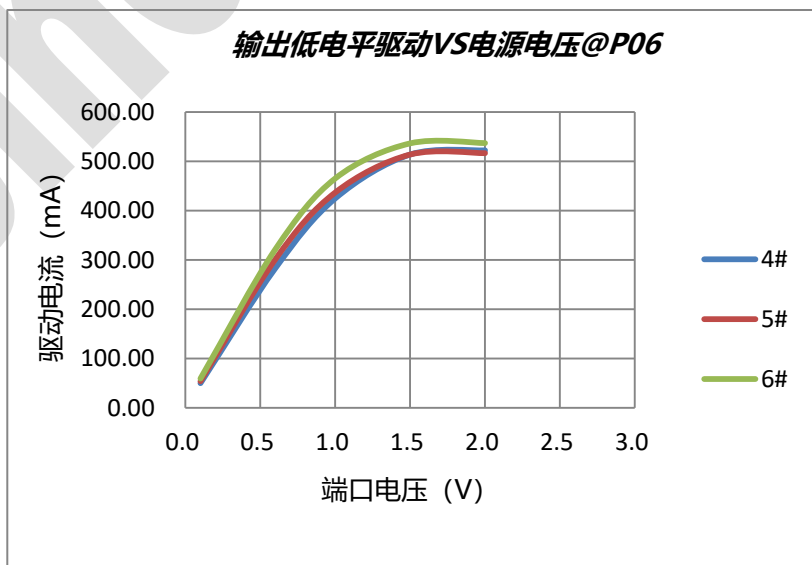
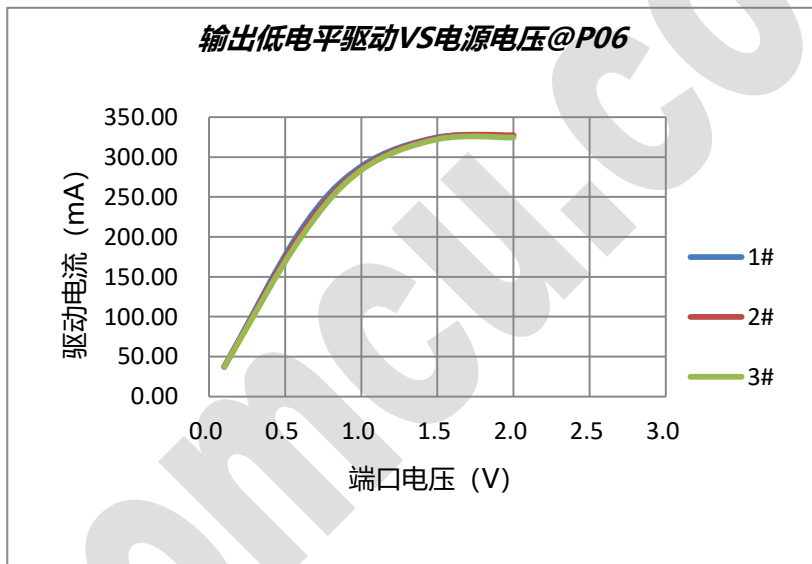
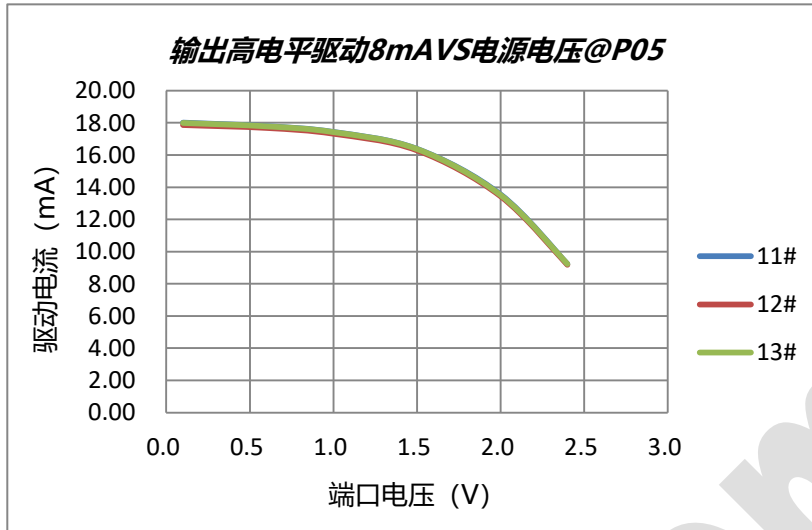


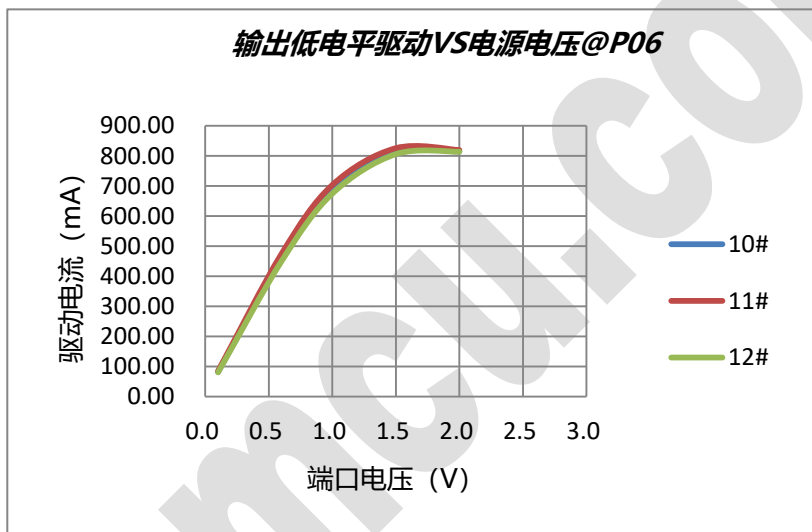
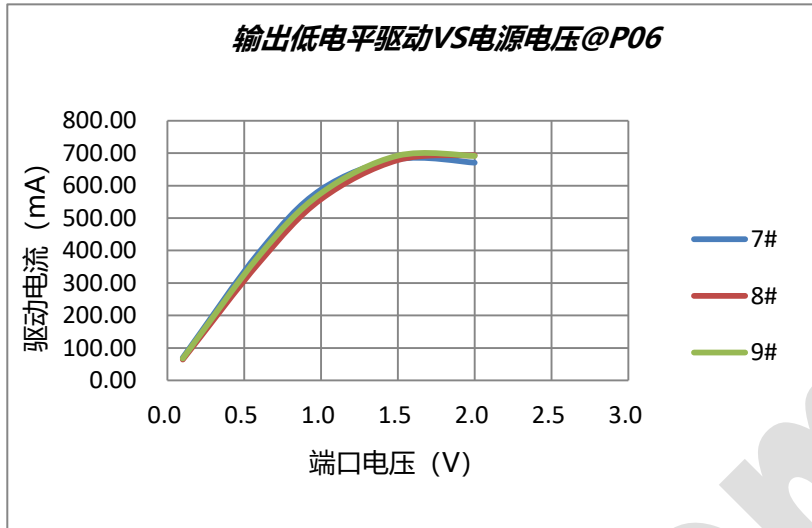


I/O 输出电流 VS 端口电压 (VDD=3V)



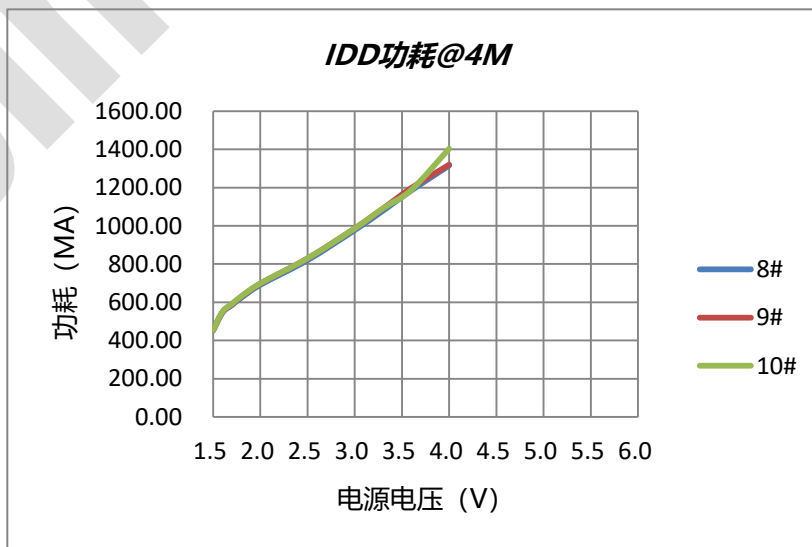


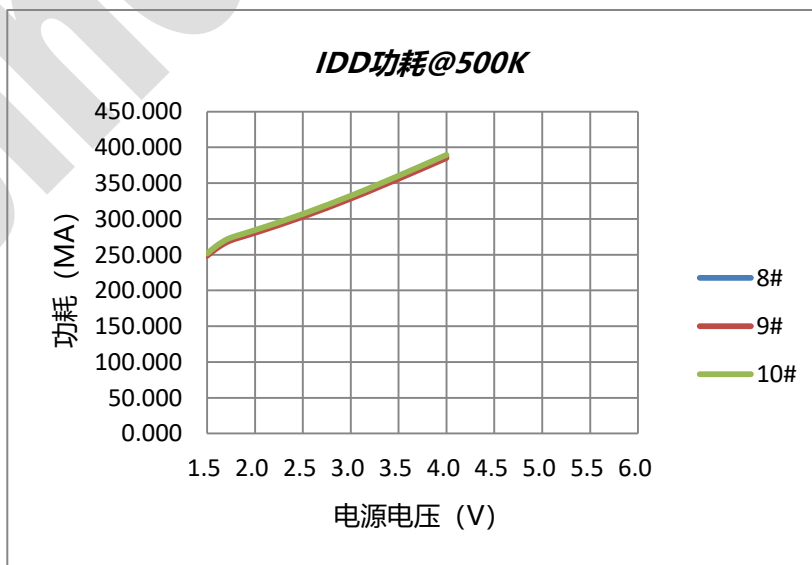
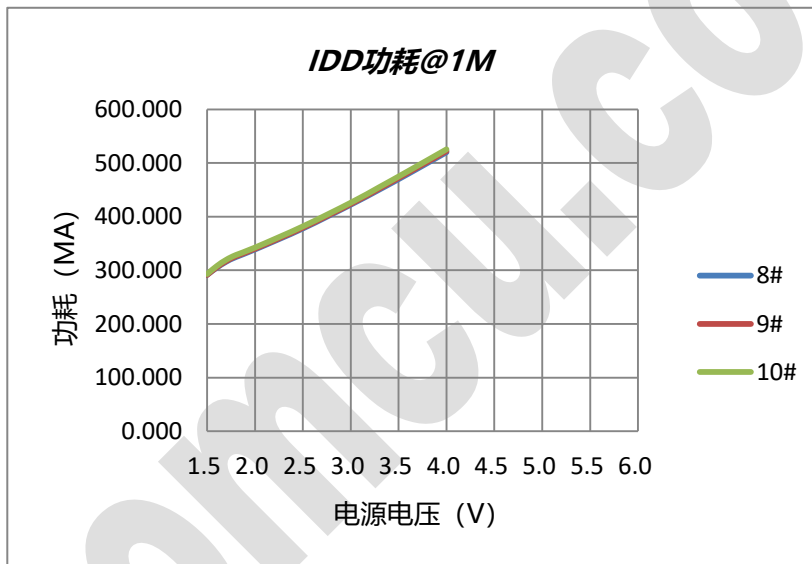
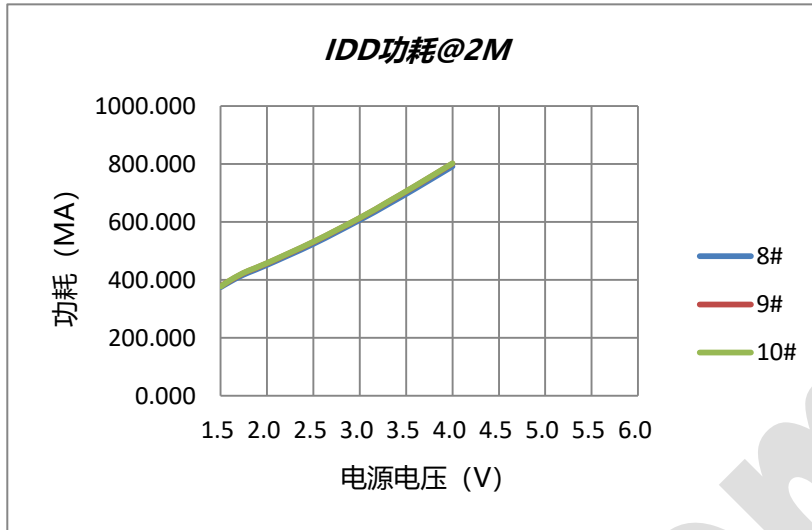




11.2 功耗特性

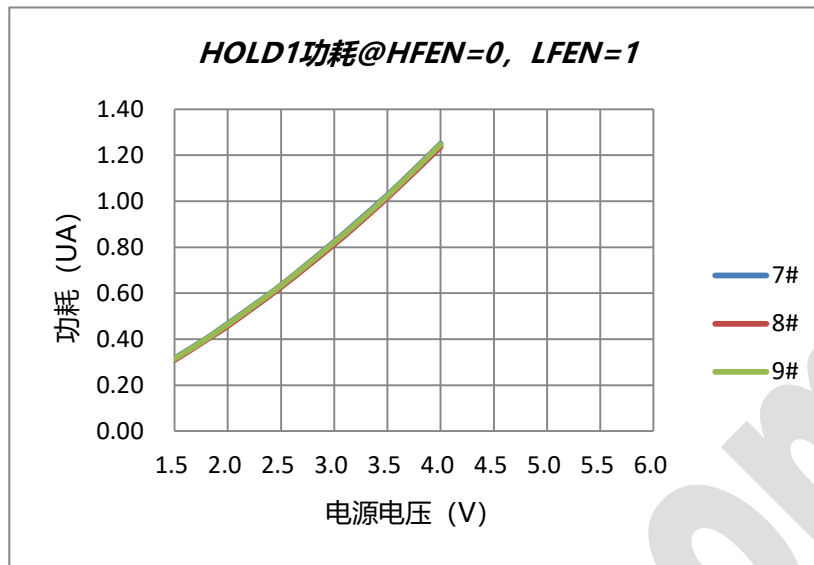
运行模式 功耗 VS 电源电压



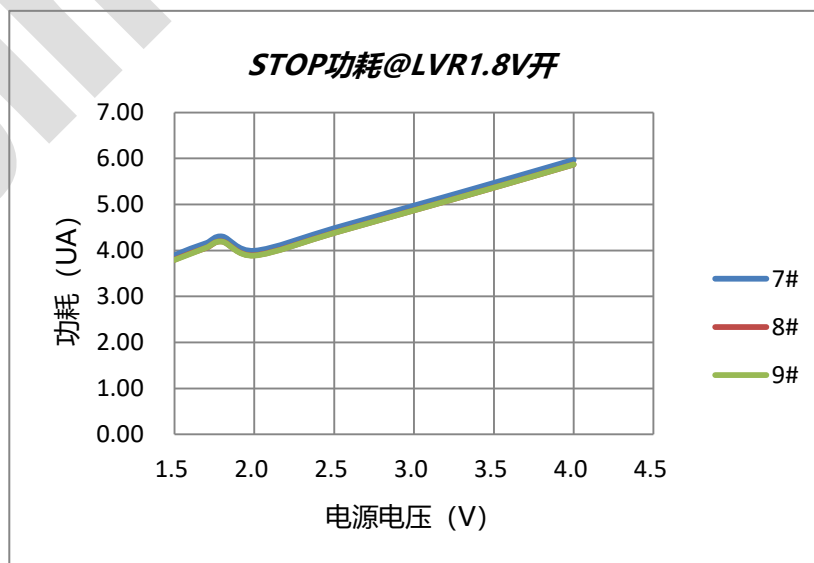
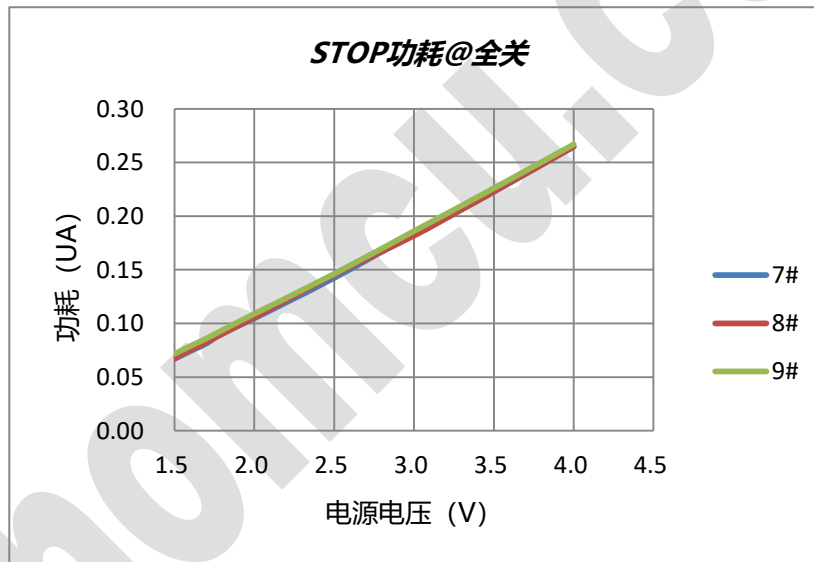


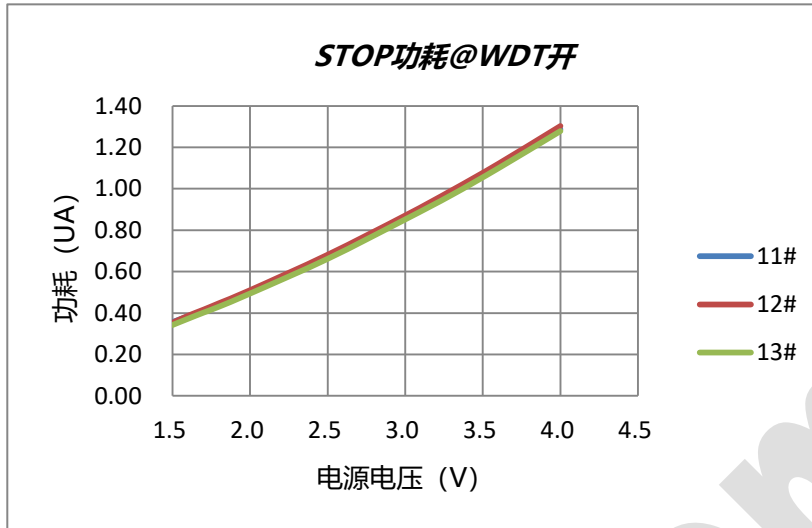


HOLD 模式 功耗 VS 电源电压

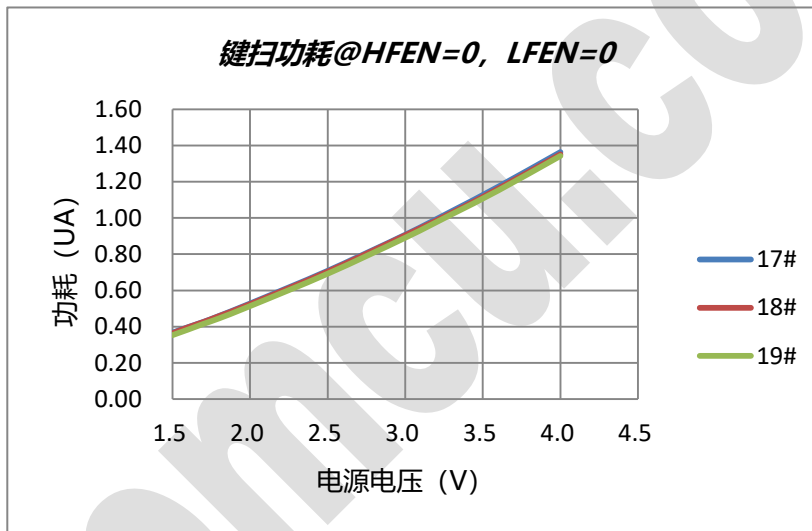


休眠模式 功耗 VS 电源电压



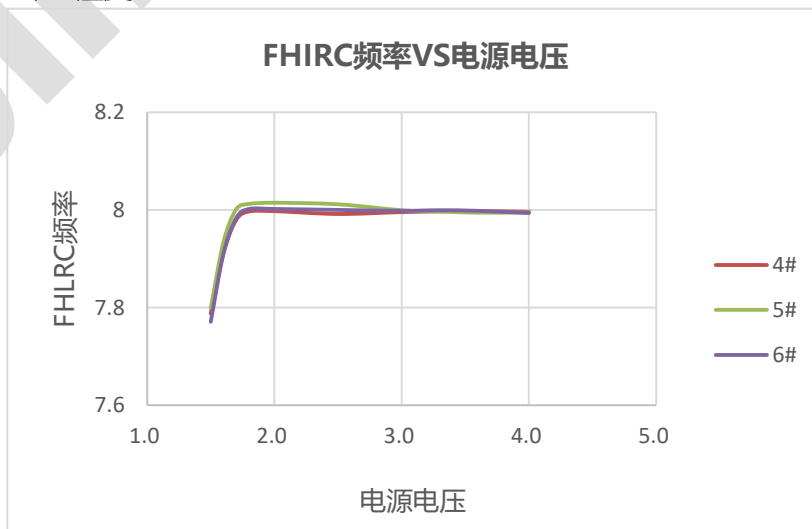


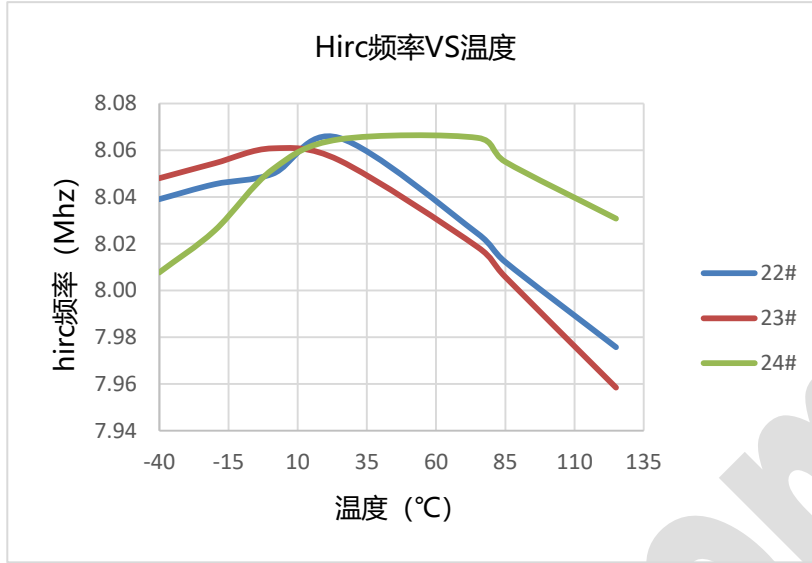
键盘扫描 功耗 VS 电源电压



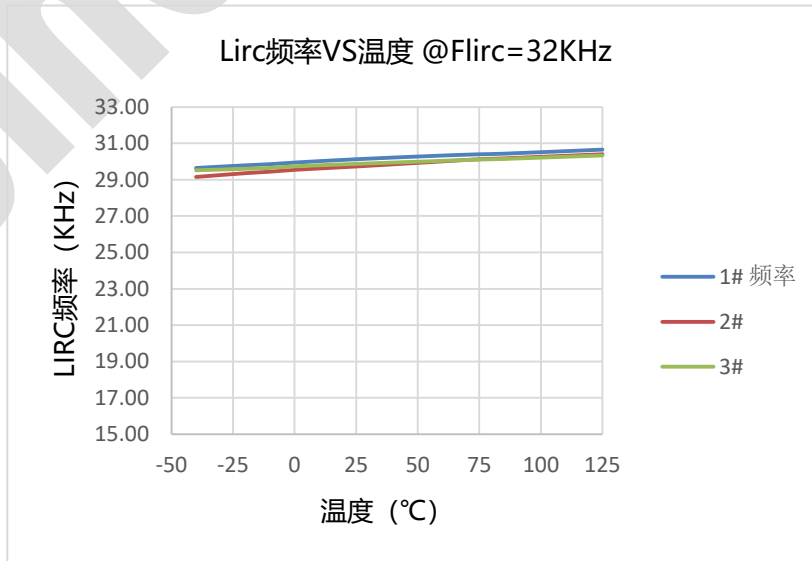
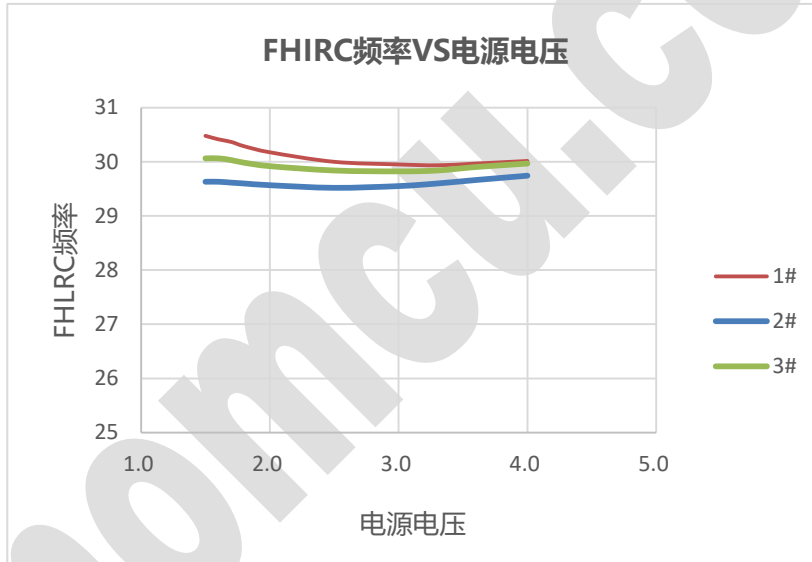
11.3 模拟电路特性

HIRC 频率 VS 电源电压/温度



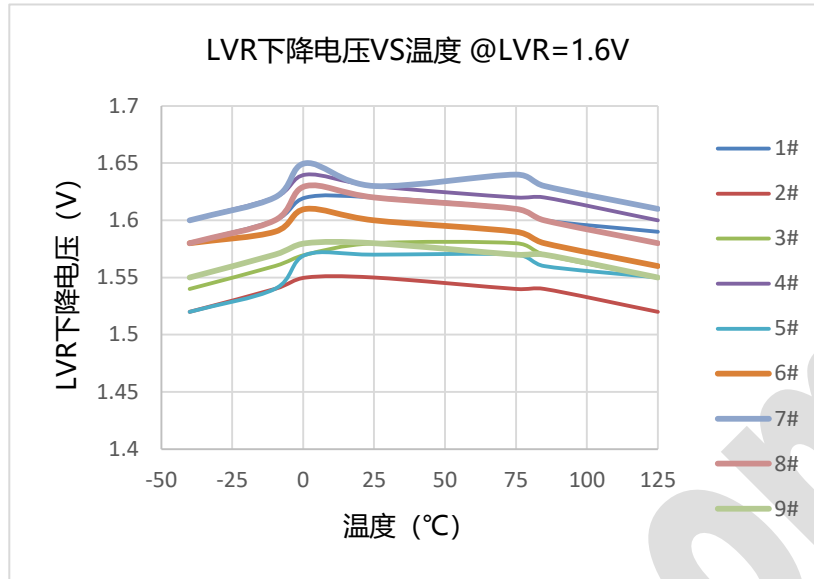


LIRC 频率 VS 电源电压/温度





LVR 阈值电压 VS 温度

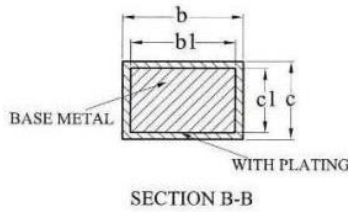
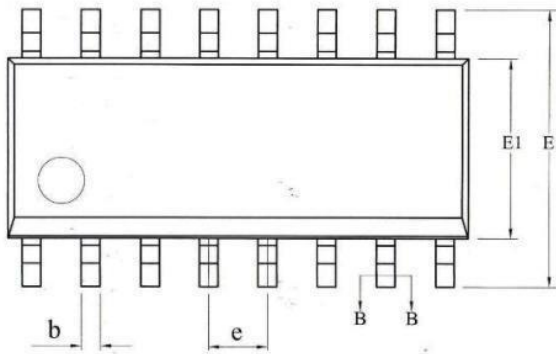
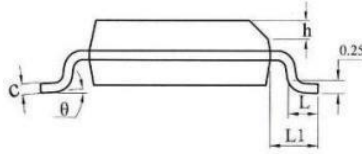
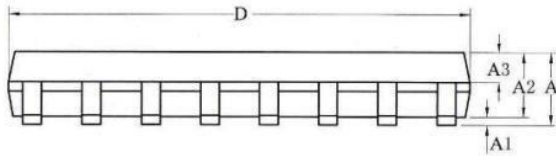


Shomcu.com



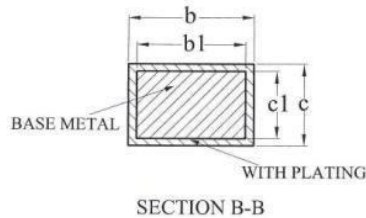
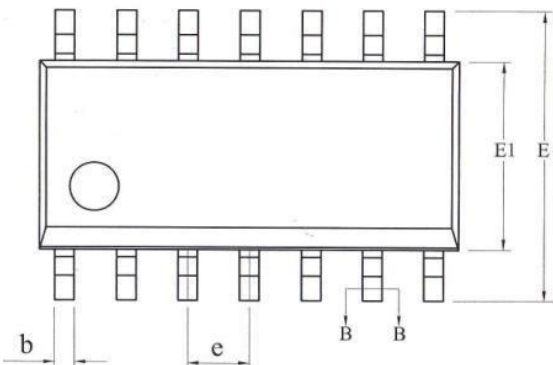
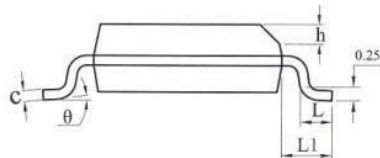
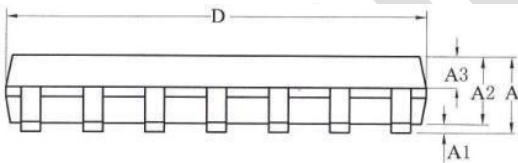
12 封装尺寸

12.1 SOP16



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	9.80	9.90	10.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
theta	0	—	8°

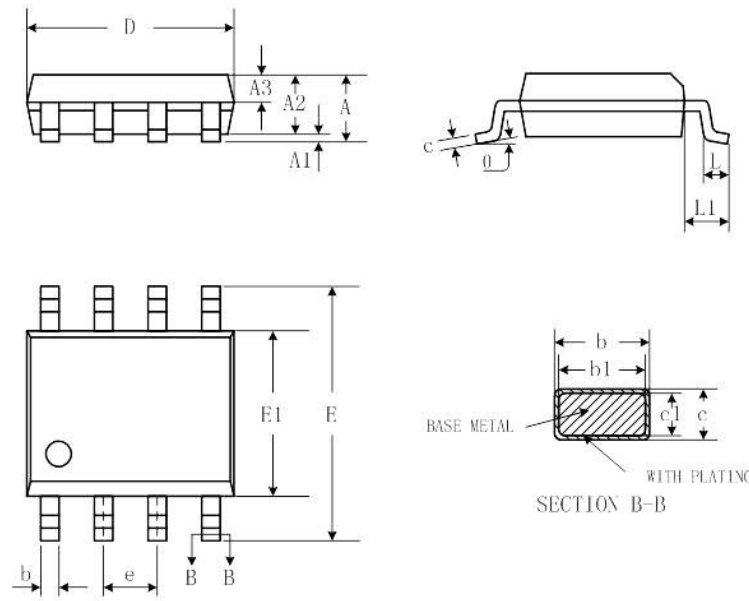
12.2 SOP14



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.05	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
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	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
theta	0	—	8°



12.3 SOP8



SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

Sinomicro



13 修订记录

版本	日期	修订内容
V1.0	2022-11-21	发布初版;
V1.1	2023-01-03	调整 SOP14 封装 A0J 脚位;
V1.2	2023-04-27	调整 WDTT 配置项为 4 档; 新增 LVR 配置项 1.4V, 并调整相关的工作电压范围; 调整 HIRC 频率常温条件下的精度为 $\pm 1.5\%$;