

*SinoMCU 8 位单片机*

**MC32P5222**

**用户手册**

V1.4



## 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 .....	5
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 .....	9
3	CPU 及存储器 .....	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.....	18
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	内部上拉电阻 .....	26
6.3	端口模式控制 .....	26
7	定时器 TIMER.....	28
7.1	看门狗定时器 WDT.....	28
7.2	定时器 T0.....	28
7.3	定时器 T1.....	31
8	键盘扫描电路.....	35
8.1	键盘扫描概述 .....	35
8.2	键盘扫描相关寄存器.....	35
8.3	键盘扫描操作步骤 .....	37
9	运算放大器 OPA.....	38
9.1	OPA 概述.....	38

9.2	OPA 相关寄存器 .....	38
9.3	OPA 操作步骤 .....	39
10	MTP 存储器 .....	40
10.1	MTP 概述 .....	40
10.2	MTP 相关寄存器 .....	40
10.3	MTP 操作示例 .....	42
10.4	MTP 数据烧录 .....	45
11	低电压检测 LVD .....	46
12	中断 .....	47
12.1	电平中断 .....	47
12.2	定时器中断 .....	47
12.3	LVD 中断 .....	48
12.4	MTP 中断 .....	48
12.5	中断相关寄存器 .....	48
13	特性曲线 .....	50
13.1	I/O 特性 .....	50
13.2	功耗特性 .....	53
13.3	模块特性 .....	57
14	封装尺寸 .....	66
14.1	SOP16 .....	66
15	修订记录 .....	67

## 1 产品概要

### 1.1 产品特性

- 8 位 CPU 内核
  - ◇ 精简指令集，8 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 2/4/8/16/32/64/128/256 分频
  - ◇ 系统低频时钟下 F<sub>CPU</sub> 固定为 2 分频
- 程序存储器
  - ◇ 4080×16 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
  - ◇ 432 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
  - ◇ 1K 字节 MTP 型数据存储器，擦写次数至少 1000 次
- 2 组共 14 个 I/O
  - ◇ P1 (P10~P17), P2 (P21/P22/P24~P27)
  - ◇ P27 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 VPP 输入
  - ◇ P17 为输入/开漏输出口，可复用为 IROUT 输入/输出，输出灌电流 4 级 (125mA/250mA/375mA/500mA) 可配置
  - ◇ 除 P17 外其余端口均内置上拉电阻，均可单独使能
  - ◇ 除 P17 外其余端口均支持键盘扫描及唤醒功能，并可单独使能
  - ◇ P21/P22 输出灌电流 4 级 (2mA/4mA/8mA/12mA) 可选
- 系统时钟源
  - ◇ 内置高频 RC 振荡器 (455KHz~16MHz)，可用作系统高频时钟源
  - ◇ 内置低频 RC 振荡器 (8KHz)，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：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/8192ms/16384ms
  - ◇ 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
  - ◇ 16 位定时器 T0，可实现输入信号的脉冲捕捉功能
  - ◇ 8 位定时器 T1，可实现外部计数和 PWM 功能
- 1 个运算放大器模块 OPA，可实现 IROUT 端口输入小信号的放大
- 中断
  - ◇ 电平中断 (INTS, INTT)
  - ◇ 定时器中断 (T0~T1)
  - ◇ LVD 中断, MTP 中断

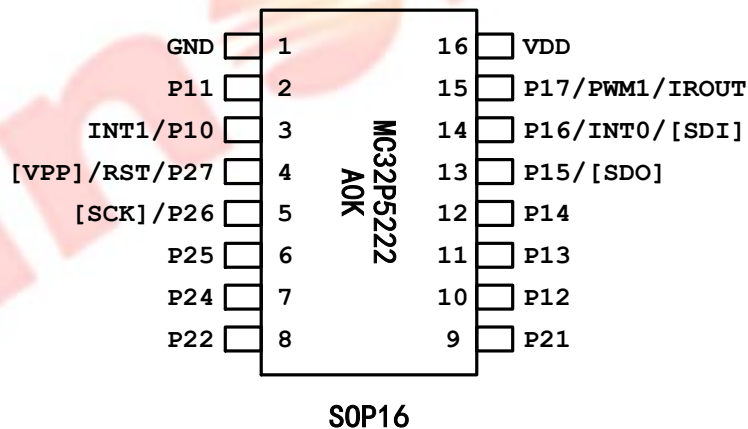
- 低电压复位 LVR
  - ◇ 关闭/1.4V/1.5V/1.6V/1.7V/1.8V/1.9V/2.0V/2.1V/2.2V/2.3V/2.4V/2.5V/2.6V/2.7V
- 低电压检测 LVD: 2.0V/2.4V
- 工作电压
  - ◇ 2.7V ~ 3.6V @ Fcpu = 0~8MHz
  - ◇ 2.0V ~ 3.6V @ Fcpu = 0~4MHz
  - ◇ 1.8V ~ 3.6V @ Fcpu = 0~2MHz
- 封装形式: SOP16

## 1.2 订购信息

产品名称	封装形式	备注
MC32P5222A0K	SOP16	

## 1.3 引脚排列

### MC32P5222A0K



## 1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P1, P2 (除 P17, P27)	D	GPIO (推挽输出), 内部上拉

P27	D	GPIO (开漏输出), 内部上拉
P17	D	GPIO (开漏输出)
INT0~INT1	DI	电平中断外部输入; 定时器 T0~T1 外部时钟输入
PWM1	DO	定时器 T1 的 PWM 输出
IROUT	D/A	脉冲接收/发送端口
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	-20~70	°C
储存温度	Tstg	-65~150	°C
焊接温度及时间	Tsld	260°C持续 10s	
流入 VDD 最大电流	IVDDmax	50	mA
流出 GND 最大电流 1 (除 P17)	IGNDmax1	50	mA
流过 GND 最大电流 2 (P17)	IGNDmax2	持续驱动: 200	mA
流过 GND 最大电流 3 (P17)	IGNDmax3	间歇驱动 (如每 1s 驱动 50ms): 600	mA

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=8MHz@FHIRC(16M)/2	2.7	3.0	3.6	V
			Fcpu=4MHz@FHIRC(16M)/4	2.0	3.0	3.6	
			Fcpu=2MHz@FHIRC(16M)/8	1.8	3.0	3.6	
			Fcpu=62.5KHz@FHIRC(16M)/256	1.8	3.0	3.6	
			Fcpu=4KHz@FLIRC/2	1.8	3.0	3.6	
输入漏电流	Ileak	所有输入脚	VDD=3V	-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=VDD-0.6V	4	8	16	mA
输出灌电流	Iol1	所有输出脚(除 P17,P21,P22)	Vol=0.6V	4	8	16	mA
	Iol2	P17	Vol=1.5V, DSEL=00	75	125	175	mA
			Vol=1.5V, DSEL=01	150	250	350	
			Vol=1.5V, DSEL=10	225	375	525	
			Vol=1.5V, DSEL=11	300	500	700	
	Iol3	P21, P22	Vol=0.6V, LSEL0,LSEL1=00	1	2	3	mA
			Vol=0.6V, LSEL0,LSEL1=01	2	4	6	
			Vol=0.6V, LSEL0,LSEL1=10	4	8	12	
			Vol=0.6V, LSEL0,LSEL1=11	6	12	18	

上拉电阻	Rpu1	P0, P1 (除 P17,P21,P22)	Vin=0, RSEL=0	30	60	120	KΩ
			Vin=0, RSEL=1	75	150	300	
	Rpu2	P21, P22	Vin=0, IRES0,IRES1=1, RSEL=0	30	60	120	KΩ
			Vin=0, IRES0,IRES1=1, RSEL=1	75	150	300	
			Vin=0, IRES0,IRES1=0	2.5	5	10	
	运行模式功耗	Irun	VDD	Fcpu=8MHz@FHIRC(16M)/2		2.0	
Fcpu=4MHz@FHIRC(16M)/4					1.3		mA
Fcpu=2MHz@FHIRC(16M)/8					1.1		mA
Fcpu=1MHz@FHIRC(16M)/16					0.7		mA
Fcpu=4KHz@FLIRC/2, 开启低功耗					10	20	uA
HOLD1 功耗	Ihold1	VDD	CPU 停, HIRC(16M)开		450		uA
			CPU 停, HIRC(8M)开		250		uA
			CPU 停, HIRC(4M)开		180		uA
			CPU 停, HIRC(2M)开		130		uA
			CPU 停, HIRC(1M)开		100		uA
			CPU 停, HIRC(455K)开		90		uA
HOLD2 功耗	Ihold2	VDD	CPU 停, HIRC 关, LIRC 开, LVR 关		1	3	uA
休眠模式功耗	Istop	VDD	休眠模式, WDT/LVR 关		0.1	1	uA
			休眠模式, WDT 关, LVR 开		1	3	uA
低压检测电压	VLVD	VDD		2.3	2.4	2.5	V
				1.9	2.0	2.1	
上电复位电压	VPOR	VDD		1.0	1.2	1.4	V
低压复位电压	VLVR	VDD	LVRVS 配置	2.55	2.7	2.85	V
			LVRVS 配置	2.45	2.6	2.75	
			LVRVS 配置	2.35	2.5	2.65	
			LVRVS 配置	2.3	2.4	2.5	
			LVRVS 配置	2.2	2.3	2.4	
			LVRVS 配置	2.1	2.2	2.3	
			LVRVS 配置	2.0	2.1	2.2	
			LVRVS 配置	1.9	2.0	2.1	
			LVRVS 配置	1.8	1.9	2.0	
			LVRVS 配置	1.7	1.8	1.9	
			LVRVS 配置	1.6	1.7	1.8	
			LVRVS 配置	1.5	1.6	1.7	
			LVRVS 配置	1.4	1.5	1.6	
			LVRVS 配置	1.3	1.4	1.5	
匹配 Fcpu 的最低 LVR 电压		VDD	Fcpu=8MHz	2.6			V
			Fcpu=4MHz	1.8			
			Fcpu=2MHz	1.4			
			Fcpu=1MHz	LVR 关闭			

注：条件项中，未注明模块默认关闭，无关端口状态为输出无负载，输入或开漏输出高则端口电压为 VDD/GND。



### 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	FHIRC1	VDD=3V, T=25°C	-1%	16	+1%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-3%		+3%	
	FHIRC2	VDD=3V, T=25°C	-1%	8	+1%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-8%		+8%	
	FHIRC3	VDD=3V, T=25°C	-1%	4	+1%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-10%		+10%	
	FHIRC4	VDD=3V, T=25°C	-2%	2	+2%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-20%		+20%	
	FHIRC5	VDD=3V, T=25°C	-2%	1	+2%	MHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-20%		+20%	
	FHIRC6	VDD=3V, T=25°C	-2%	455	+2%	KHz
		VDD=1.8V~3.6V, T=-20°C~70°C	-20%		+20%	
LIRC 振荡频率	FLIRC	VDD=3V, T=25°C	-50%	8	+50%	KHz

### 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
ADCRA 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		清零 ACC	0→ACC	1	1	Z
CLRR	R	清零 R	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	清零看门狗定时器	0→WDT	1	1	TO,PD
STOP	进入低功耗模式	0→WDT; CPU 暂停	1	1	TO,PD

注:

1. ACC-算术逻辑单元累加器, 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, 4080×16 位的地址空间范围为 0000H~0EFFH。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 - 0EFFH)

程序存储器支持间接寻址, 可通过寄存器 INDF3 访问 FSR1×256+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。

数据存储器还包括 MTP 存储器，需通过特殊寄存器 SFR 进行读写操作。

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-1AFH	GPR	通用数据存储器区							
1B0H-1B7H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
1B8H-1BFH		MCR	INDF3	INTE	INTF	OSCM	LVDCR		
1C0H-1C7H		MTPADH	MTPADL	MTPDB	MTPMD	MTPCTR			
1C8H-1CFH		IOP1	OEP1	PUP1				DKWP1	
1D0H-1D7H		IOP2	OEP2	PUP2				DKWP2	
1D8H-1DFH		T0CR	T0LOADH	T0LOADL	T0LATRH	T0LATRL	T0LATFH	T0LATFL	
1E0H-1E7H		T1CR	T1DATA	T1LOAD		OPCR0	OPCR1	DKW0	DKW1
1E8H-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

8 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

#### 数据指针寄存器

	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。

	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。

**间接寻址寄存器**

	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。

	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。

	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。

	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。

### 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:  $PC = PC + 1$ ;
- ◇ 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/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
MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
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] **MINT1[1:0]** – 电平中断 INTT (INT1) 触发方式选择位

MINT1[1:0]	INTT (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	

BIT[1:0] **MINT0[1:0]** – 电平中断 INTS (INT0) 触发方式选择位

MINT0[1:0]	INTS (INT0) 触发方式
00	上升沿触发
01	下降沿触发
10	电平变化触发
11	

### 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中，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。

芯片的用户配置字，定义如下：

符号	功能说明
<b>HIRCFS</b>	HIRC 振荡频率选择： FHIRC=16MHz; 8MHz; 4MHz; 2MHz; 1MHz; 455KHz;
<b>FCPUS</b>	高频时钟下 FCPU 分频选择： FCPU=FHOSC/2; FCPU=FHOSC/4; FCPU=FHOSC/8; FCPU=FHOSC/16; FCPU=FHOSC/32; FCPU=FHOSC/64; FCPU=FHOSC/128; FCPU=FHOSC/256;
<b>RSTEN</b>	RST 外部复位设置： P27 为外部复位脚; P27 为输入/输出脚;
<b>LVRVS</b>	LVR 复位电压选择: (LVR 电压应满足由 FCPU 决定的工作电压特性) LVR 关闭; 1.4V; 1.5V; 1.6V; 1.7V; 1.8V; 1.9V; 2.0V; 2.1V; 2.2V; 2.3V; 2.4V; 2.5V; 2.6V; 2.7V;
<b>LVRMD</b>	LVR 模式设置： LVR 始终开启; LVR 在运行模式下开启，在低功耗模式下关闭;
<b>WDTM</b>	WDT 模式设置： WDT 始终关闭; WDT 在低功耗模式下关闭; WDT 始终开启;
<b>WDTT</b>	WDT 溢出时间选择： 上电延时=WDT 溢出时间=16ms; 上电延时=WDT 溢出时间=64ms; 上电延时=WDT 溢出时间=256ms; 上电延时=WDT 溢出时间=1024ms; 上电延时=16ms, WDT 溢出时间=2048ms; 上电延时=64ms, WDT 溢出时间=4096ms; 上电延时=256ms, WDT 溢出时间=8192ms; 上电延时=1024ms, WDT 溢出时间=16384ms;
<b>DSEL</b>	IROUT 输出灌电流选择： 125mA; 250mA; 375mA; 500mA;
<b>ROTP</b>	OTP 低功耗模式设置： 关闭 OTP 低功耗模式; 开启 OTP 低功耗模式;
<b>ENCR</b>	程序代码加密设置： 程序代码加密; 程序代码不加密;

## 4 系统时钟

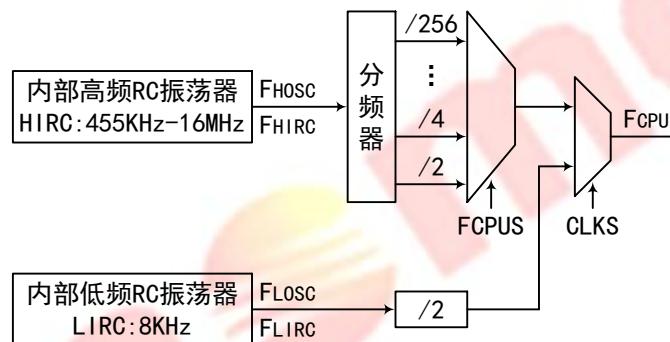
芯片为双时钟系统，内部电路均在系统高频时钟  $F_{HOSC}$  或系统低频时钟  $F_{LOSC}$  下工作，部分模块的时钟还可在  $F_{HOSC}$  和  $F_{LOSC}$  之间切换。

系统高频时钟  $F_{HOSC}$  固定为内部高频 RC 振荡器  $HIRC$  (455KHz~16MHz) 时钟  $F_{HIRC}$ ；系统低频时钟  $F_{LOSC}$  固定为内部低频 RC 振荡器  $LIRC$  (8KHz) 时钟  $F_{LIRC}$ 。

CPU 为双时钟，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 个振荡频率可通过配置字  $HIRCFS$  选择 (16MHz/8MHz/4MHz/2MHz/1MHz/455KHz) 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 8KHz 的  $LIRC$  振荡器，可用作系统低频时钟源，用于键盘扫描电路，也用于系统上电延时控制、WDT 定时器等电路。

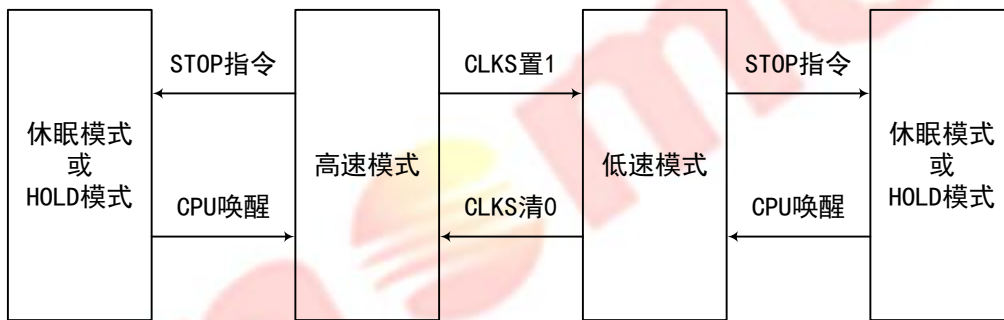
### 4.3 系统工作模式

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。

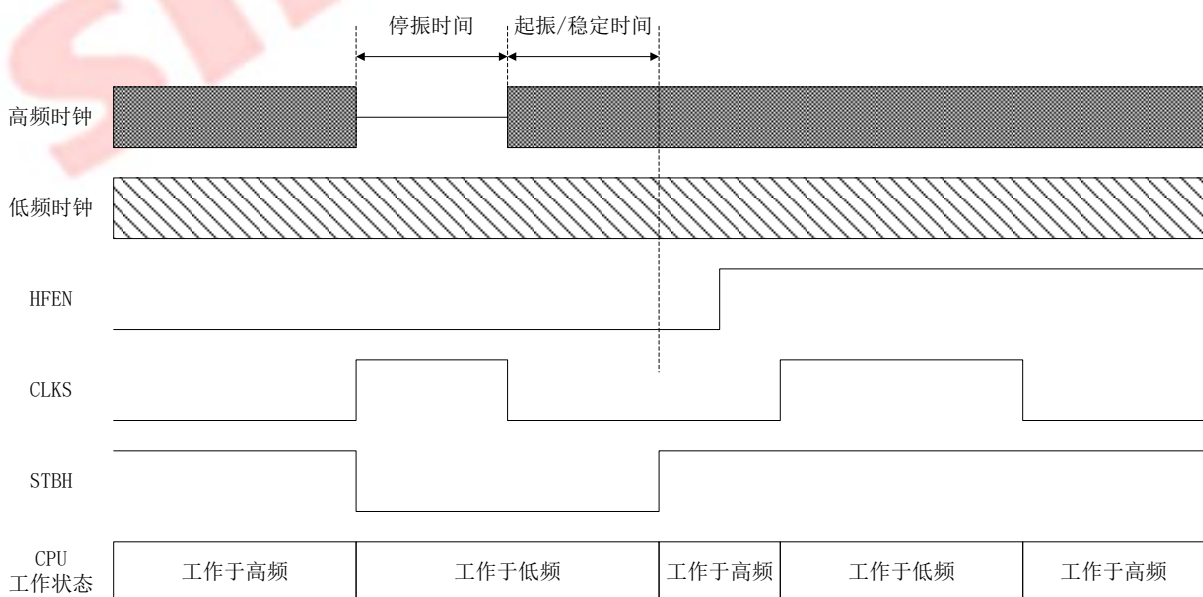
工作模式	切入条件	系统状态
高速	任意模式下，系统复位	CPU 高速运行，高/低频时钟源均工作
	低速模式下，CLKS 清 0	
	HOLD1/HOLD2/休眠模式下，CPU 唤醒	
低速	高速模式下，CLKS 置 1	CPU 低速运行，高频时钟源由 HFEN 决定
	HOLD1/HOLD2/休眠模式下，CPU 唤醒	
HOLD1	高/低速模式下，HFEN 置 1，执行 STOP 指令	CPU 暂停，高频时钟源工作，低频时钟源由 LFEN 决定
HOLD2	高/低速模式下，HFEN 清 0，LFEN 置 1，执行 STOP 指令	CPU 暂停，高频时钟源停止，低频时钟源工作
休眠	高/低速模式下，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>OSCM</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）；
- ◇ 键盘扫描端口有电平变化发生（若有键盘扫描功能）；

系统退出低功耗模式后，先经过 4 个低频周期和振荡等待（内部高频等待 16 个周期，内部低频等待 16 个周期），然后 CPU 开始工作。

**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态，以免因引脚浮空而产生漏电流或非预期的中断唤醒；

## 5 复位

### 5.1 复位条件

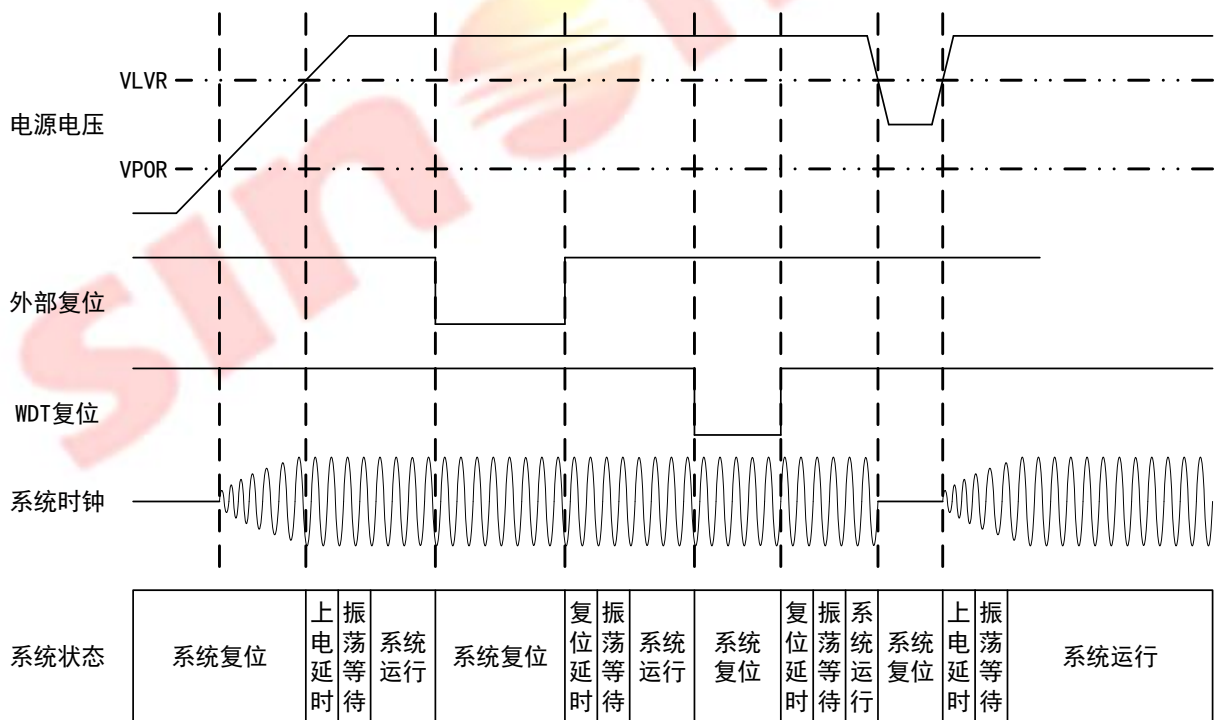
芯片共有如下几种复位方式：

- ◇ 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 开始重新从程序存储器 0000H 地址处执行指令。

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使能。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配置字 **LVRVS** 选择。电压检测电路有一定的回滞特性，回滞电压为 0.1V 左右（典型值），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 P1 和一组 6 位端口 P2。所有 I/O 端口均支持施密特输入，除 P17/P27 为开漏输出外其余 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、键盘扫描输入/输出、或红外遥控信号输入/输出等复用功能。

#### 端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
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] P1nD – P1n 端口数据位 (n=7-0)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP2	P27D	P26D	P25D	P24D	-	P22D	P21D	-
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	-
初始值	X	X	X	X	-	X	X	-

BIT[7:4,2,1] P2nD – P2n 端口数据位 (n=7-4,2,1)

**注：开漏输出口输出高电平时，若端口电压低于 VDD 电压则会产生漏电流。**

#### 端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP1	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
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] P1nOE – P1n 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；

**注：**

1. P17 为输入/开漏输出口，输出灌电流可通过用户配置字 DSEL 设置；
2. P17 输出低电平后不能直接转为输入状态，需先输出高电平至少 10us 后才能转为输入状态；
3. P17 在进入低功耗模式前必须设置为输入状态（需按上述 2 操作），否则将产生静态功耗；
4. P17 在 OPA 开启后自动转换为 OPA 模拟输入端口，相应控制位无效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP2	P27OE	P26OE	P25OE	P24OE	-	P22OE	P21OE	-

R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	-
初始值	0	0	0	0	-	0	0	-

- BIT[7:4,2,1] **P2nOE** – P2n 端口输出使能位 (n=7-4,2,1)
- 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

除 P17 外其余端口均有内部上拉电阻, 且均有单独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>PUP1</b>	-	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
<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[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;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>PUP2</b>	P27PU	P26PU	P25PU	P24PU	-	P22PU	P21PU	-
<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[7:4,2,1] **P2nPU** – P2n 端口上拉电阻控制位 (n=7-4,2,1)
- 0: 端口内部上拉电阻无效;
  - 1: 端口内部上拉电阻有效;

## 6.3 端口模式控制

除 P17 外其余端口均可复用为 T 型键盘扫描端口, 可通过寄存器设置端口的工作模式。

### 键盘端口模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>DKWP1</b>	-	P16DKW	P15DKW	P14DKW	P13DKW	P12DKW	P11DKW	P10DKW
<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[6:0] **P1nDKW** – P1n 端口键盘模式控制位 (n=6-0)

- 0: 端口为通用 I/O 模式;
- 1: 端口为键盘扫描模式;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>DKWP2</b>	P25DKW	P24DKW	P25DKW	P24DKW	-	P22DKW	P21DKW	-
<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[7:4,2,1] **P2nDKW** – P2n 端口键盘模式控制位 (n=7-4,2,1)

- 0: 端口为通用 I/O 模式;
- 1: 端口为键盘扫描模式;

**注:**

1. 休眠模式下, 设置为键盘扫描模式的端口, 将按照寄存器位 *WSEL* 选择的时间周期性的切换输入/输出状态;
2. 设置为键盘扫描模式的端口, 必须同时设置上拉电阻有效;

## 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，WDT 计数器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。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。

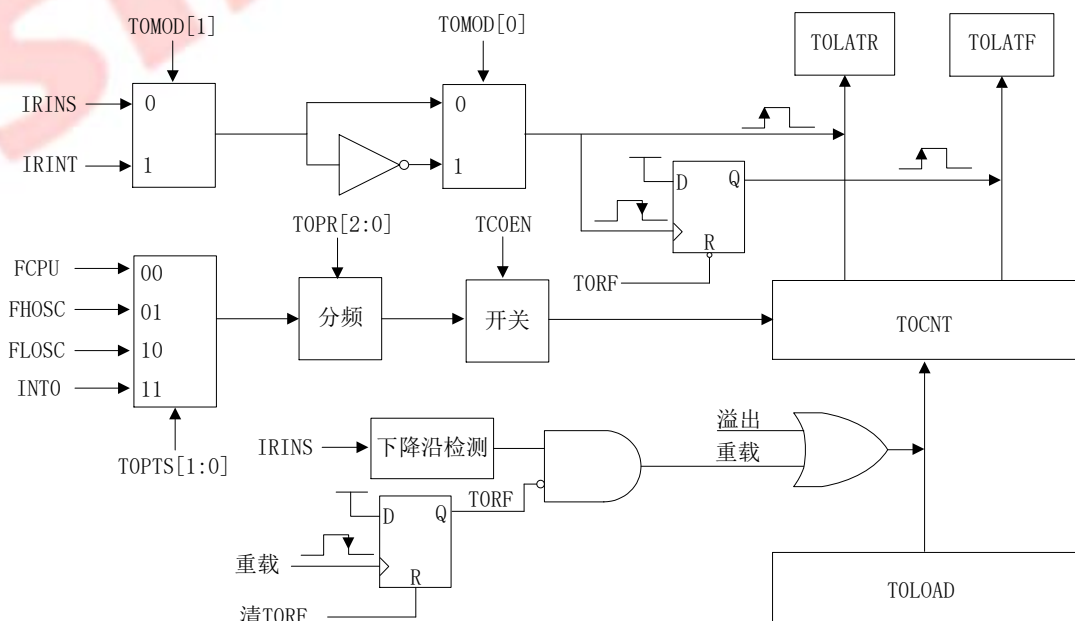
WDT 溢出时间可配置为 16ms/64ms/256ms/1024ms/2048ms/4096ms/8192ms/16384ms。

**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 16 位定时/计数器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器及 2 个 16 位捕捉寄存器。

- ◇ 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；
- ◇ 支持脉冲捕捉功能，可通过捕捉寄存器获得上升/下降沿时间；
- ◇ 可设置 IRINS 有效沿触发重载；
- ◇ 支持溢出中断和溢出唤醒功能；

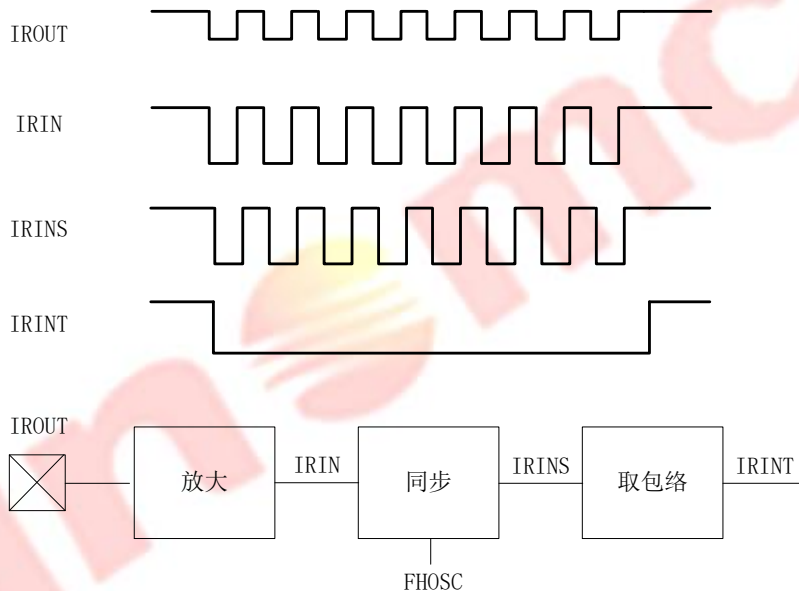


定时器 T0，可通过寄存器位 T0PTS 选择时钟源，通过 T0PR 选择预分频比，预分频比可选择 1~128 分频，所选时钟源通过预分频器后产生 T0 计数器 T0CNT 的计数时钟（上升沿计数）。当 TCOEN=0 时，T0CNT 保持不变，写重载寄存器 T0LOAD 将立即载入 T0CNT 中；当 TCOEN=1 时，T0CNT 递减计数，计数到 0 的时钟结束后产生溢出信号并触发中断，中断标志 T0IF 将被置 1，同时 T0 自动将当前 T0LOAD 值载入 T0CNT 中重新开始计数。

16 位 T0LOAD 配有 1 个 8 位的高位缓冲器，写 T0LOADL 时会同时将 T0LOADH 值载入该缓冲器中。此时若 TCOEN=0，则会同时再将[缓冲器:T0LOADL]载入 T0CNT；若 TCOEN=1，则需在 T0 溢出后才将[缓冲器:T0LOADL]的值载入 T0CNT。调整 T0LOAD 值时需先写 T0LOADH，再写 T0LOADL。

定时器 T0，可实现红外遥控输入信号的脉冲捕捉功能。从 IROUT 端口输入的小信号通过 OPA 放大后的信号 IRIN，经系统高频时钟 FHOSC 同步后，转换成载波脉冲信号 IRINS 及其包络信号 IRINT，T0 可通过 2 个 16 位的捕捉寄存器，锁存 IRINS 或 IRINT 上升/下降沿的 T0CNT 值。

### 遥控接收信号



当 T0RF 为 0 时，IRINS 下降沿将触发重载操作，重载寄存器 T0LOAD 值自动载入 T0CNT，同时 T0RF 置 1；T0RF 置 1 后，T0LATR 和 T0LATF 将在所选信号源的有效沿（T0MOD 选择）触发时锁存 T0CNT 的值。T0RF 需通过软件清 0。

### 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>	TC0EN	T0MOD1	T0MOD0	T0PTS1	T0PTS0	T0PR2	T0PR1	T0PR0
<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] **TC0EN** – 定时器 T0 使能位  
 0: 关闭定时器 T0;  
 1: 开启定时器 T0;

BIT[6:5] **T0MOD[1:0]** – T0LATR/T0LATF 锁存操作的触发有效沿选择位

T0MOD[1:0]	触发信号源	触发沿
00	IRINS	上升沿触发 T0LATR 锁存, 下降沿触发 T0LATF 锁存
01	IRINS	下降沿触发 T0LATR 锁存, 上升沿触发 T0LATF 锁存
10	IRINT	上升沿触发 T0LATR 锁存, 下降沿触发 T0LATF 锁存
11	IRINT	下降沿触发 T0LATR 锁存, 上升沿触发 T0LATF 锁存

BIT[4:3] **TOPTS[1:0]** – T0 时钟源选择位

TOPTS[1:0]	T0 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	INT0 上升沿

BIT[2:0] **TOPR[2:0]** – T0 预分频比选择位

TOPR[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 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0LOADH</b>	T0LOAD15	T0LOAD14	T0LOAD13	T0LOAD12	T0LOAD11	T0LOAD10	T0LOAD9	T0LOAD8
<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[15:8]** – T0 重载寄存器高 8 位, 用于设置 T0 的计数周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>T0LOADL</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 重载寄存器低 8 位, 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 捕捉寄存器**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOLATRH</b>	TOLATR15	TOLATR14	TOLATR13	TOLATR12	TOLATR11	TOLATR10	TOLATR9	TOLATR8
<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] **TOLATR[15:8]** – T0 捕捉寄存器 TOLATR 高 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOLATRL</b>	TOLATR7	TOLATR6	TOLATR5	TOLATR4	TOLATR3	TOLATR2	TOLATR1	TOLATR0
<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] **TOLATR[7:0]** – T0 捕捉寄存器 TOLATR 低 8 位

**注：在有效沿 (TOMOD 选择) 将当前 TOCNT 的值存储到 TOLATR 中。**

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOLATFH</b>	TOLATF15	TOLATF14	TOLATF13	TOLATF12	TOLATF11	TOLATF10	TOLATF9	TOLATF8
<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] **TOLATF[15:8]** – T0 捕捉寄存器 TOLATF 高 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TOLATFL</b>	TOLATF7	TOLATF6	TOLATF5	TOLATF4	TOLATF3	TOLATF2	TOLATF1	TOLATF0
<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] **TOLATF[7:0]** – T0 捕捉寄存器 TOLATF 低 8 位

**注：当 TORF=1 时在有效沿 (TOMOD 选择) 将当前 TOCNT 的值存储到 TOLATF 中，并停止存储，必须软件将 TORF 清 0 才能再次启动存储功能，清 TORF 同时也将清零 TOLATF。**

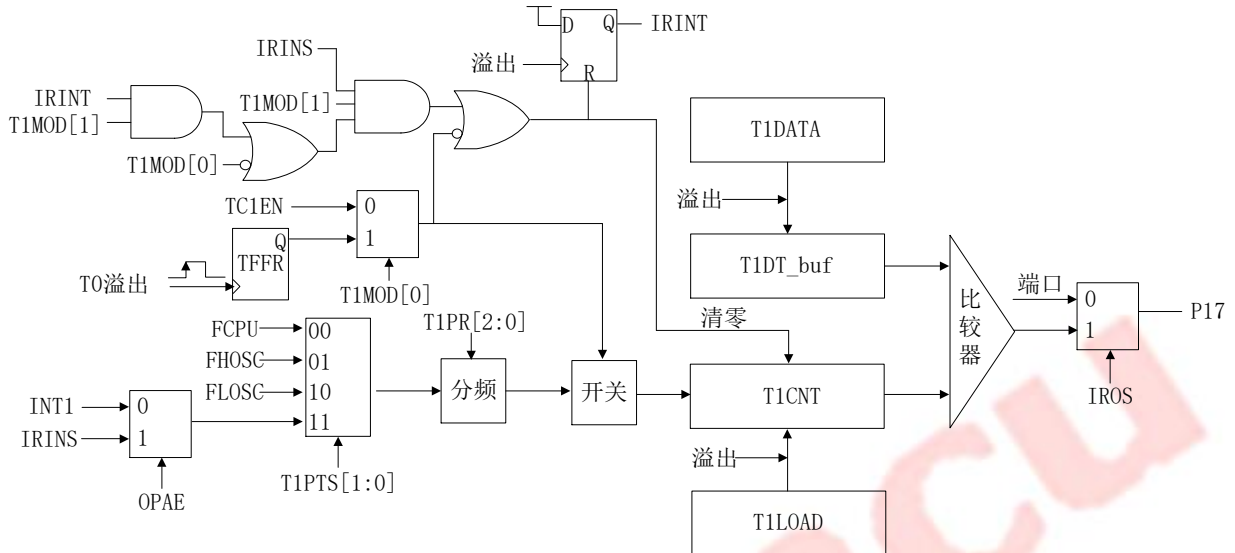
### 7.3 定时器 T1

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 8 位 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ◇ 支持溢出中断和溢出唤醒功能；
- ◇ 支持触发模式、包络模式、和单次触发模式

定时器 T1，可通过寄存器位 T1PTS 选择时钟源，通过 T1PR 选择预分频比，预分频比可选择 1~128 分频，所选时钟源通过预分频器后产生 T1 计数器 TICNT 的计数时钟（上升沿计数）。当 TC1EN=0 时，

T1CNT 保持不变，写重载寄存器 T1LOAD 将立即载入 T1CNT 中；当 TC1EN=1 时，T1CNT 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 T1IF 将被置 1，同时 T1 自动将当前 T1LOAD 值载入 T1CNT 中重新开始计数。



如图所示，定时器 T1 可实现 PWM 功能（PWM1），可通过寄存器位使能/关闭 PWM 功能，或控制端口是否输出 PWM 波形。PWM1 关闭时 PWM 信号为低电平。PWM1 使能后，T1CNT 从重载值开始递减计数直到计数溢出为一个 PWM 周期：当计数到与比较寄存器 T1DATA 相等时，PWM 信号变为高电平；当计数溢出时，PWM 信号变为低电平。

T1DATA 配有 1 个 8 位的比较缓冲器（T1DT\_buf）用于与 T1CNT 比较，PWM1 关闭时写 T1DATA 将立即载入比较缓冲器中，而 PWM1 使能后写 T1DATA 则将在 T1 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，再使能 PWM，最后开启定时器。

PWM1 的占空比计算如下：

- ◇ 高电平时间 = (T1DATA) × T1CNT 计数时钟周期
- ◇ 周期 (T1 的溢出周期) = (T1LOAD+1) × T1CNT 计数时钟周期
- ◇ 占空比 = 高电平时间 / 周期 = (T1DATA) / (T1LOAD+1)

定时器 T1 支持 4 种工作模式：定时/计数/PWM 模式，触发模式，包络模式，单次触发模式。

触发模式，当 TC1EN=1 时 T1 由 T0 的溢出触发在开与关之间交替切换，当 TC1EN=0 时不触发，保持当前状态。

包络模式，当 IRINS 为低电平时 T1CNT 清零，并清零 IRINT，当 IRINS 为高电平时 T1CNT 重载并运行，当 T1CNT 溢出时置位 IRINT。

单次触发模式，当 TC1EN=1 时 T1 由 T0 的溢出触发开启，当 T1 溢出时自动关闭，当 TC1EN=0 时不触发，保持当前状态。



**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>	TC1EN	T1MOD1	T1MOD0	T1PTS1	T1PTS0	T1PR2	T1PR1	T1PR0
<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] **TC1EN** – 定时器 T1 使能位  
 0: 关闭定时器 T1;  
 1: 开启定时器 T1;

BIT[6:5] **T1MOD[1:0]** – T1 工作模式选择位

<b>T1MOD[1:0]</b>	<b>工作模式</b>	<b>描述</b>
00	定时/计数/PWM 模式	由 TC1EN 决定开关
01	触发模式	由 T0IF 触发交替开关
10	包络模式	IRINS 低电平清零 T1CNT
11	单次触发模式	T0IF 触发 T1 开, 溢出触发 T1 关

BIT[4:3] **T1PTS[1:0]** – T1 时钟源选择位

<b>T1PTS[1:0]</b>	<b>T1 时钟源</b>
00	FCPU
01	FHOSC
10	FLOSC
11	INT1 (OPAE=0) 或 IRINS (OPAE=1)

BIT[2:0] **T1PR[2:0]** – T1 预分频比选择位

<b>T1PR[2:0]</b>	<b>T1 时钟预分频比</b>
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 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>	0	0	0	0	0	0	0	0

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 键盘扫描电路

### 8.1 键盘扫描概述

芯片的 P16~P10 和 P27~P24/P22/P21 可设置为键盘扫描模式，当系统进入休眠/HOLD 模式时，硬件自动按照下图设置相应端口为输出低电平或者输入上拉状态。键盘扫描时钟为系统低频时钟 F<sub>LIRC</sub>（即 FLIRC），扫描周期可通过寄存器位 WSEL 选择为 16ms、32ms、48ms 或 64ms，未设置为键盘扫描模式的端口不会输出波形但不会影响下图的时序。



### 8.2 键盘扫描相关寄存器

#### 键盘扫描寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKW0	DKWE	IROS	IROT	WSEL1	WSEL0	RSEL	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
初始值	0	0	0	0	0	0	-	-

BIT[7] DKWE – 键盘唤醒选择位  
 0: 矩阵型键盘唤醒;  
 1: T 型键盘唤醒;

**注: T 型键盘需同时设置 OSCM 的 LFEN 位为 1; 矩阵型键盘需同时设置 OSCM 的 LFEN 位为 0.**

BIT[6] **IROS** – IROUT 输出选择位  
0: P17;  
1: PWM1;

BIT[5] **IROT** – IRIN 输出选择位  
0: P14;  
1: IRIN;

BIT[4:3] **WSEL[1:0]** – 键盘扫描周期选择位

WSEL[1:0]	键盘扫描周期
00	16ms
01	32ms
10	48ms
11	64ms

BIT[2] **RSEL** – 端口上拉电阻选择位  
0: 端口内部上拉电阻为 60KΩ;  
1: 端口内部上拉电阻为 150KΩ;

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>DKW1</b>	-	-	IRES1	IRES0	LSEL11	LSEL10	LSEL01	LSEL00
<b>R/W</b>	-	-	R/W	R/W	R/W	R/W	R/W	R/W
<b>初始值</b>	-	-	1	1	0	0	0	0

BIT[5] **IRES1** – P22 端口上拉电阻选择位  
0: 端口内部上拉电阻为 5KΩ;  
1: 端口内部上拉电阻由 RSEL 决定;

BIT[4] **IRES0** – P21 端口上拉电阻选择位  
0: 端口内部上拉电阻为 5KΩ;  
1: 端口内部上拉电阻由 RSEL 决定;

BIT[3:2] **LSEL1[1:0]** – P22 端口输出灌电流选择位

LSEL1[1:0]	P22 输出灌电流
00	2mA
01	4mA
10	8mA
11	12mA

BIT[1:0] **LSEL0[1:0]** – P21 端口输出灌电流选择位

LSEL0[1:0]	P21 输出灌电流
00	2mA
01	4mA

10	8mA
11	12mA

### 8.3 键盘扫描操作步骤

- (1) 设置 DKWP1 和 DKWP2，将相应端口设置为键盘扫描模式；
- (2) 设置 PUP1 和 PUP2，开启相应端口的上拉电阻；
- (3) 设置 DKWE，开启键盘扫描；
- (4) 关闭中断总使能位 GIE；
- (5) 执行 STOP 指令；
- (6) 等待端口唤醒后，CPU 将执行 STOP 指令后的下一条指令；
- (7) 设置 DKWP1 和 DKWP2，将相应端口设置为通用 I/O 端口；

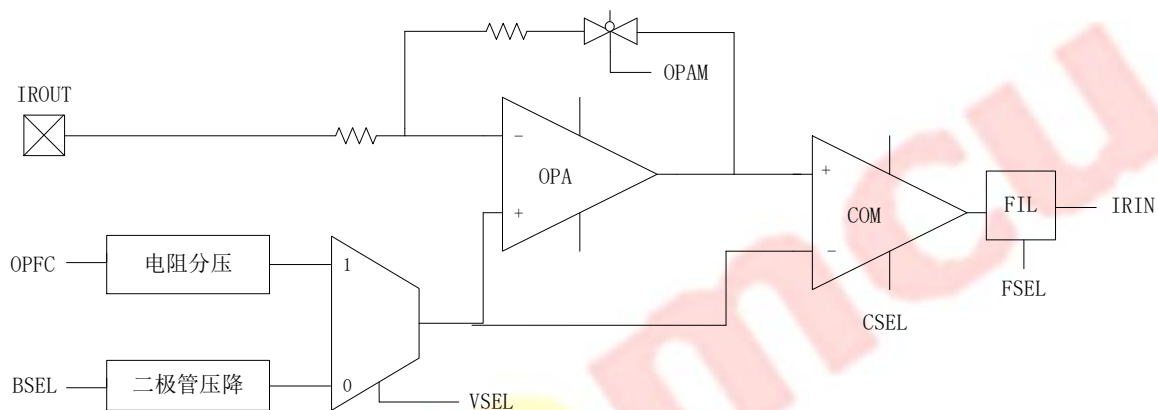
**注：**若上述操作中第 (3) 步不使能 DKWE，则芯片将只能由端口 (设置为键盘扫描模式) 的下降沿触发唤醒，此方法可使芯片支持矩阵型键盘。

## 9 运算放大器 OPA

### 9.1 OPA 概述

芯片内置 1 个运算放大器 OPA，可实现 IROUT 引脚输入小信号的放大，从而实现遥控信号的学习功能。

- ◇ 可设置灵敏度；
- ◇ 可设置不同滤波频率；
- ◇ 支持二极管校准和电阻校准等校准模式；



### 9.2 OPA 相关寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>OPCR0</b>	OPAE	OPAM	OPFC5	OPFC4	OPFC3	OPFC2	OPFC1	OPFC0
<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] **OPAE** – 运放 OPA 使能位

- 0: 关闭 OPA;
- 1: 开启 OPA;

BIT[6] **OPAM** – OPA 模式选择位

- 0: 工作模式;
- 1: 校准模式;

BIT[5:0] **OPFC[5:0]** – OPA 校准位

VSEL	BIT[5:0]	用途
0	BIT[5:3]	无效
	BIT[2:0]	用作二极管压降调节位 BSEL[2:0]
1	BIT[5:0]	用作电阻分压调节位 OPFC[5:0]

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPCR1	IRIN	VSEL	-	-	FSEL1	FSEL0	CSEL1	CSEL0
R/W	R	R/W	-	-	R/W	R/W	R/W	R/W
初始值	0	0	-	-	0	0	0	0

BIT[7] **IRIN** – OPA 输出状态标志位  
 0: OPA 输出低, 即 OPA 正端输入电压低于负端输入电压;  
 1: OPA 输出高, 即 OPA 正端输入电压高于负端输入电压;

BIT[6] **VSEL** – OPA 校准模式选择位  
 0: 二极管模式 (推荐);  
 1: 电阻分压模式;

BIT[3:2] **FSEL[1:0]** – OPA 滤波选择位

FSEL[1:0]	OPA 滤波时间
00	1.5us (推荐)
01	3us
10	3.5us
11	4us

BIT[1:0] **CSEL[1:0]** – OPA 的 OFFSET 选择位

CSEL[1:0]	OPA OFFSET
00	5mV (灵敏度最低)
01	20mV
10	36mV
11	65mV (灵敏度最高)

### 9.3 OPA 操作步骤

- (1) 设置 FSEL 和 CSEL;
- (2) 设置 VSEL 位选择不同模式;
- (3) 置位 OPAE 使能 OPA;
- (4) 置位 OPAM 进入 OPA 校准模式;
- (5) 根据不同的 VSEL 设置调整 OPFC 或 BSEL 从最大值以步长为 1 递减, 直到 IRIN 为 1, 此时 OPFC 或 BSEL 的值即为校准值; 每步读取 IRIN 需等待 100us, 判断次数不少于 3 次;
- (6) 清零 OPAM 进入正常工作模式;

**注:** OPA 开启后, IROUT 引脚自动转换为 OPA 输入端, 需保证其他端口始终处于输入上拉状态, 避免干扰 OPA 工作。

## 10 MTP 存储器

### 10.1 MTP 概述

芯片内置 1 个 64 页（每页 16 字节）共 1024 字节的 MTP 数据存储器。MTP 的操作需通过 MTP 地址寄存器 MTPAD 和 MTP 数据缓存器 MTPDB 执行，MTP 写入完成后将触发 MTP 中断。

MTP 的存储内容擦除后其值为 FFH，使用 MTP 页写需预先对相应区块执行擦除操作（页擦除或全片擦除），而使用字节写则不需要预先擦除。

MTP 写操作的典型时间见下表：

MTP 写操作模式	BUSY 时间
字节写	12ms
页写	3ms
页擦除	9ms
全片擦除	9ms

### 10.2 MTP 相关寄存器

#### MTP 模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MTPMD</b>	MTPEN	-	-	CHIPERS	PAGEERS	PAGEWR	BYTEWR	BYTERD
<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[7] **MTPEN** – MTP 操作使能位

- 0: 禁止执行 MTP 操作；
- 1: 允许执行 MTP 操作；

BIT[4] **CHIPERS** – 全擦除模式选择位

- 0: 未选择；
- 1: 全擦除模式；

BIT[3] **PAGEERS** – 页擦除模式选择位

- 0: 未选择；
- 1: 页擦除模式；

BIT[2] **PAGEWR** – 页写模式选择位

- 0: 未选择；
- 1: 页写模式；



BIT[1] **BYTEWR** – 字节写模式选择位  
0: 未选择;  
1: 字节写模式;

BIT[0] **BYTERD** – 字节读模式选择位  
0: 未选择;  
1: 字节读模式;

**注: 同一时刻只能选择一种模式, 清 0 使能位 MTPEN 将清零寄存器 MTPMD。**

### MTP 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MTPCTR</b>	MTPBUSY	MTPTOOUT	-	-	-	MTPLD	MTPWR	MTPRD
<b>R/W</b>	R	R	-	-	-	R/W	R/W	R/W
<b>初始值</b>	0	0	-	-	-	0	0	0

BIT[7] **MTPBUSY** – MTP 操作状态标志位  
0: MTP 操作完成;  
1: MTP 操作执行中;

BIT[6] **MPTOOUT** – MTP 操作超时标志位  
0: MTP 操作正常;  
1: MTP 操作超时;

BIT[2] **MTPLD** – MTP 页缓存数据加载控制位  
0: 已启动页缓存加载操作 (写 1 后硬件自动清 0);  
1: 启动将 MTPDB 载入 MTP 页缓存的操作 (MTP 页缓存区为 16 字节);

BIT[1] **MTPWR** – MTP 写操作控制位  
0: 已启动 MTP 写操作 (写 1 后硬件自动清 0);  
1: 启动 MTP 写操作;

BIT[0] **MTPRD** – MTP 读操作控制位  
0: 已启动 MTP 读操作 (写 1 后硬件自动清 0);  
1: 启动 MTP 读操作;

### MTP 地址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MTPADH</b>	-	-	MTPAD9	MTPAD8	MTPAD7	MTPAD6	MTPAD5	MTPAD4
<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[5:0] **MTPAD[9:4]** – MTP 操作的地址高 6 位, 即页地址

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MTPADL</b>	-	-	-	-	MTPAD3	MTPAD2	MTPAD1	MTPAD0
<b>R/W</b>	-	-	-	-	R/W	R/W	R/W	R/W
<b>初始值</b>	-	-	-	-	0	0	0	0

BIT[3:0] **MTPAD[3:0]** – MTP 操作的地址低 4 位，即页内偏移地址

### MTP 数据缓存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>MTPDB</b>	MTPDB7	MTPDB6	MTPDB5	MTPDB4	MTPDB3	MTPDB2	MTPDB1	MTPDB0
<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] **MTPDB[7:0]** – MTP 操作的 8 位数据缓存

## 10.3 MTP 操作示例

### 页设置

MTP\_PAGE\_SET:

```

MOVRA    MTPADH    ; 设置 PAGE 地址
CLRR     MTPADL    ; 清零 PAGE 缓存地址
CLRR     FSR0      ; 设置 RAM 映射地址 (00H-0FH 用作 PAGE 数据缓冲区)
RETURN
    
```

### 页写

PAGE\_WRITE:

```

CALL     PAGE_ERASE
BSET     PAGEWR    ; 配置为页写模式
BSET     MTPEN     ; 允许 MTP 操作
CALL     DELAY_10us
MOVAI   PAGE      ; 设置 PAGE 地址
CALL     MTP_PAGE_SET
    
```

PAGE\_DATA\_LOAD:

```

MOVAR   INDF0
MOVRA   MTPDB     ; 读取 RAM 中的数据
BSET   MTPDL     ; 数据写入缓存
NOP     ; 需增加两条 NOP 指令
NOP
INCR   FSR0     ; RAM 地址更改
INCR   MTPADL   ; 缓存地址更改
MOVAR   MTPADL
    
```

```

JBSET      Z                ; 判断 PAGE 缓存是否满
GOTO      PAGE_DATA_LOAD
BSET      MTPWR            ; 缓存数据写入 MTP
CALL      MTP_BUSY_CHK
RETURN
    
```

### 页擦除

PAGE\_ERASE:

```

BSET      PAGEERS          ; 配置为页擦除模式
BSET      MTPEN           ; 允许 MTP 操作
CALL      DELAY_10us
MOVAI    PAGE             ; 设置 PAGE 地址
CALL      MTP_PAGE_SET
BSET      MTPWR           ; 页擦除操作
CALL      MTP_BUSY_CHK
RETURN
    
```

### 全擦除

CHIP\_ERASE:

```

BSET      CHIPERS         ; 配置为全擦除模式
BSET      MTPEN           ; 允许 MTP 操作
CALL      DELAY_10us
BSET      MTPWR           ; 全擦除操作
CALL      MTP_BUSY_CHK
RETURN
    
```

### BUSY 判断

MTP\_BUSY\_CHK:

```

NOP                ; 查询 MTPBUSY 前需增加四条 NOP 指令
NOP
NOP
NOP
JBCLR     MTPBUSY    ; 判断是否写入完成
;          JBSET     MTPIF    ; 也可用中断标志判断，或者使用中断模式
GOTO     MTP_BUSY_CHK
CALL     DELAY_10us
BCLR     MTPEN       ; 禁止 MTP 操作
JBCLR     MTP_TOUT   ; 判断是否写入错误
GOTO     MTP_PROG_ERR ; MTP 编程错误
RETURN
    
```

## 字节写

BYTE\_WRITE:

BSET	BYTEWR	; 配置为字节编程模式
BSET	MTPEN	; 允许 MTP 操作
CALL	DELAY_10us	
MOVAI	PAGE	; 设置 PAGE 地址
MOVRA	MTPADH	
MOVAI	PAGE_OFFSET	; 设置 PAGE 偏移地址
MOVRA	MTPADL	
MOVAR	DATA_IN	; 设置写入数据
MOVRA	MTPDB	
BSET	MTPWR	; 缓存数据写入 MTP
CALL	MTP_BUSY_CHK	
RETURN		

## 字节读

BYTE\_READ:

BSET	BYTERD	; 配置为字节编程模式
BSET	MTPEN	; 允许 MTP 操作
CALL	DELAY_10us	
MOVAI	PAGE	; 设置 PAGE 地址
MOVRA	MTPADH	
MOVAI	PAGE_OFFSET	; 设置 PAGE 偏移地址
MOVRA	MTPADL	
BSET	MTPRD	; MTP 数据存储到 MTPDB
NOP		; 需增加两条 NOP 指令
NOP		
MOVAR	MTPDB	; 读取数据
MOVRA	DATA_OUT	; 保存数据
RETURN		

## 页校验

PAGE\_VERIFY:

BSET	BYTERD	; 配置为字节读模式
BSET	MTPEN	; 允许 MTP 操作
CALL	DELAY_10us	
MOVAI	PAGE	; 设置 PAGE 地址
CALL	MTP_PAGE_SET	

PAGE\_DATA\_READ:

MOVAR	INDFO	
BSET	MTPRD	; MTP 数据存储到 MTPDB
NOP		; 需增加两条 NOP 指令
NOP		

XORAR	MTPDB	: 数据校验
JBSET	Z	
GOTO	MTP_VERIFY_ERR	: 校验错误
INCR	FSRO	: RAM 地址更改
INCR	MTPADL	: 读取地址更改
MOVAR	MTPADL	
JBSET	Z	: 判断 PAGE 读取是否完成
GOTO	PAGE_DATA_READ	
BCLR	MTPEN	: 校验完成, 禁止 MTP 操作
RETURN		

## 10.4 MTP 数据烧录

MTP 数据的初始化可通过烧录器实现，烧录器将 MTP 数据区映射至地址 4000H~43FFH。

MTP 的初始化示例如下：

```

ORG      4000H
DW      55H
DW      66H
...

```

## 11 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 选择电压检测测量值。当 VDD 电压低于电压检测测量值时检测状态标志位 LVDF 将被置 1，同时中断标志 LVDIF 置 1 触发 LVD 中断；反之 LVDF 将被清 0（但中断标志不会自动清 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	-	-	-	LVDVS	-	-	LVDF
<b>R/W</b>	R/W	-	-	-	R/W	-	-	R
<b>初始值</b>	0	-	-	-	0	-	-	0

BIT[7] **LVDEN** – 低电压检测 LVD 使能位

- 0: 关闭 LVD;
- 1: 开启 LVD;

BIT[3] **LVDVS** – LVD 电压检测测量值选择位

- 0: 2.0V;
- 1: 2.4V;

BIT[0] **LVDF** – LVD 检测状态标志位

- 0: VDD 电压高于电压检测测量值，或 LVD 关闭;
- 1: VDD 电压低于电压检测测量值;

## 12 中断

芯片的中断源包括电平中断（INTS，INTT）、定时器中断（T0~T1）、LVD 中断和 MTP 中断等。可通过中断总使能位 GIE 屏蔽所有中断。

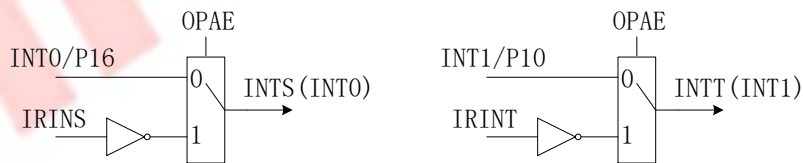
CPU 响应中断的过程如下：

- ◇ 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。

### 12.1 电平中断

芯片有 2 路电平中断源 INTS/INTT，可选择上升沿、下降沿或电平变化等触发方式。如下图所示，当内部 OPA 关闭时，电平中断的输入源为端口 INTO/INT1 输入的外部电平信号，而当 OPA 开启后，电平中断的输入源自动切换为 OPA 输出信号 IRIN 的同步信号 IRINS 和包络信号 IRINT。

电平中断触发时，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：要使用外部输入电平的电平中断功能，需将相应端口设为输入状态。**

### 12.2 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。

## 12.3 LVD 中断

当 VDD 电压从高于电压检测量值降为低于量值时触发 LVD 中断，中断标志 LVDIF 将被置 1，若中断总使能位 GIE 为 1 且 LVD 中断使能位 LVDIE 为 1，则产生 LVD 中断。

## 12.4 MTP 中断

启动 MTP 写操作后，MTP 写操作完成时将触发 MTP 中断，中断标志 MTPIF 将被置 1，若中断总使能位 GIE 为 1 且 MTP 中断使能位 MTPIE 为 1，则产生 MTP 中断。

## 12.5 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>	MTPIE	-	LVDIE	-	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[7] **MTPIE** – MTP 中断使能位  
 0: 屏蔽 MTP 中断;  
 1: 使能 MTP 中断;

BIT[5] **LVDIE** – LVD 中断使能位  
 0: 屏蔽 LVD 中断;  
 1: 使能 LVD 中断;

BIT[3] **INT1IE** – INTT (INT1) 中断使能位  
 0: 屏蔽 INTT (INT1) 中断;  
 1: 使能 INTT (INT1) 中断;

BIT[2] **INT0IE** – INTS (INT0) 中断使能位  
 0: 屏蔽 INTS (INT0) 中断;  
 1: 使能 INTS (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>	MPIF	TORF	LVDIF	-	INT1IF	INTOIF	T1IF	TOIF
<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] **MPIF** – MTP 中断标志位  
 0: 未触发 MTP 中断;  
 1: 已触发 MTP 中断, 需软件清 0;

BIT[6] **TORF** – 定时器 T0 重载标志位  
 0: 定时器 T0 未发生重载;  
 1: 定时器 T0 发生重载, 需软件清 0;

BIT[5] **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** – INTT (INT1) 中断标志位  
 0: 未触发 INTT (INT1) 中断;  
 1: 已触发 INTT (INT1) 中断, 需软件清 0;

BIT[2] **INTOIF** – INTS (INT0) 中断标志位  
 0: 未触发 INTS (INT0) 中断;  
 1: 已触发 INTS (INT0) 中断, 需软件清 0;

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 0;

BIT[0] **TOIF** – 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 0;

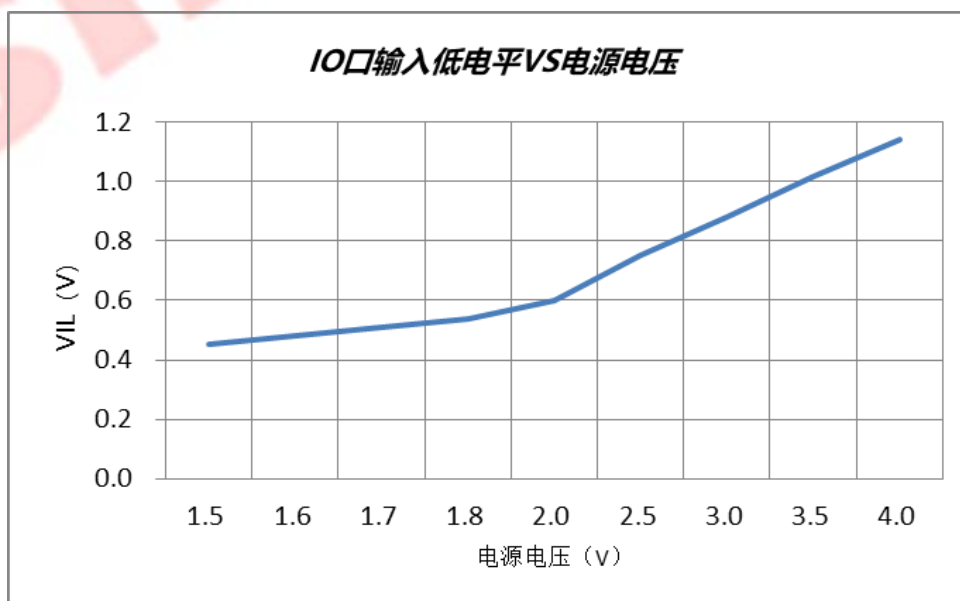
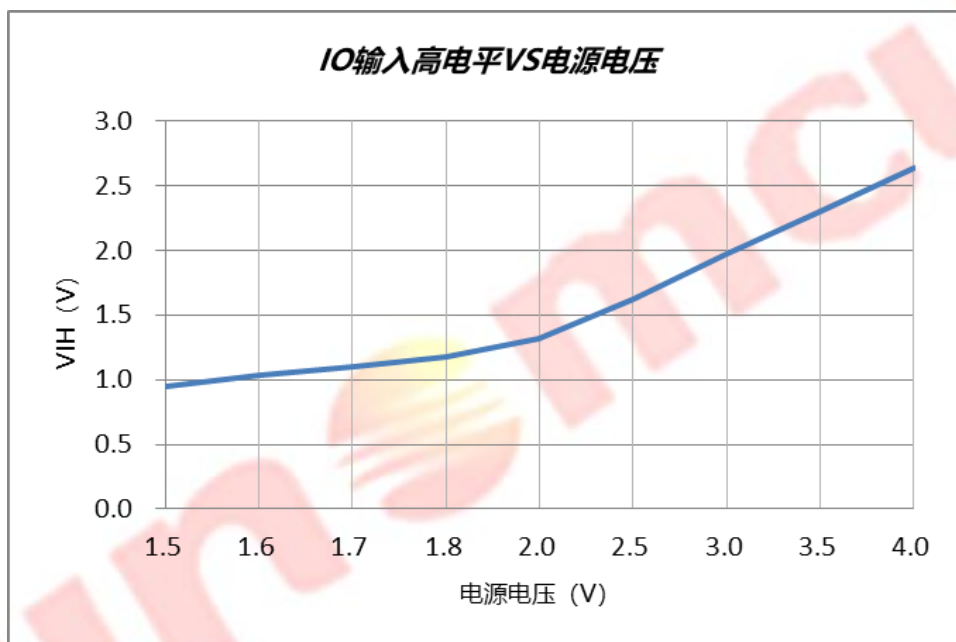
## 13 特性曲线

注:

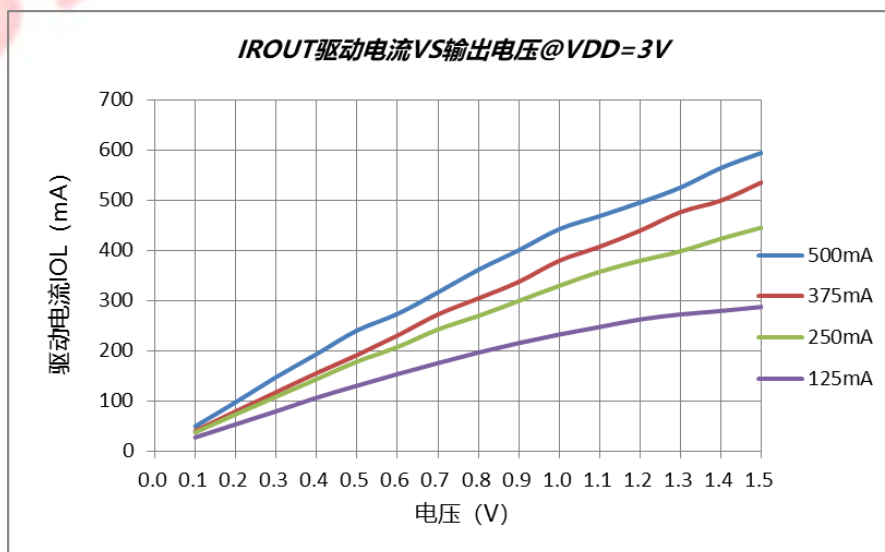
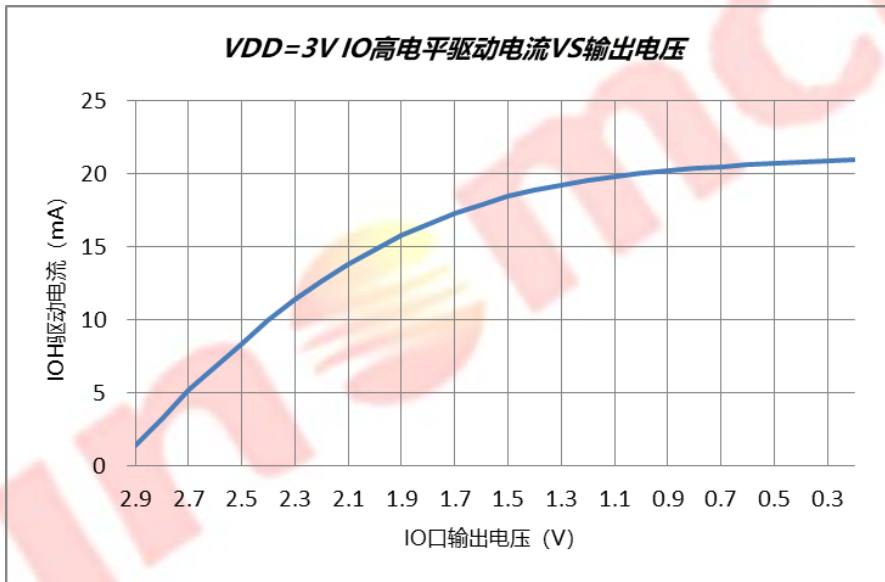
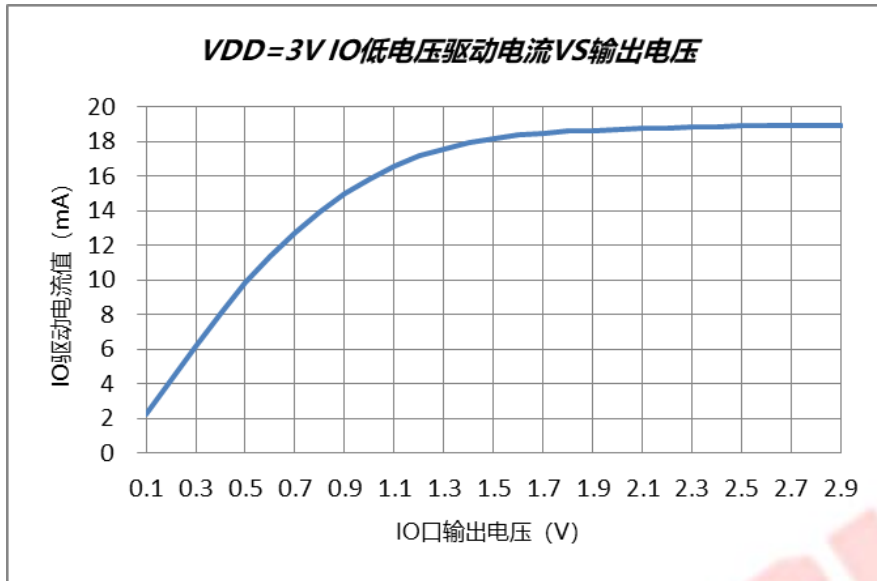
1. 特性曲线图中数据均来自抽样实测, 仅作为应用参考, 部分数据因生产工艺偏差, 可能与实际芯片不符; 为保证芯片能正常工作, 请确保其工作条件符合电气特性参数说明;
2. 若图文中无特别说明, 则电压特性曲线的温度条件为  $T=25^{\circ}\text{C}$ , 温度特性曲线的电压条件为  $V_{DD}=3\text{V}$ ;

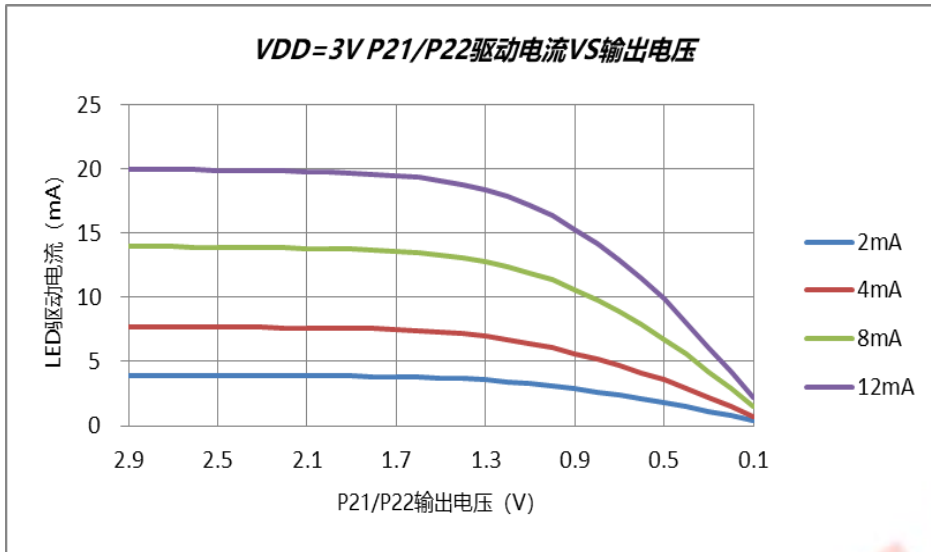
### 13.1 I/O 特性

#### 输入 SMT 阈值电压 VS 电源电压

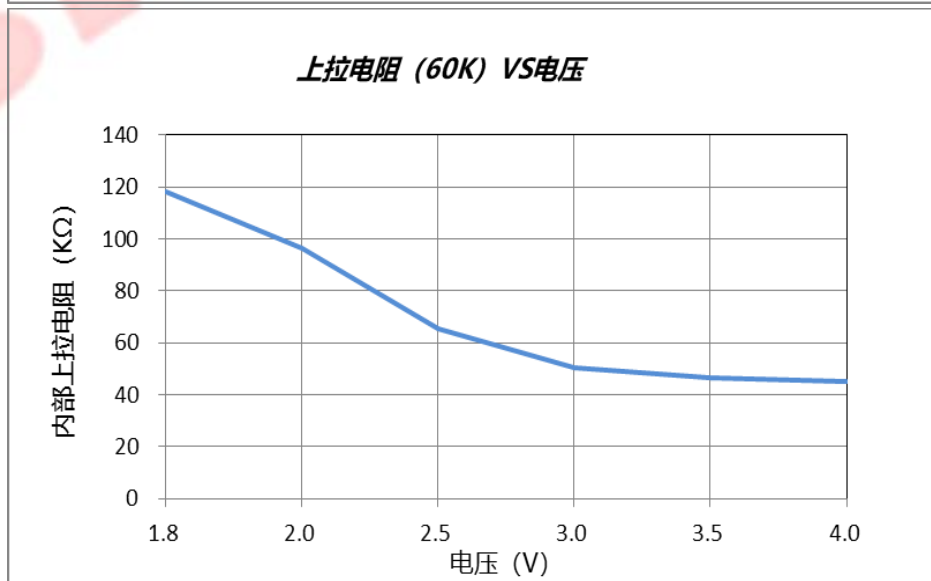
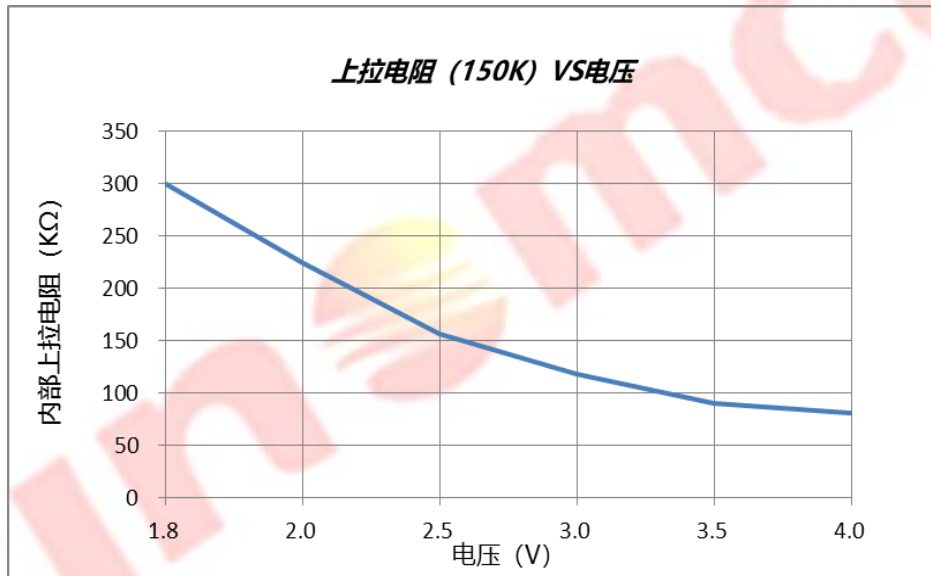


I/O 输出 驱动电流 VS 端口电压 (VDD=3V)



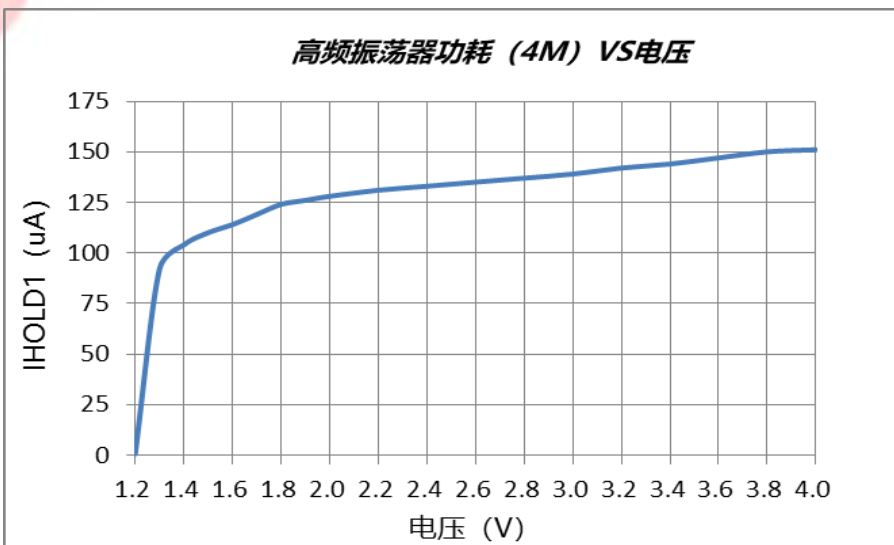
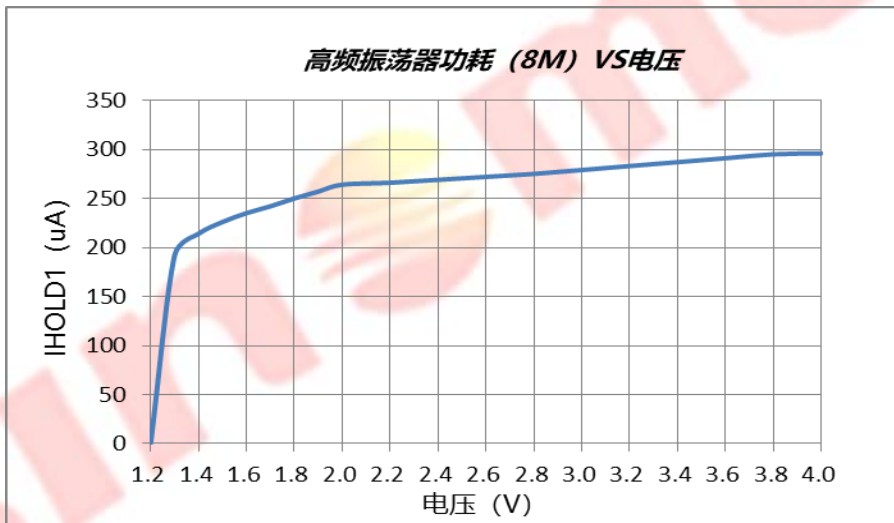
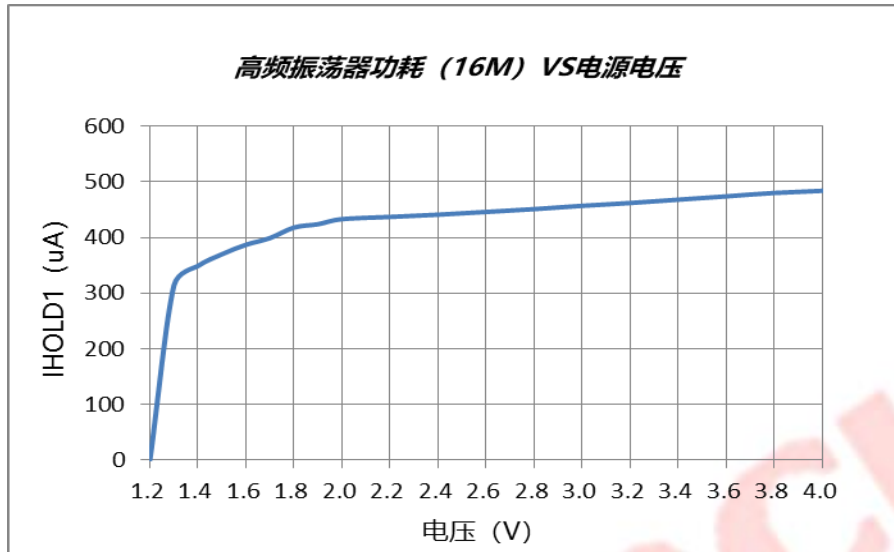


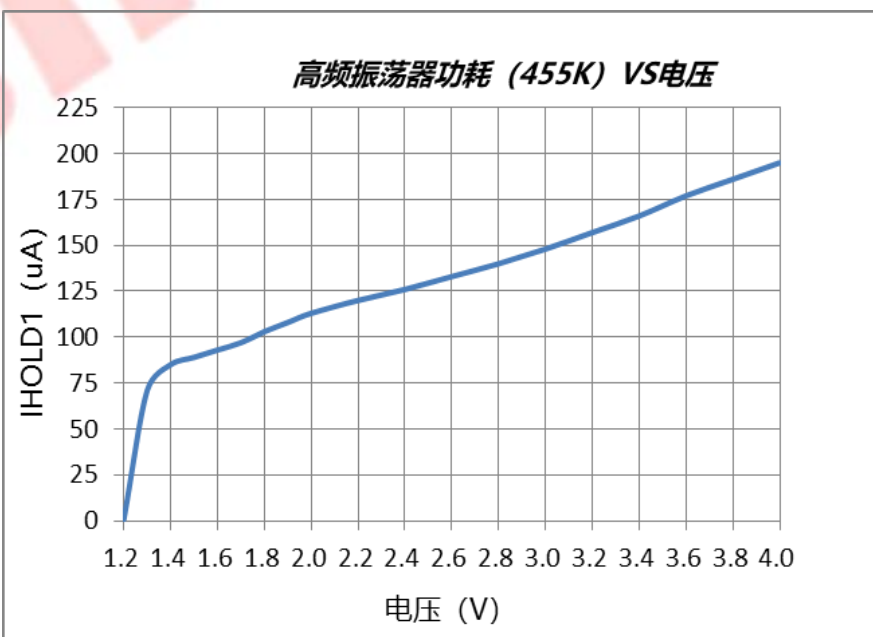
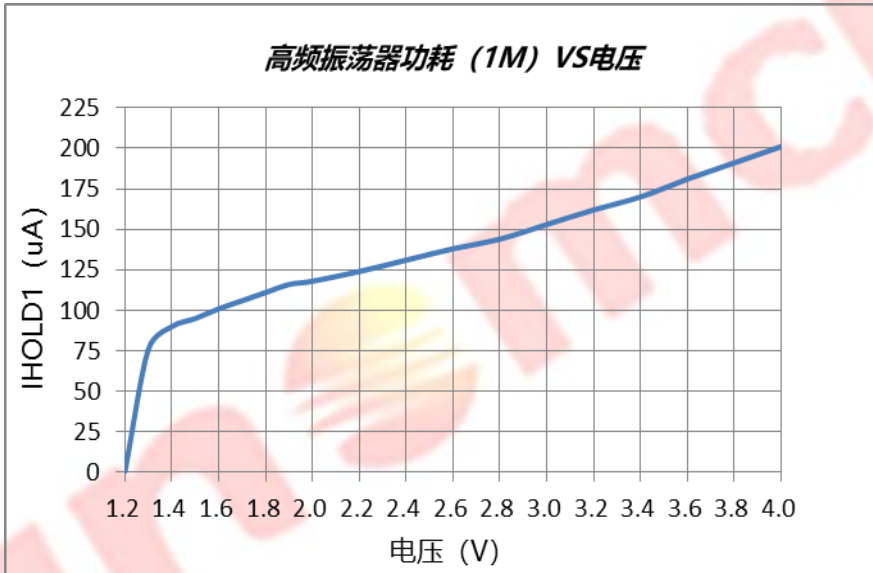
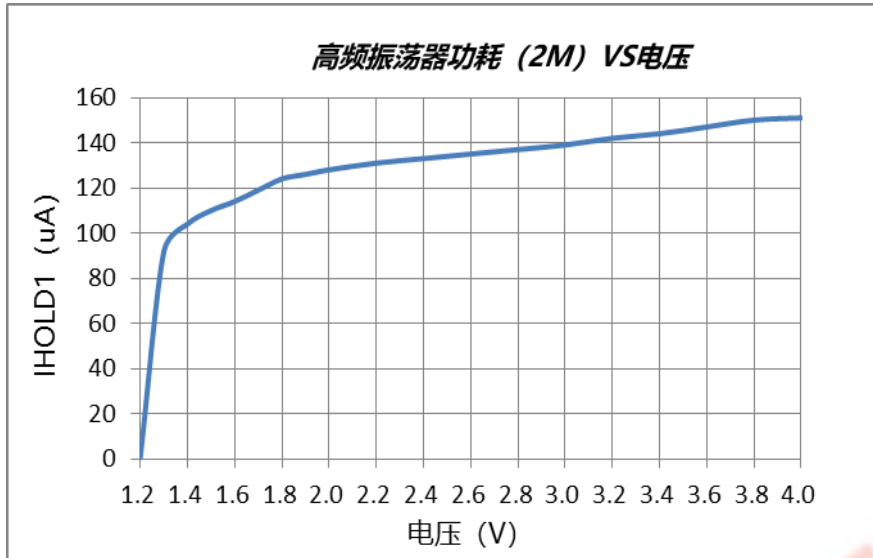
**上拉电阻值 VS 电源电压**



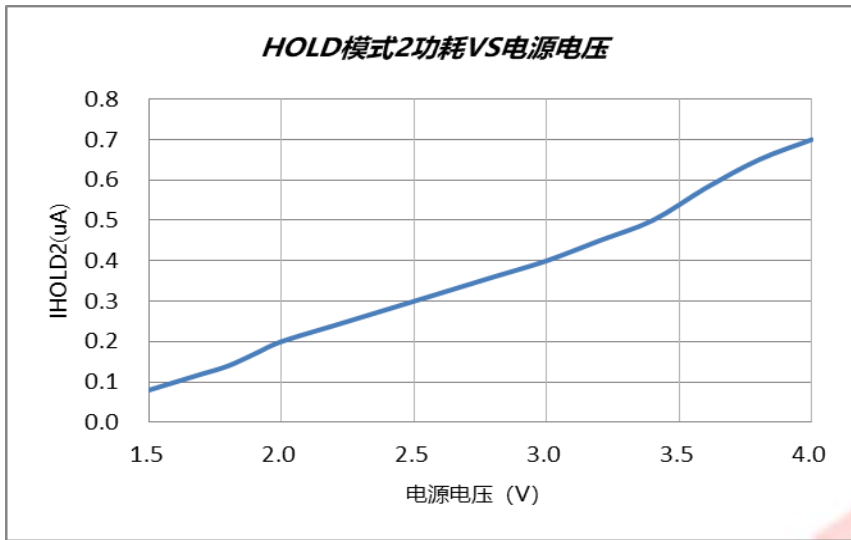
## 13.2 功耗特性

### HOLD1 模式 功耗 VS 电源电压

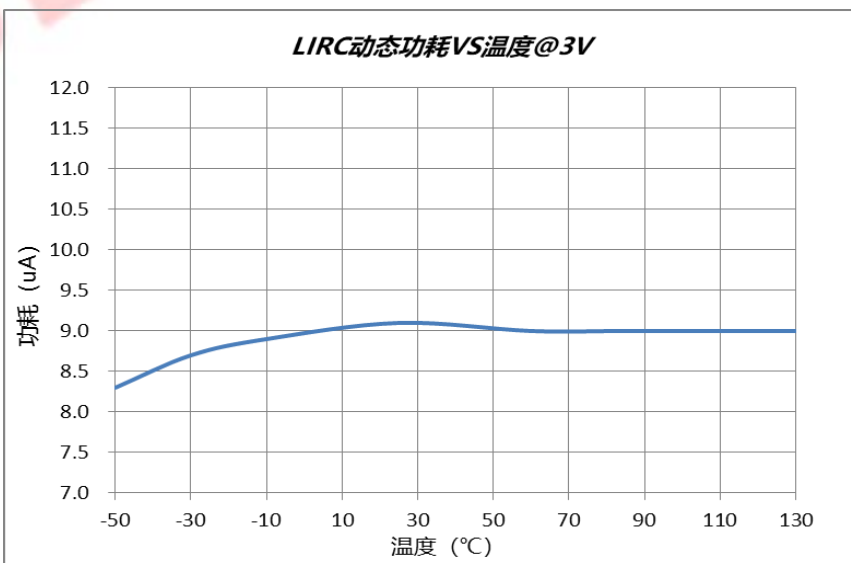
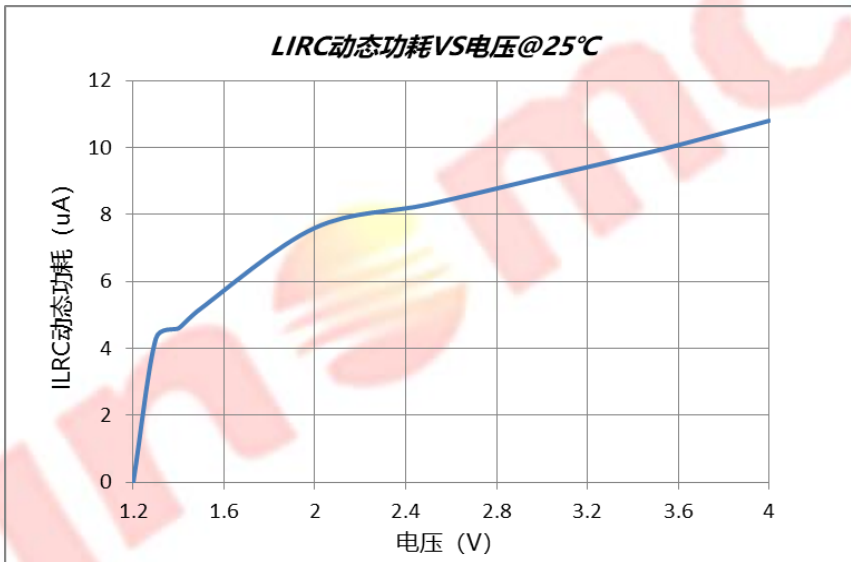




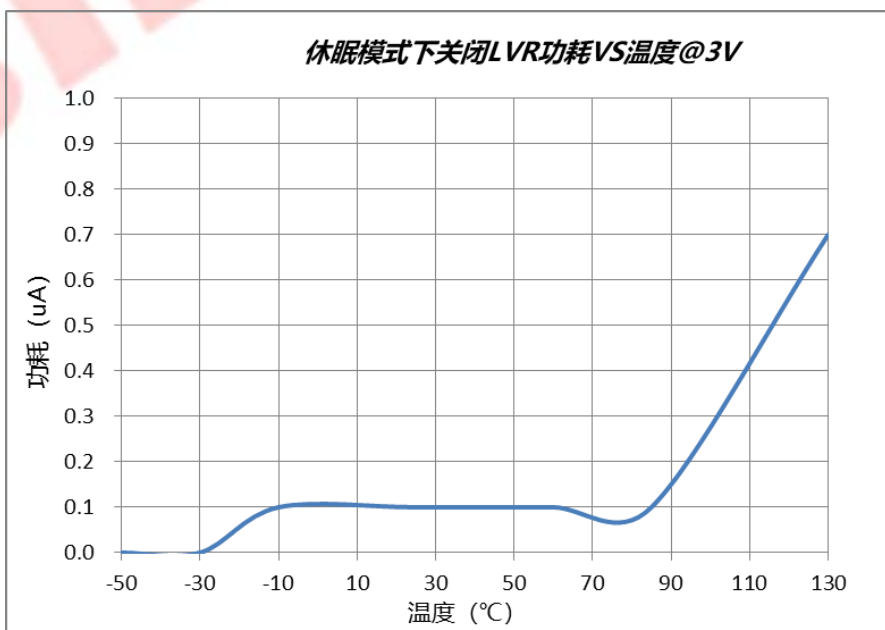
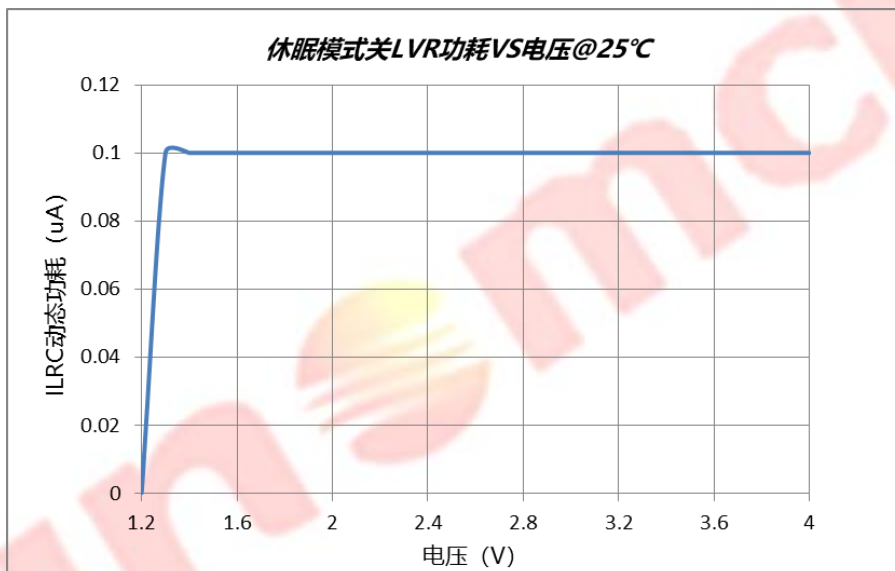
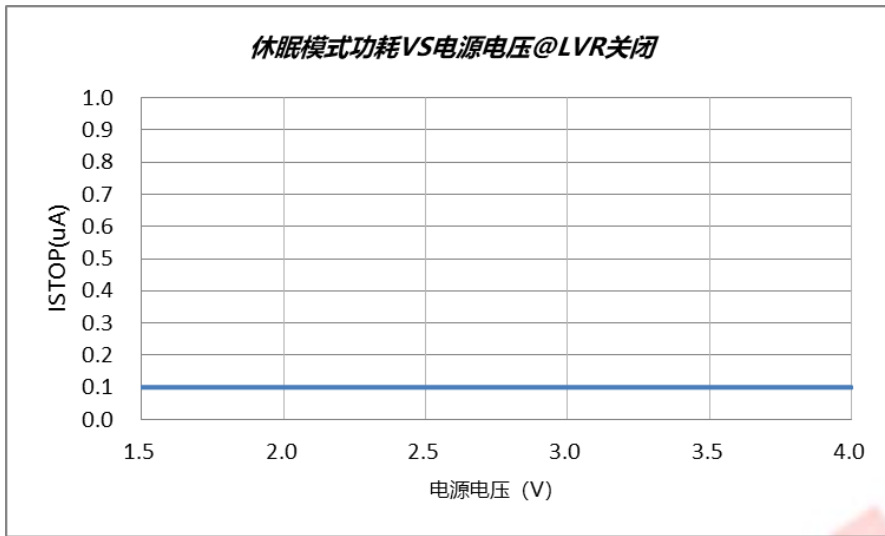
**HOLD2 模式 功耗 VS 电源电压**



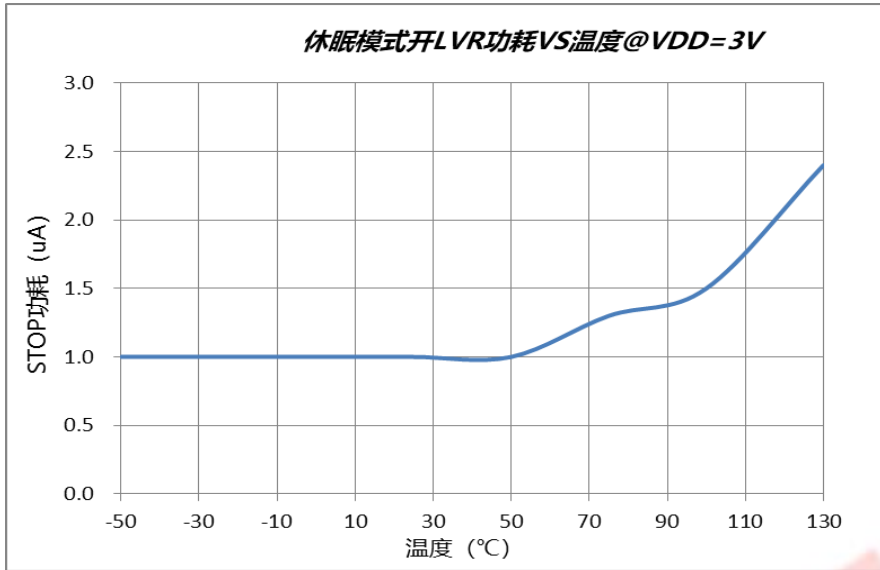
**LIRC 动态功耗 VS 电源电压/温度**



休眠模式 功耗 VS 电源电压/温度

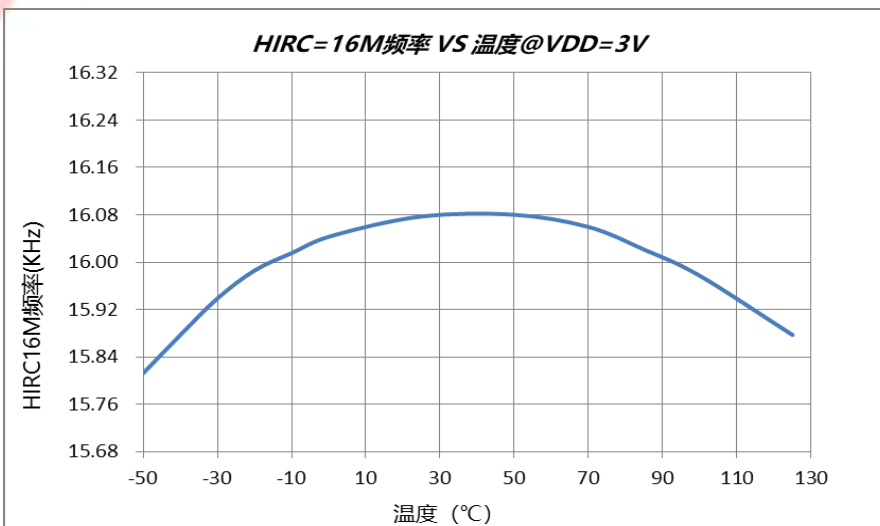
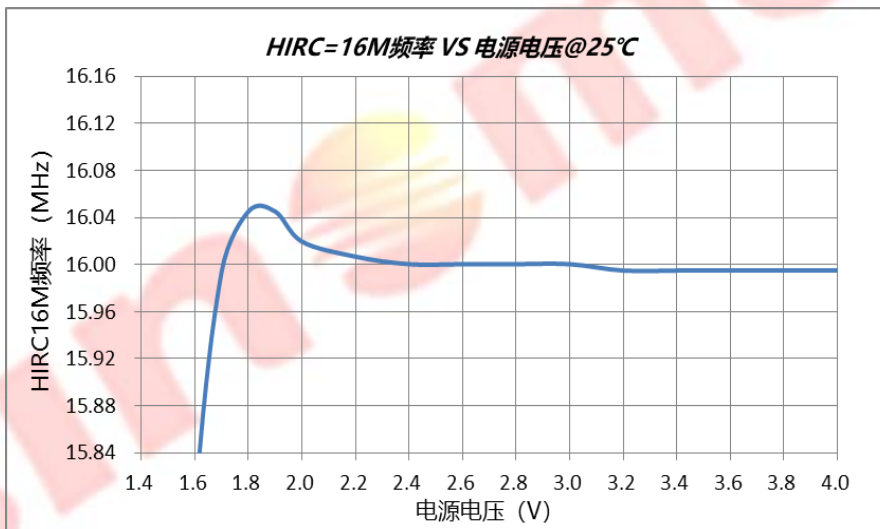


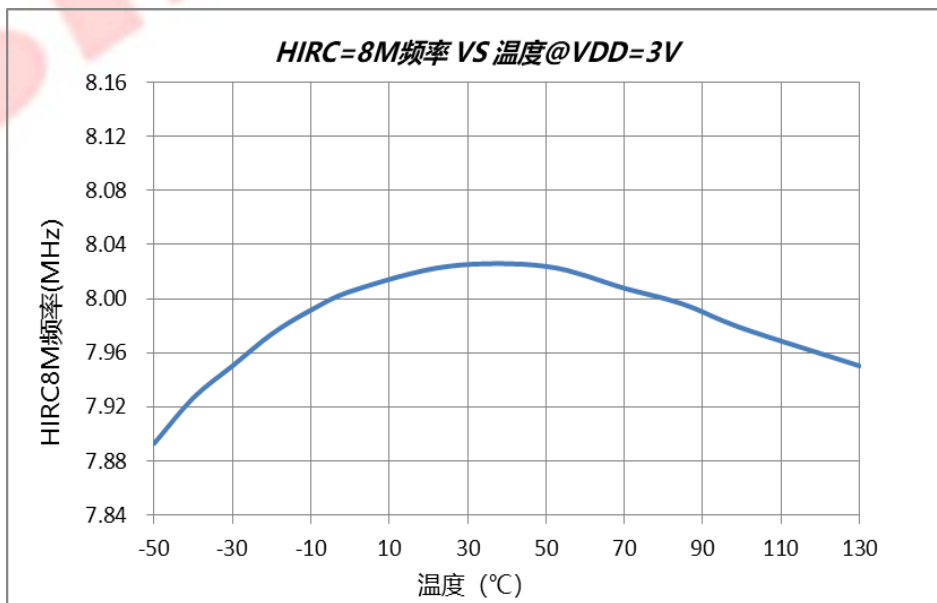
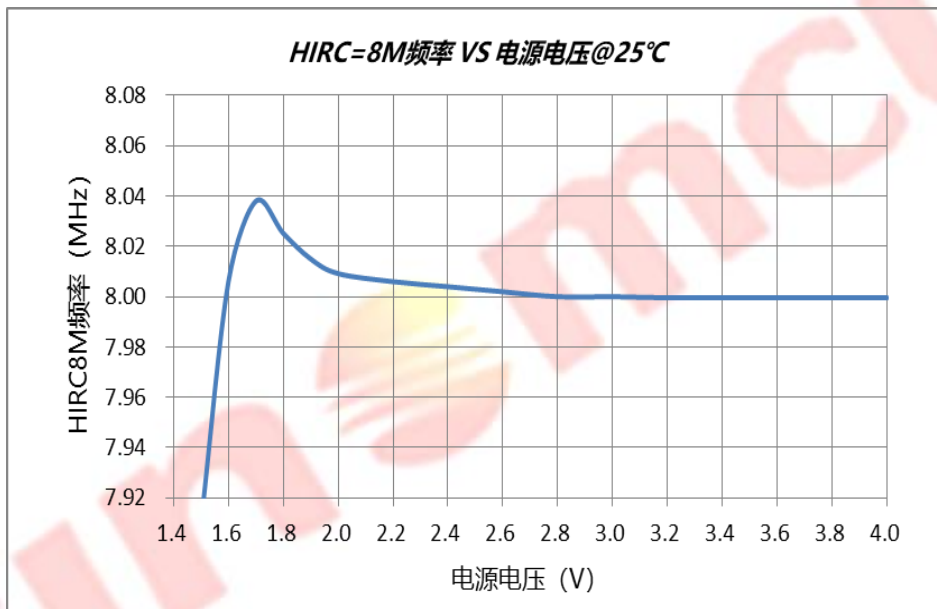
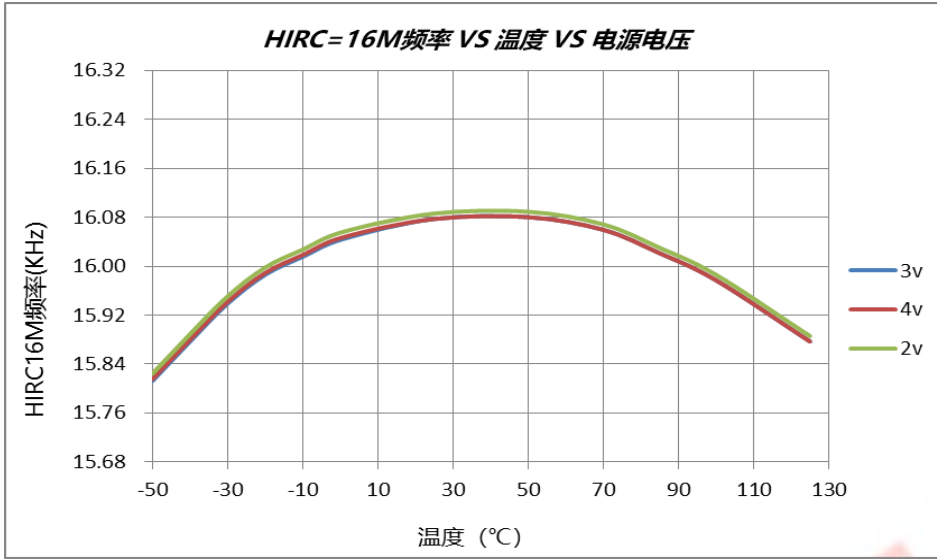


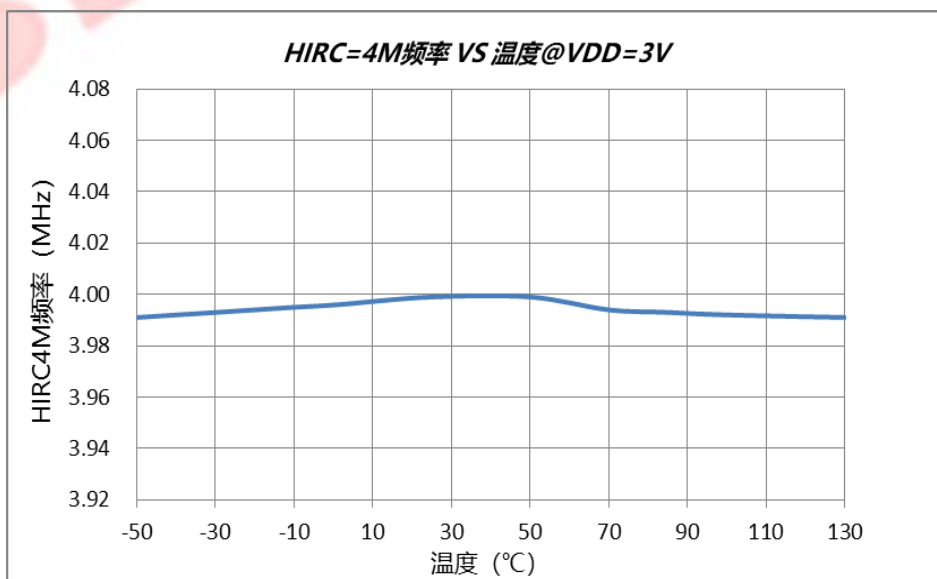
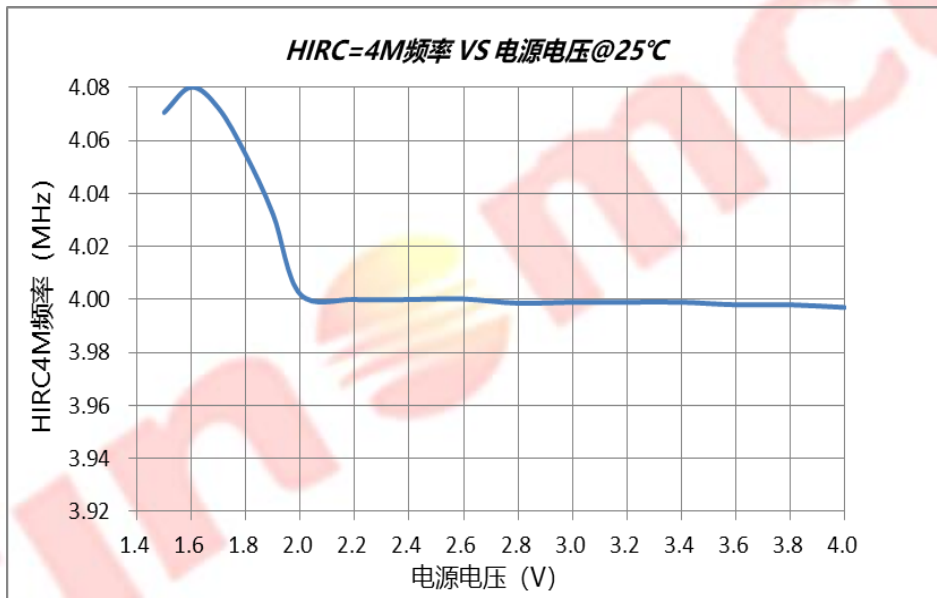
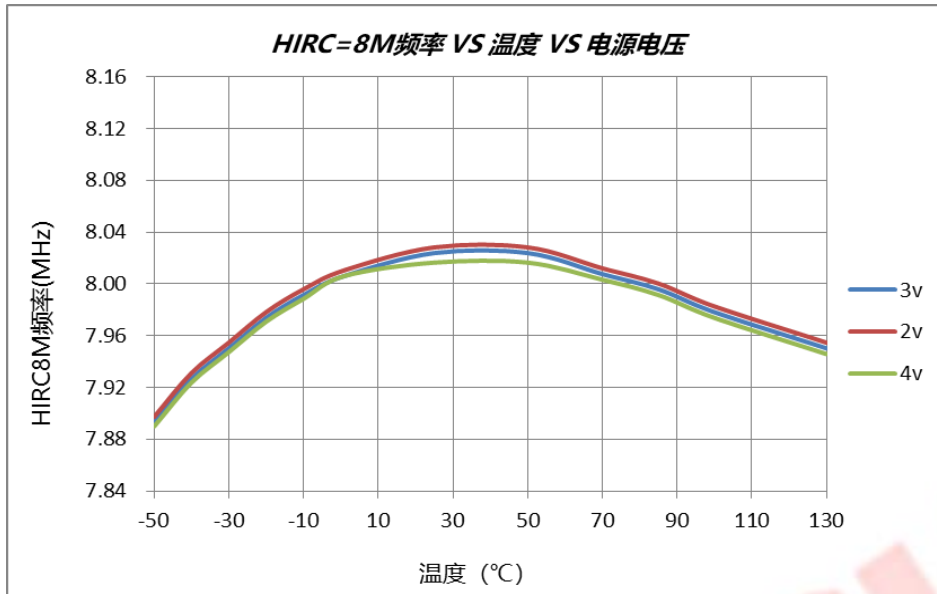


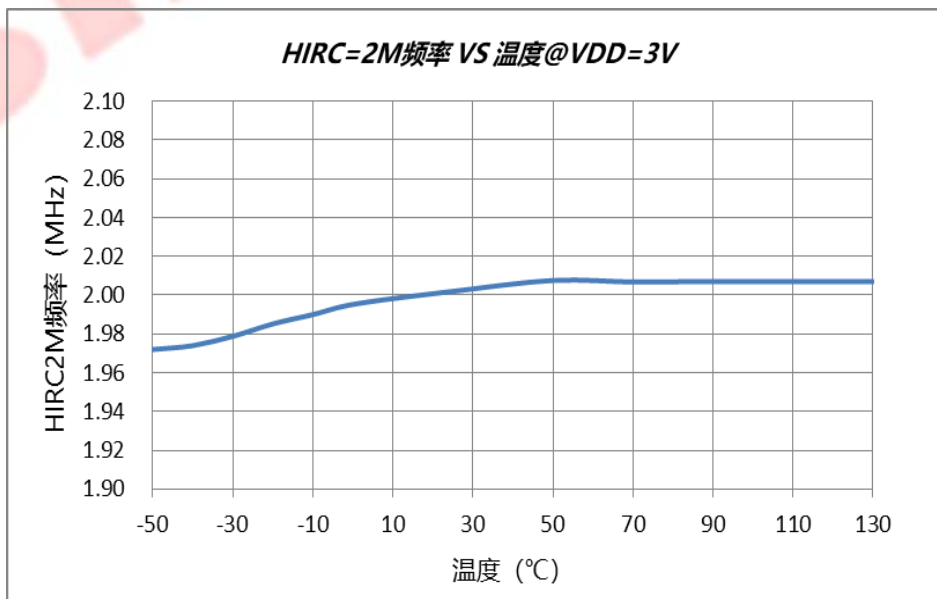
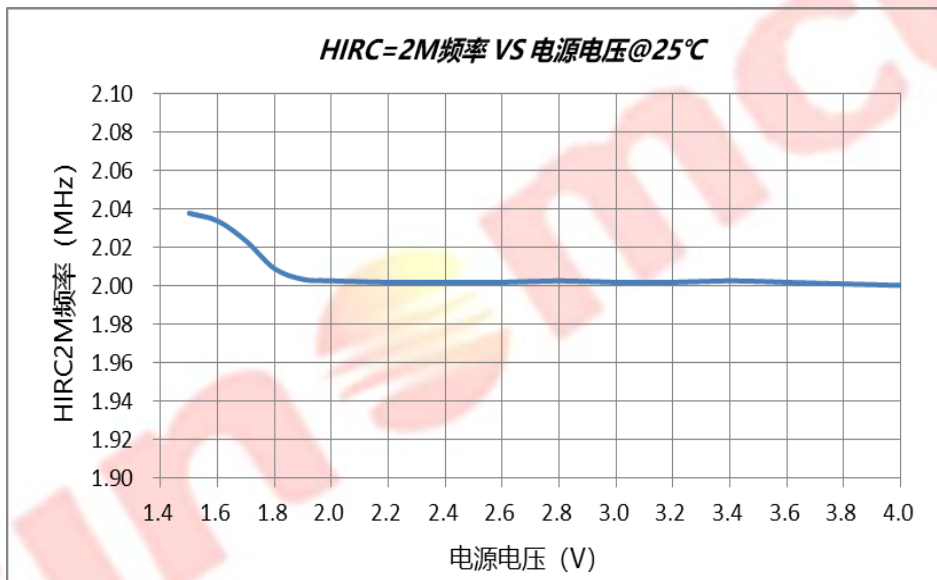
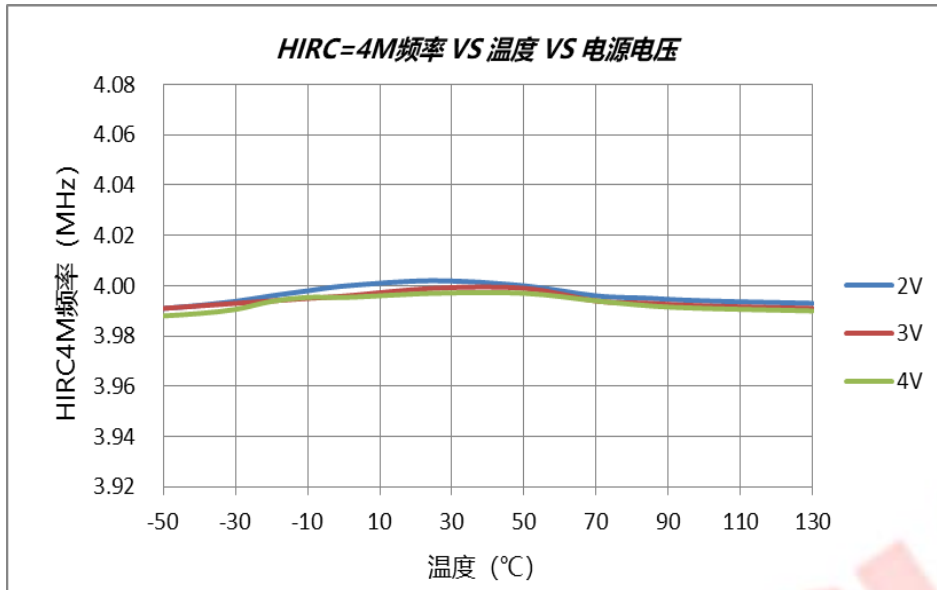
### 13.3 模块特性

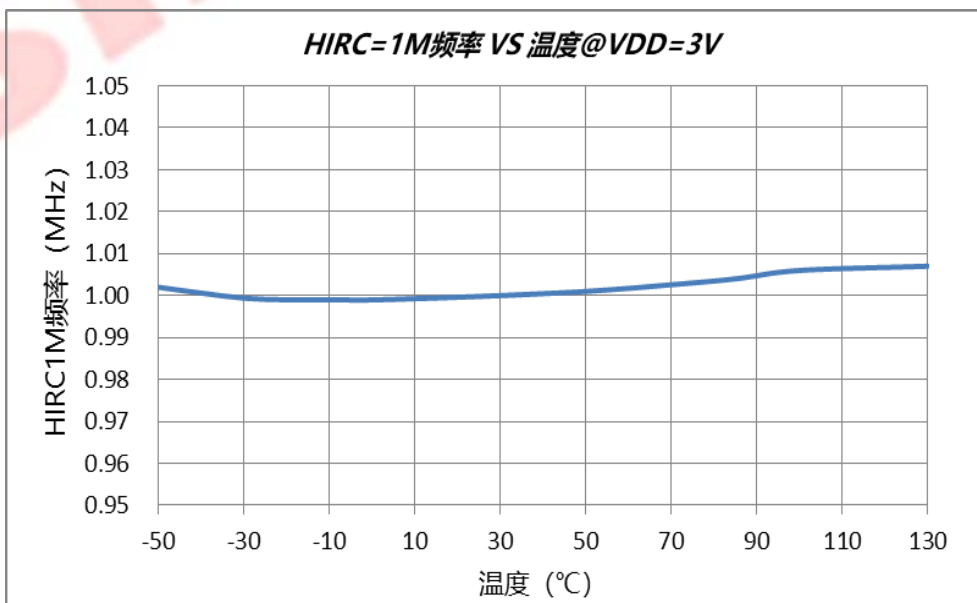
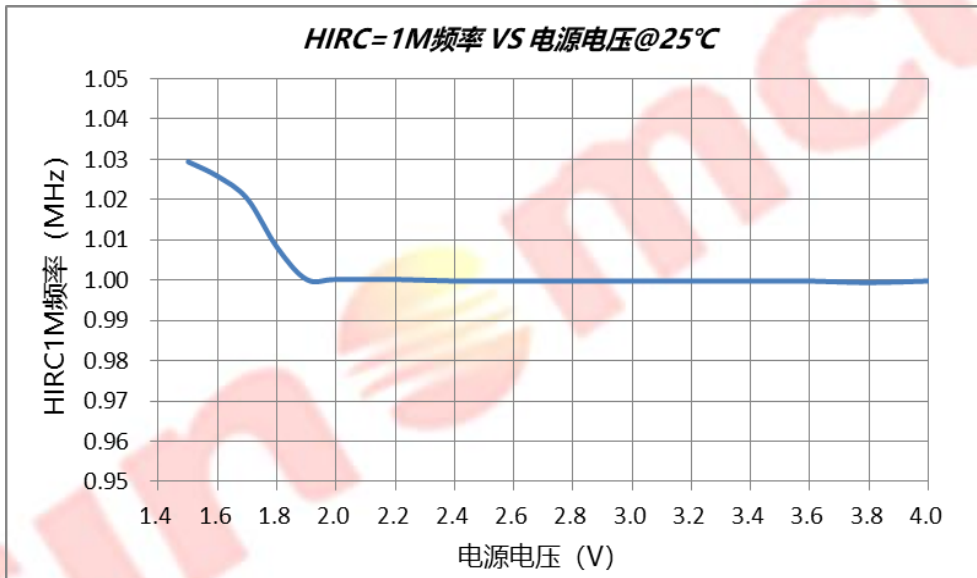
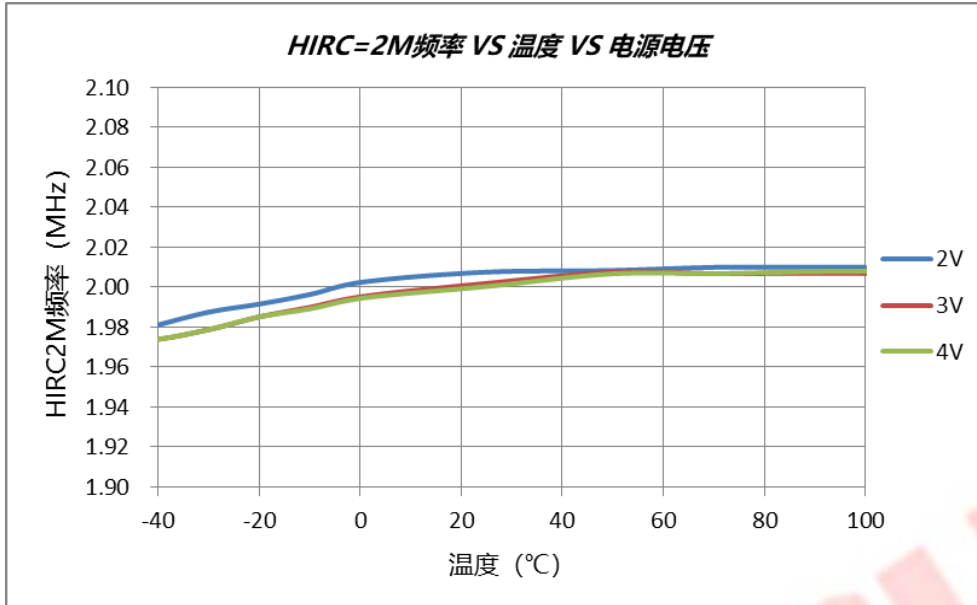
#### HIRC 频率 VS 电源电压/温度

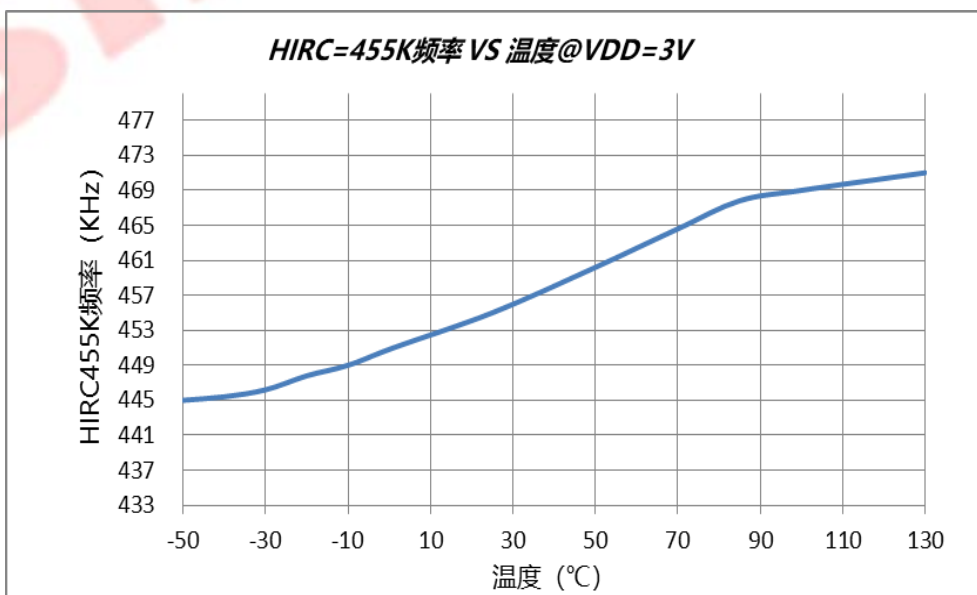
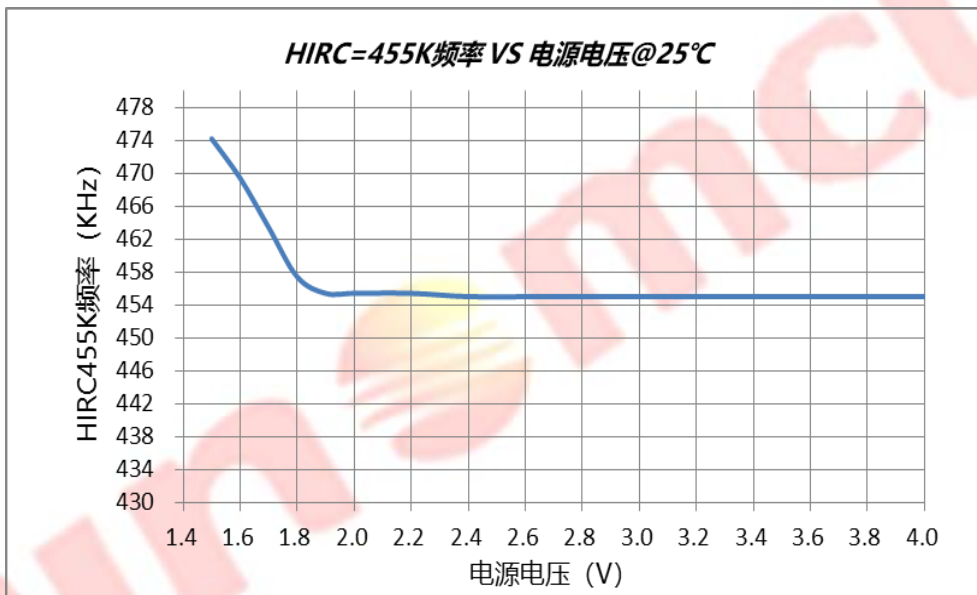
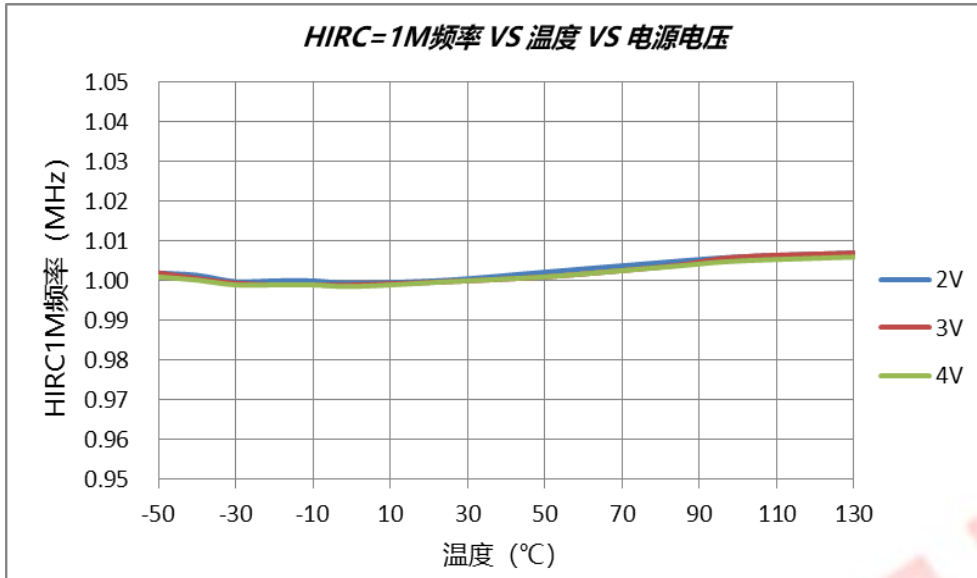


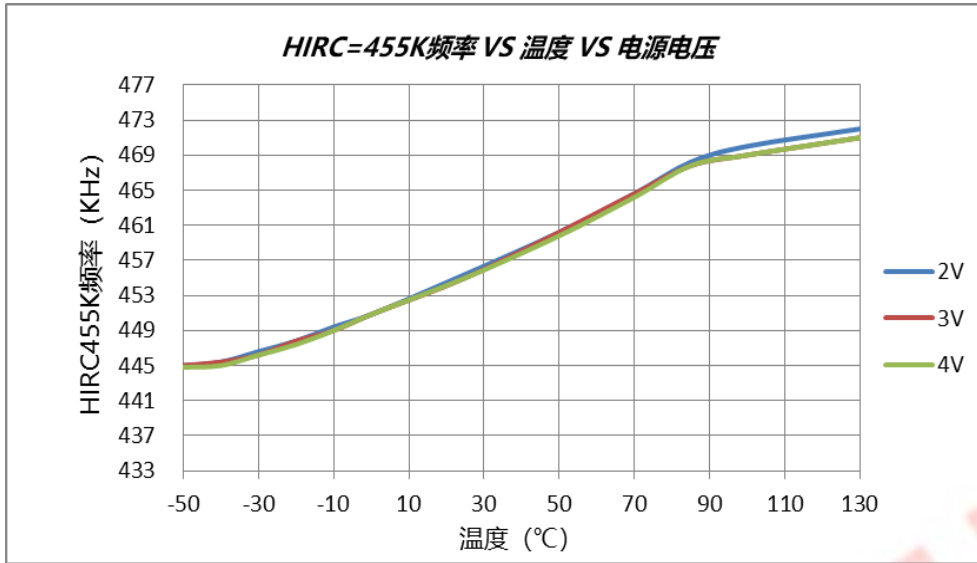




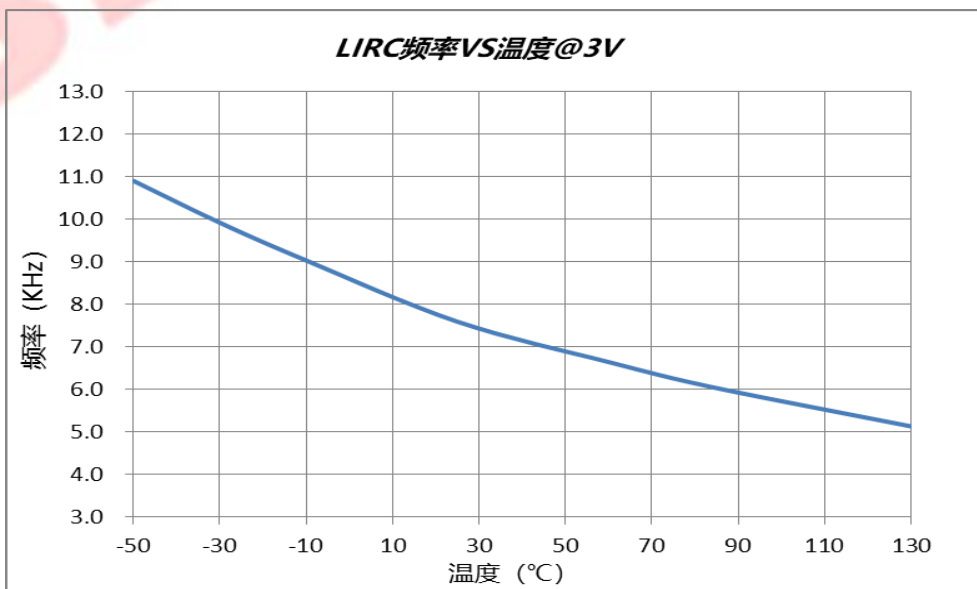
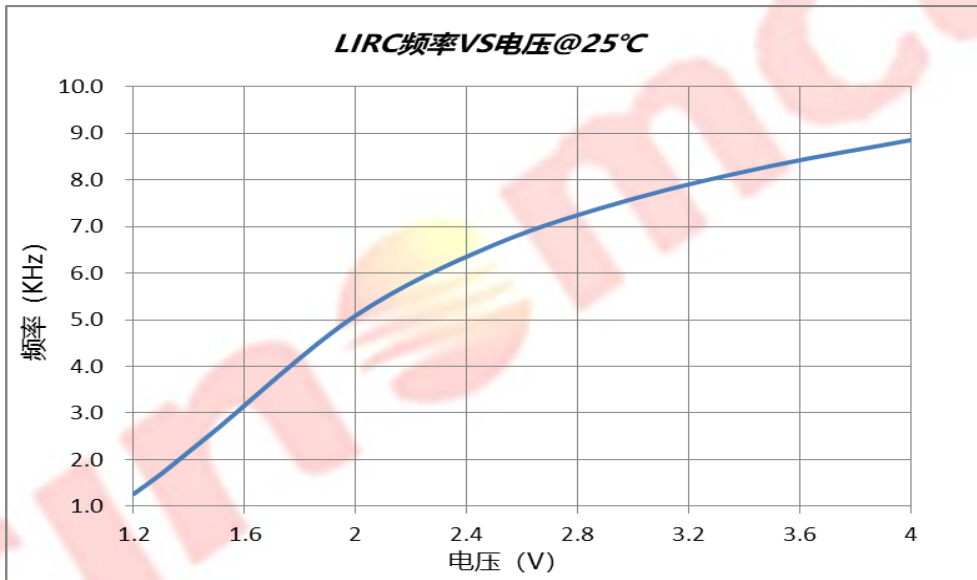




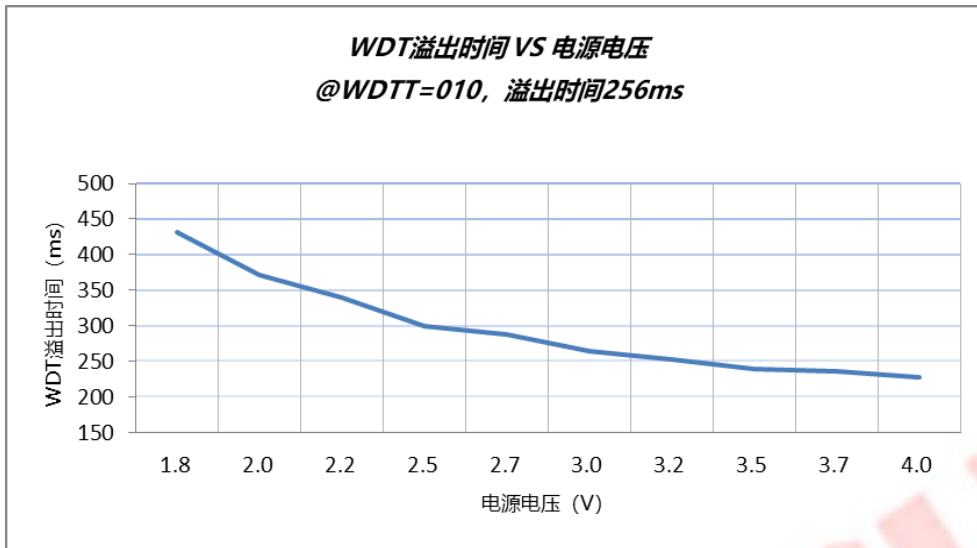




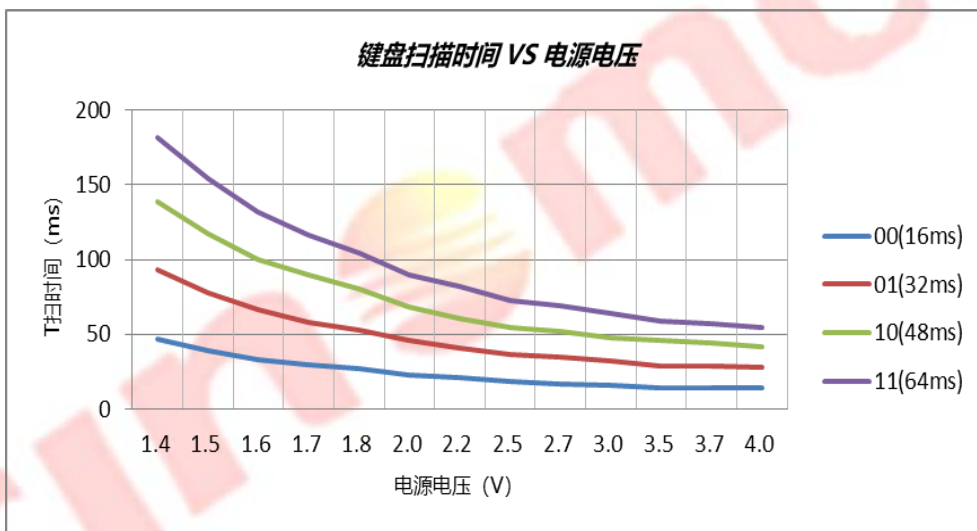
**LIRC 频率 VS 电源电压/温度**



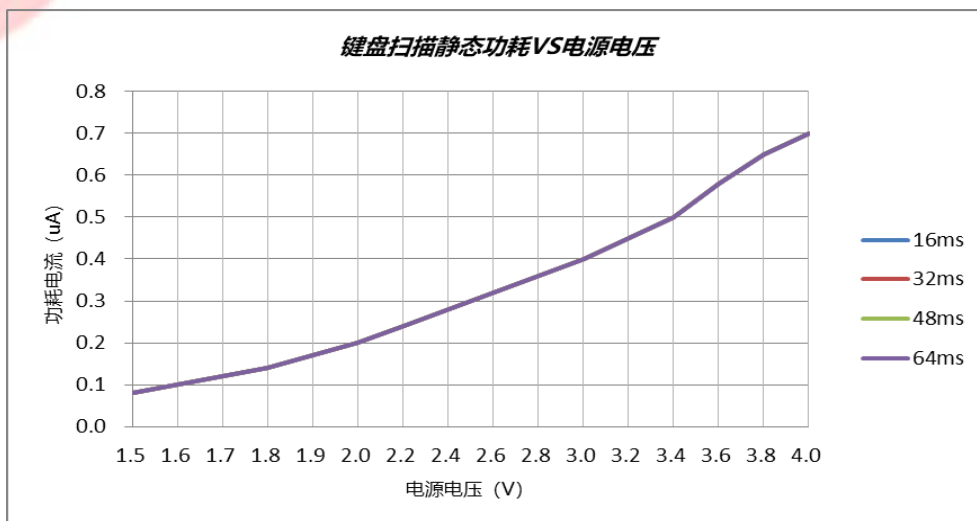
WDT 溢出时间 VS 电源电压



键盘扫描时间 VS 电源电压

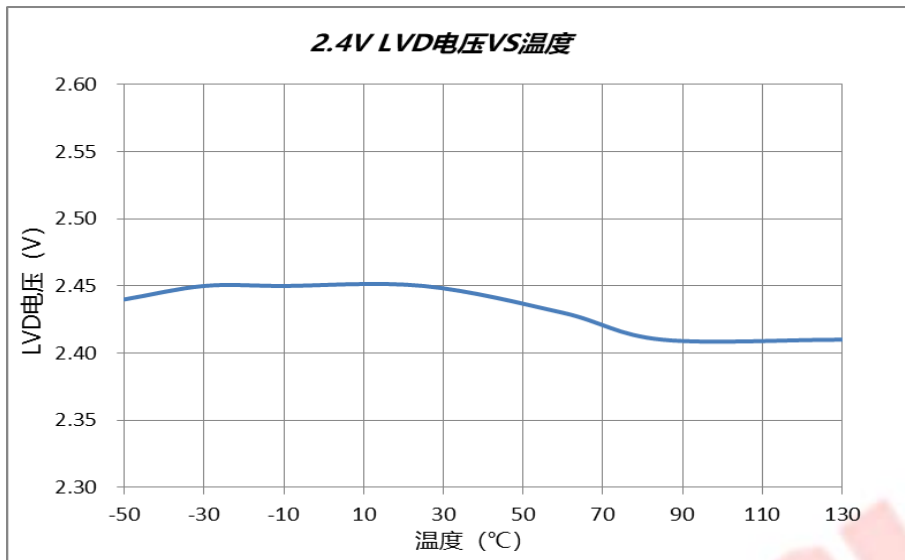


键盘扫描功耗 VS 电源电压



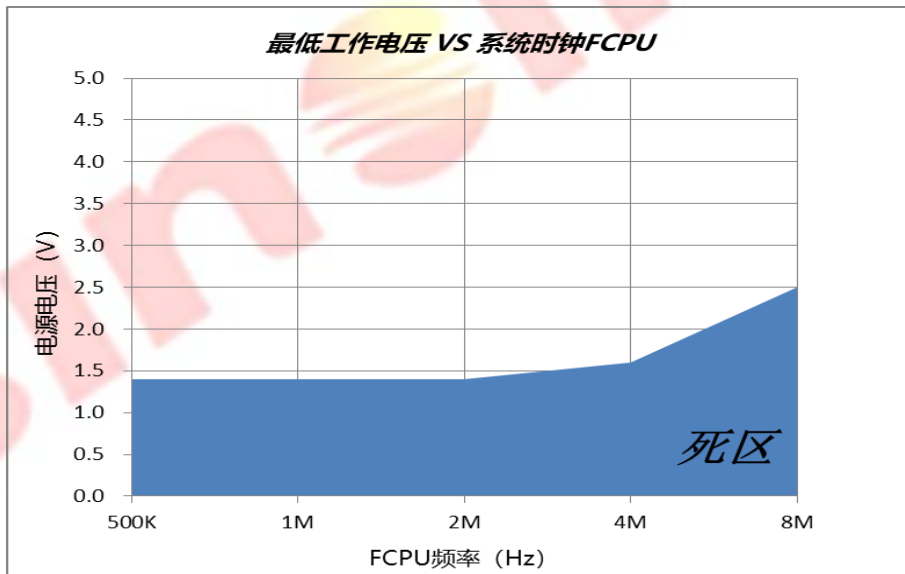


LVD 电压 VS 温度



最低工作电压 VS 系统时钟 F<sub>cpu</sub>

芯片最低工作电压和系统时钟 F<sub>cpu</sub> 有关，CPU 工作频率不同最低工作电压也不同。如下图所示，当 F<sub>cpu</sub> 提高时系统正常工作电压也随之提高，但由于 POR 电压固定（1.5V@25°C），在系统最低工作电压和 POR 电压之间就会出现一个电压区域，若电源电压在此区域内，则系统不能正常工作也不会产生 POR 复位，一般称之为死区，必须根据不同的 CPU 工作频率设置合适的 LVR 电压以避免出现死区。

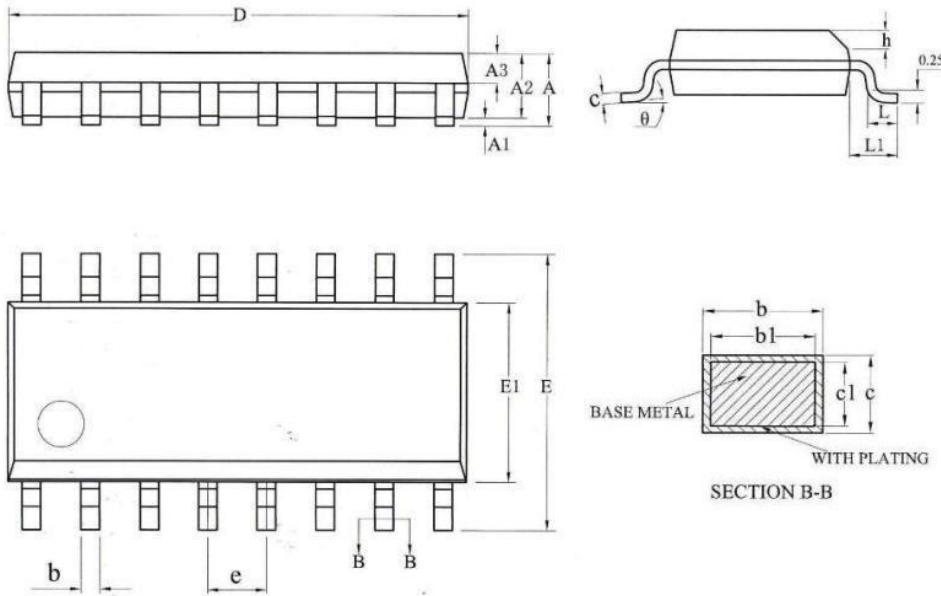


推荐 LVR 配置:

F <sub>cpu</sub>	死区电压	推荐 LVR
8MHz	2.5V	2.6V
4MHz	1.6V	1.9V
2MHz	1.4V	1.5V
1MHz	1.4V	1.5V
500KHz	1.4V	1.5V
32768Hz/2	1.4V	1.5V

## 14 封装尺寸

### 14.1 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		
θ	0	—	8°

## 15 修订记录

版本	修订日期	修订内容
V1.0	2015-11-25	初版发布;
V1.1	2018-01-12	修改 P17 设置输入状态的注释;
V1.2	2018-05-10	修订笔误;
V1.3	2020-06-03	调整文档格式与排版; 新增封装尺寸; 修订笔误; 新增低功耗模式未用引脚设置的注释;
V1.4	2021-04-20	调整 HIRC 频率特性参数;