

*SinoMCU 8 位单片机*

**MC30P6230**

**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        | 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  |
| 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       | 11 |
| 3.4 | 堆栈          | 12 |
| 3.5 | 控制寄存器       | 12 |
| 3.6 | 用户配置字       | 14 |
| 4   | 系统时钟        | 16 |
| 4.1 | 内部高频 RC 振荡器 | 16 |
| 4.2 | 内部低频 RC 振荡器 | 16 |
| 4.3 | 系统工作模式      | 17 |
| 4.4 | 低功耗模式       | 17 |
| 5   | 复位          | 18 |
| 5.1 | 复位条件        | 18 |
| 5.2 | 上电复位        | 19 |
| 5.3 | 低电压复位       | 19 |
| 5.4 | 看门狗复位       | 19 |
| 6   | I/O 端口      | 20 |
| 6.1 | 通用 I/O 功能   | 20 |
| 6.2 | 内部上/下拉电阻    | 20 |
| 6.3 | 端口模式控制      | 21 |
| 7   | 定时器 TIMER   | 22 |
| 7.1 | 看门狗定时器 WDT  | 22 |
| 7.2 | 定时器 T0      | 22 |
| 8   | 中断          | 25 |
| 8.1 | 外部中断        | 25 |
| 8.2 | 定时器中断       | 25 |
| 8.3 | 键盘中断        | 25 |
| 8.4 | 中断相关寄存器     | 26 |
| 9   | 特性曲线        | 28 |
| 9.1 | I/O 特性      | 28 |
| 9.2 | 功耗特性        | 31 |

---

|      |               |    |
|------|---------------|----|
| 9.3  | 模拟电路特性 .....  | 34 |
| 10   | 封装尺寸 .....    | 38 |
| 10.1 | SOP8.....     | 38 |
| 10.2 | DIP8 .....    | 38 |
| 10.3 | SOT23-6 ..... | 39 |
| 11   | 修订记录 .....    | 40 |

## 1 产品概要

### 1.1 产品特性

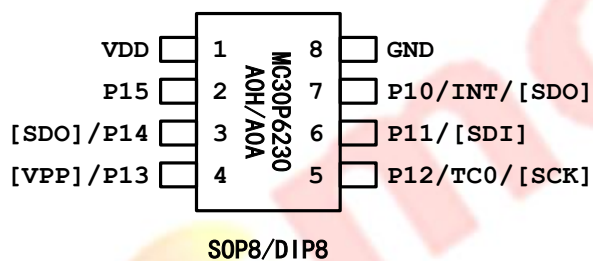
- 8 位 CPU 内核
  - ◇ 精简指令集，5 级深度硬件堆栈
  - ◇ CPU 为单时钟，仅在系统主时钟下运行
  - ◇ 系统主时钟下 F<sub>cpu</sub> 固定为 2 分频
- 程序存储器
  - ◇ 1K×14 位 OTP 型程序存储器（烧录 1 次）
  - ◇ 0.5K×14 位 OTP 型程序存储器（烧录 2 次）
- 数据存储器
  - ◇ 50 字节 SRAM/REG 型通用数据存储器，支持直接寻址、间接寻址等多种寻址方式
- 1 组共 6 个 I/O
  - ◇ P1 (P10~P15)
  - ◇ P13 为输入/开漏输出口，编程时为高压 V<sub>PP</sub> 输入
  - ◇ P10~P15 内置上拉电阻，P10~P12 内置下拉电阻，均可单独使能
  - ◇ P10~P12/P14/P15 可选开漏或推挽输出
  - ◇ P1 所有端口均支持键盘中断唤醒功能，并可单独使能
- 系统时钟源
  - ◇ 内置高频 RC 振荡器（8MHz/4MHz/2MHz/1MHz/455KHz），可用作系统主时钟源
  - ◇ 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
  - ◇ 运行模式：CPU 在系统主时钟下运行
  - ◇ 休眠模式：CPU 停止运行，系统主时钟源停止工作
- 内部自振式看门狗计数器（WDT）
  - ◇ 与定时器 T0 共用预分频器
  - ◇ 溢出时间可配置：4.5ms/18ms/72ms/288ms（无预分频）
  - ◇ 工作模式可配置：开启 WDT、关闭 WDT，也可软件控制开启或关闭
- 1 个定时器
  - ◇ 8 位定时器 T0，支持系统低频时钟，可实现外部计数功能，与 WDT 共用预分频器
- 中断
  - ◇ 外部中断（INT），键盘中断（P10~P15）
  - ◇ 定时器中断（T0）
- 低电压复位 LVR
  - ◇ 2.0V/2.7V/3.0V/3.6V
- 工作电压
  - ◇ V<sub>LVR27</sub> ~ 5.5V @ F<sub>cpu</sub> = 0~4MHz
  - ◇ V<sub>LVR20</sub> ~ 5.5V @ F<sub>cpu</sub> = 0~2MHz
- 封装形式
  - ◇ SOP8/DIP8/SOT23-6

## 1.2 订购信息

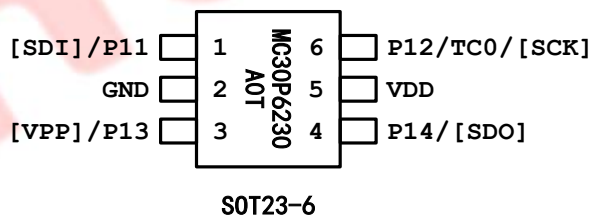
| 产品名称         | 封装形式    | 备注 |
|--------------|---------|----|
| MC30P6230A0H | SOP8    |    |
| MC30P6230A0A | DIP8    |    |
| MC30P6230A0T | SOT23-6 |    |
| MC30P6230A1T | SOT23-6 |    |

## 1.3 引脚排列

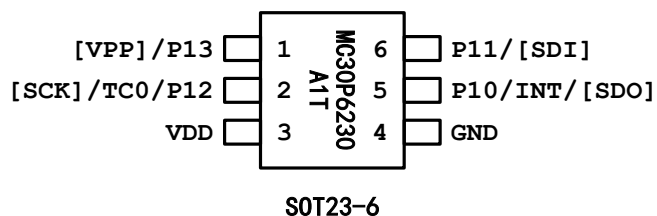
### MC30P6230A0H/A0A



### MC30P6230A0T



### MC30P6230A1T



## 1.4 端口说明

| 端口名称          | 类型 | 功能说明                     |
|---------------|----|--------------------------|
| VDD           | P  | 电源                       |
| GND           | P  | 地                        |
| P10~P12       | D  | GPIO (可选推挽/开漏输出), 内部上/下拉 |
| P14, P15      | D  | GPIO (可选推挽/开漏输出), 内部上拉   |
| P13           | D  | GPIO (开漏输出), 内部上拉        |
| INT           | DI | 外部中断输入                   |
| TC0           | DI | 定时器 T0 的外部计数输入           |
| 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     | -0.3~6.0     | 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 |
| 流入 VDD 最大电流 | IVDDmax | 50           | mA |
| 流出 GND 最大电流 | IGNDmax | 50           | mA |

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=5V, 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=0~4MHz              | VLVR27 |     | 5.5    | V  |
|        |       |         | Fcpu=0~2MHz              | VLVR20 |     | 5.5    |    |
|        |       |         | Fcpu=0~1MHz              | VLVR20 |     | 5.5    |    |
|        |       |         | Fcpu=0~500KHz            | VLVR20 |     | 5.5    |    |
|        |       |         | Fcpu=0~455KHz/2          | VLVR20 |     | 5.5    |    |
| 输入漏电流  | Ileak | 所有输入脚   | VDD=5V                   | -1     |     | 1      | uA |
| 输入高电平  | Vih   | 所有输入脚   | SMT 开启                   | 0.8VDD |     |        | V  |
|        |       |         | SMT 关闭                   | 2.0    |     |        | V  |
| 输入低电平  | Vil   | 所有输入脚   | SMT 开启                   |        |     | 0.2VDD | V  |
|        |       |         | SMT 关闭                   |        |     | 1.0    | V  |
| 输出拉电流  | Ioh   | 推挽输出脚   | Voh=4.3V                 | 6      |     |        | mA |
| 输出灌电流  | Iol   | 所有输出脚   | Vol=0.7V                 | 10     |     |        | mA |
| 上拉电阻   | Rpu   | P1      | VDD=5V, Vin=0            | 40     | 80  | 160    | KΩ |
| 下拉电阻   | Rpd   | P10-P12 | Vin=VDD=5V               | 60     | 125 | 250    | KΩ |
| 运行模式功耗 | Irun  | VDD     | Fcpu=4MHz@HIRC(8M)       |        | 1.5 |        | mA |
|        |       |         | Fcpu=2MHz@HIRC(4M)       |        | 1.0 |        | mA |
|        |       |         | Fcpu=1MHz@HIRC(2M)       |        | 800 |        | uA |
|        |       |         | Fcpu=500KHz@HIRC(1M)     |        | 700 |        | uA |
|        |       |         | Fcpu=227.5KHz@HIRC(455K) |        | 600 |        | uA |
| 休眠模式功耗 | Istop | VDD     | VDD=5V, WDT 关, LVR 关     |        | 0.2 | 1      | uA |
|        |       |         | VDD=5V, WDT 关, LVR 开     |        | 10  | 20     | uA |

|          |        |     |                      |      |     |      |    |
|----------|--------|-----|----------------------|------|-----|------|----|
|          |        |     | VDD=5V, WDT 开, LVR 关 |      | 2   | 4    | uA |
|          |        |     | VDD=3V, WDT 开, LVR 关 |      | 1   | 2    | uA |
| 低压复位电压   | VLVR20 | VDD |                      | -15% | 2.0 | +15% | V  |
|          | VLVR27 |     |                      | -15% | 2.7 | +15% |    |
|          | VLVR30 |     |                      | -15% | 3.0 | +15% |    |
|          | VLVR36 |     |                      | -15% | 3.6 | +15% |    |
| LVR 回滞电压 |        | VDD |                      |      | 6%  | 12%  |    |

注:

- 1、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、低温下 LVR 偏高, 具体参见特性曲线;

## 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=5V, T=25°C,<br>VDD 外接 0.1uF 对地电容              | -3%  | 455K- | +3%  | Hz  |
|           |                   | VDD=2.0V~5.0V, T=-20°C~70°C,<br>VDD 外接 0.1uF 对地电容 | -6%  | 8M    | +6%  |     |
| LIRC 振荡频率 | F <sub>LIRC</sub> | VDD=5V, T=25°C                                    | -50% | 32    | +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         | $R + ACC \rightarrow ACC$                                                       | 1  | 1  | C,DC,Z |
| ADDRA R  | R 和 ACC 相加，结果存入 R           | $R + ACC \rightarrow R$                                                         | 1  | 1  | C,DC,Z |
| ADCAR R  | R 和 ACC 相加（带 C 标志），结果存入 ACC | $R + ACC + C \rightarrow ACC$                                                   | 1  | 1  | C,DC,Z |
| ADCRA R  | R 和 ACC 相加（带 C 标志），结果存入 R   | $R + ACC + C \rightarrow R$                                                     | 1  | 1  | C,DC,Z |
| RSUBAR R | R 和 ACC 相减，结果存入 ACC         | $R - ACC \rightarrow ACC$                                                       | 1  | 1  | C,DC,Z |
| RSUBRA R | R 和 ACC 相减，结果存入 R           | $R - ACC \rightarrow R$                                                         | 1  | 1  | C,DC,Z |
| RSBCAR R | R 和 ACC 相减（带 C 标志），结果存入 ACC | $R - ACC - C \rightarrow ACC$                                                   | 1  | 1  | C,DC,Z |
| RSBCRA R | R 和 ACC 相减（带 C 标志），结果存入 R   | $R - ACC - C \rightarrow R$                                                     | 1  | 1  | C,DC,Z |
| ANDAR R  | R 和 ACC 与操作，结果存入 ACC        | $R \text{ and } ACC \rightarrow ACC$                                            | 1  | 1  | Z      |
| ANDRA R  | R 和 ACC 与操作，结果存入 R          | $R \text{ and } ACC \rightarrow R$                                              | 1  | 1  | Z      |
| ORAR R   | R 和 ACC 或操作，结果存入 ACC        | $R \text{ or } ACC \rightarrow ACC$                                             | 1  | 1  | Z      |
| ORRA R   | R 和 ACC 或操作，结果存入 R          | $R \text{ or } ACC \rightarrow R$                                               | 1  | 1  | Z      |
| XORAR R  | R 和 ACC 异或操作，结果存入 ACC       | $R \text{ xor } ACC \rightarrow ACC$                                            | 1  | 1  | Z      |
| XORRA R  | R 和 ACC 异或操作，结果存入 R         | $R \text{ xor } ACC \rightarrow R$                                              | 1  | 1  | Z      |
| COMAR R  | 对 R 取反，结果存入 ACC             | $R \text{ 取反} \rightarrow ACC$                                                  | 1  | 1  | Z      |
| COMR R   | 对 R 取反，结果存入 R               | $R \text{ 取反} \rightarrow R$                                                    | 1  | 1  | Z      |
| RLAR R   | R 循环左移（带 C 标志），结果存入 ACC     | $R[7] \rightarrow C$<br>$R[6:0] \rightarrow ACC[7:1]$<br>$C \rightarrow ACC[0]$ | 1  | 1  | C      |
| RLR R    | R 循环左移（带 C 标志），结果存入 R       | $R[7] \rightarrow C$<br>$R[6:0] \rightarrow R[7:1]$<br>$C \rightarrow R[0]$     | 1  | 1  | C      |
| RRAR R   | R 循环右移（带 C 标志），结果存入 ACC     | $R[0] \rightarrow C$<br>$R[7:1] \rightarrow ACC[6:0]$<br>$C \rightarrow ACC[7]$ | 1  | 1  | C      |
| RRR R    | R 循环右移（带 C 标志），结果存入 R       | $R[0] \rightarrow C$<br>$R[7:1] \rightarrow R[6:0]$<br>$C \rightarrow R[7]$     | 1  | 1  | C      |
| SWAPAR R | 交换 R 的高低半字节，结果存入 ACC        | $R[7:4] \rightarrow ACC[3:0]$<br>$R[3:0] \rightarrow ACC[7:4]$                  | 1  | 1  | -      |
| SWAPR R  | 交换 R 的高低半字节，结果存入 R          | $R[7:4] \rightarrow R[3:0]$<br>$R[3:0] \rightarrow R[7:4]$                      | 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      |
| 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      |
| 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      |
| 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 | -      |
| 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 |
| ISUBAI K   | K 和 ACC 相减, 结果存入 ACC            | K-ACC→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<br>K→PC[10:0]   | 2   | 1 | -      |
| GOTO K     | 无条件跳转                           | K→PC[10:0]               | 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<br>K→ACC          | 2   | 1 | -      |
| RETIE      | 从中断返回                           | TOS→PC<br>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，可通过配置字 PAGE 选择存储器的地址空间范围。

1K×14 位的地址空间范围为 0000H~03FFH，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<br>(0001H - 0007H) |
| 中断入口地址 (0008H)           |
| 通用程序区<br>(0009H - 03FFH) |

0.5K×14 位的地址空间范围为 0000H~01FFH，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<br>(0001H - 0007H) |
| 中断入口地址 (0008H)           |
| 通用程序区<br>(0009H - 01FFH) |

### 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/SFR0 可直接寻址或通过 INDF 间接寻址，SFR1 仅支持直接寻址。

**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  |
|---------|------|----------|--------|--------|--------|-------|-------|------|------|
| 00H-07H | SFR0 | INDF     | T0CNT  | PCL    | STATUS | FSR   | GPR0  | P1   | GPR1 |
| 08H-0FH |      | MCR      | P1KBCR | PCLATH | PDCON  | ODCON | PUCON | INTE | INTF |
| 10H-3FH | GPR  | 通用数据存储器区 |        |        |        |       |       |      |      |
| 40H-47H | SFR1 |          | T0CR   |        |        |       |       | DDR1 |      |
| 48H-4FH |      |          |        |        | TM0CR  |       |       |      |      |
| 50H-7FH | 保留   | 保留       |        |        |        |       |       |      |      |

**注：**

- 1、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；
- 2、SFR 中的 GPR0 (05H)、GPR1 (07H) 可用作通用数据存储器 GPR；

### 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   |
|----|----|----|----|----|----|---|---|---|-------------|-----|---|---|---|---|---|--------|
| /  | /  | /  | /  | /  | /  | / | / | / | 来自指令的 7 位地址 |     |   |   |   |   |   | 直接寻址模式 |
| /  | /  | /  | /  | /  | /  | / | / | / | 0           | FSR |   |   |   |   |   | 间接寻址模式 |

直接寻址模式，是以指令的低 7 位为数据存储器地址，通过指令访问，寻址范围为 00H~7FH。例：  
通过直接寻址模式将数据 55H 写入数据存储器 10H 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，是以 FSR 为数据存储器地址指针，通过 INDF 访问，寻址范围为 00H~7FH。例：  
通过间接寻址模式将数据 55H 写入数据存储器 10H 地址

```
MOVAI    10H
MOVRA    FSR
MOVAI    55H
MOVRA    INDF          ; 将数据 55H 写入 FSR 指向的数据存储器中
```

## 3.4 堆栈

5 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>FSR</b> | -     | -     | FSR5  | FSR4  | FSR3  | FSR2  | FSR1  | FSR0  |
| <b>R/W</b> | R     | R     | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |
| <b>初始值</b> | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |

BIT[5:0] **FSR[5:0]** – 数据指针寄存器  
FSR：间接寻址模式的指针，仅低 6 位有效。

### 间接寻址寄存器

|             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>INDF</b> | INDF7 | INDF6 | INDF5 | INDF4 | INDF3 | INDF2 | INDF1 | INDF0 |
| <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] **INDF[7:0]** – 间接寻址寄存器  
INDF：INDF 不是物理寄存器，对 INDF 寻址实际是对 FSR 指向的数据存储器地址进行访问，从而实现间接寻址功能。

### 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

### 程序指针计数器高位缓存器

|               | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>PCLATH</b> | -     | -     | -     | -     | -     | -     | PCH1  | PCH0  |
| <b>R/W</b>    | -     | -     | -     | -     | -     | -     | R/W   | R/W   |
| <b>初始值</b>    | -     | -     | -     | -     | -     | -     | 0     | 0     |

BIT[1:0] **PCH[1:0]** – 程序指针计数器高 2 位缓存器

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 10 位；
- ✧ 返回指令 RETIE/RETURN/RETAI：PC = 堆栈栈顶（TOS）；

对 PCL 操作指令（兼容义隆模式，晟矽模式）：

- ✧ 对 PCL 操作的加法指令：PC = (PC[9:0]+ALU[7:0])；
- ✧ 对 PCL 操作的其他指令：PC = {PC[9:8]:ALU[7:0](ALU 运算结果)}；

对 PCL 操作指令（兼容菲林模式）：

- ✧ 对 PCL 操作指令：PC = {PCLATH[1:0]:ALU[7:0](ALU 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>STATUS</b> | RST   | -     | -     | TO    | PD    | Z     | DC    | C     |
| <b>R/W</b>    | R/W   | -     | -     | R     | R     | R/W   | R/W   | R/W   |
| <b>初始值</b>    | 0     | -     | -     | 1     | 1     | X     | X     | X     |

BIT[7] **RST** – 唤醒源标志位

- 0: 芯片通过其他唤醒源唤醒；
- 1: 芯片通过外部中断或键盘中断唤醒；

BIT[4] **TO** – 看门狗溢出标志位

- 0: 发生 WDT 溢出；
- 1: 上电复位，或执行 CLRWDT/STOP 指令；

BIT[3] **PD** – 进入低功耗模式标志位

- 0: 执行 STOP 指令；
- 1: 上电复位，或执行 CLRWDT 指令；



|               |                                                                |
|---------------|----------------------------------------------------------------|
| <b>RDPORT</b> | 输出口读操作设置：<br>输出口读操作将读取寄存器内容；      输出口读操作将读取端口电平状态；             |
| <b>MCUSEL</b> | 芯片模式选择：<br>兼容义隆模式；兼容菲林模式；晟矽模式；                                 |
| <b>ENCR</b>   | 程序代码加密设置：<br>程序代码加密；                                  程序代码不加密； |

### 芯片模式说明

| 功能              | 晟矽模式                                                                      | 兼容义隆模式           | 兼容菲林模式                                   |
|-----------------|---------------------------------------------------------------------------|------------------|------------------------------------------|
| 复位地址            | 000H                                                                      | 000H             | 存储空间最后地址                                 |
| 键盘中断允许位(KBIM)   | 有                                                                         | 无                | 有                                        |
| 外部中断触发沿         | 上升沿或下降沿                                                                   | 下降沿              | 上升沿或下降沿                                  |
| 指令操作 GIE        | 无指令限制                                                                     | 只能用 BSET/BCLR 指令 | 无指令限制                                    |
| PC 高位缓存器 PCLATH | 无                                                                         | 无                | 有                                        |
| 指令操作 PCL        | ADDRA 指令：PC={PC[9:0]+ALU[7:0]};<br>其它指令：PC={PC[9:8]:ALU[7:0]}; (ALU 运算结果) |                  | PC={PCLATH[1:0]:ALU[7:0]};<br>(ALU 运算结果) |



## 4 系统时钟

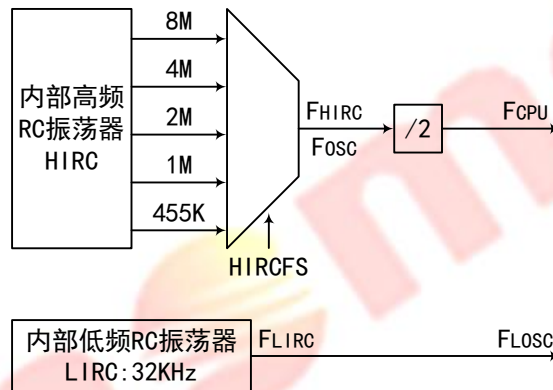
芯片为双时钟系统，内部电路均在系统主时钟 Fosc 或系统低频时钟 FLOSC 下工作，部分模块的时钟还可在 Fosc 和 FLOSC 之间切换。

系统主时钟 Fosc 为内部高频 RC 振荡器 HIRC（8MHz/4MHz/2MHz/1MHz/455KHz）时钟 FHIRC；系统低频时钟 FLOSC 为内部低频 RC 振荡器 LIRC（32KHz）时钟 FLIRC。

CPU 为单时钟，时钟源固定为系统主时钟 Fosc，CPU 的时钟频率 Fcpu 固定为 Fosc 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（8MHz/4MHz/2MHz/1MHz/455KHz）的高精度 HIRC 振荡器，可用作系统主时钟源。

### 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

芯片支持运行模式和休眠模式 2 种系统工作模式。

| 工作模式 | 切入条件             | 系统状态          |
|------|------------------|---------------|
| 运行模式 | 系统复位             | CPU 运行，主时钟源工作 |
|      | 休眠模式下，CPU 唤醒     |               |
| 休眠模式 | 运行模式下，执行 STOP 指令 | CPU 暂停，主时钟源停止 |

**注：**系统低频时钟源 LIRC 同时也是 WDT 时钟源，LIRC 是否工作由 WDT 决定，不受工作模式影响。

### 4.4 低功耗模式

芯片的低功耗模式即为休眠模式。

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：**

- 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

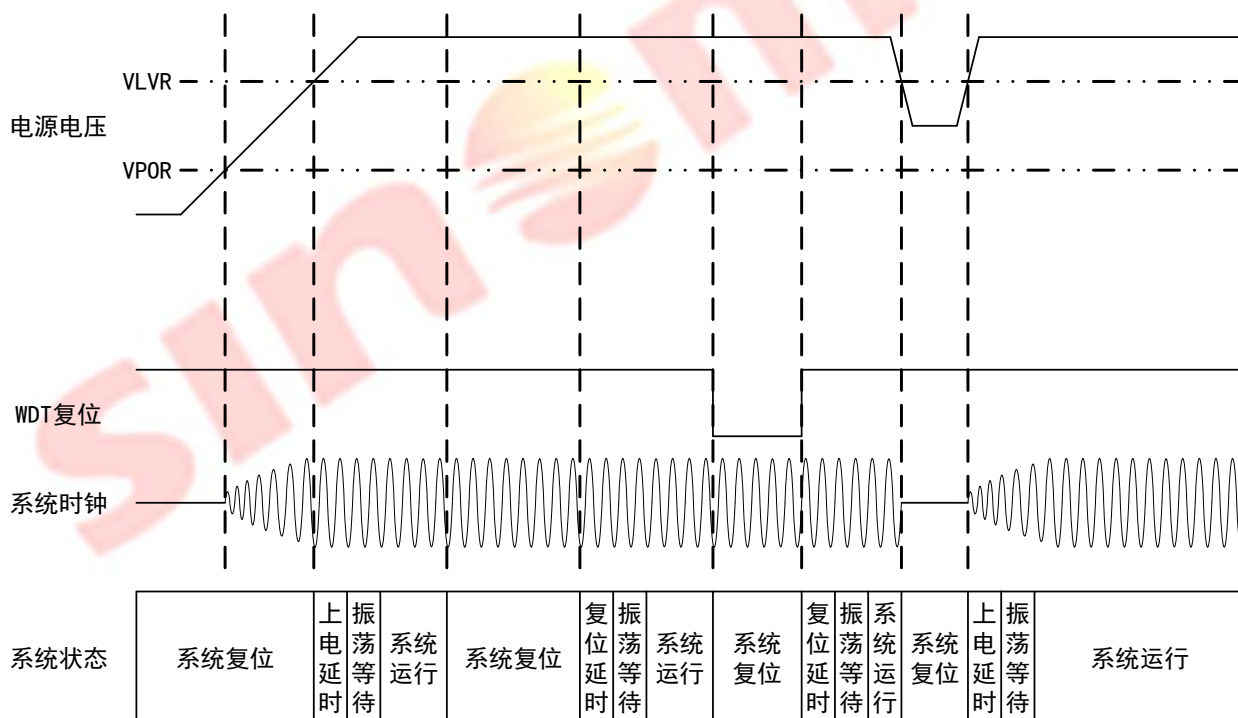
芯片共有如下几种复位方式：

- ✧ 上电复位 POR；
- ✧ 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 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：



**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，则应在复位后进行软件延时，以确保芯片开始工作时 VDD 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 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 6% 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 +6% 后 LVR 复位才解除。

## 5.4 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/输出端口为一组 6 位端口 P1。所有 I/O 端口均支持施密特输入，除 P13 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、键盘中断输入等复用功能。

#### 端口数据寄存器

|     | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| P1  | -     | -     | P15D  | P14D  | P13D  | P12D  | P11D  | P10D  |
| R/W | -     | -     | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |
| 初始值 | -     | -     | X     | X     | X     | X     | X     | X     |

BIT[5:0] P1nD – P1n 端口数据位 (n=5-0)

#### 端口方向寄存器

|      | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| DDR1 | -     | -     | DDR15 | DDR14 | DDR13 | DDR12 | DDR11 | DDR10 |
| R/W  | -     | -     | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |
| 初始值  | -     | -     | 1     | 1     | 1     | 1     | 1     | 1     |

BIT[5:0] DDR1n – P1n 端口方向控制位 (n=5-0)

- 0: 端口作为输出口，读端口操作可配置为读取端口数据寄存器值或端口输入电平状态；
- 1: 端口作为输入口，读端口操作将读取端口的输入电平状态；

### 6.2 内部上/下拉电阻

所有端口均有内部上拉或下拉电阻，且均有单独的寄存器位控制其上/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 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PUCON | -     | -     | P15PU | P14PU | P13PU | P12PU | P11PU | P10PU |
| R/W   | -     | -     | R/W   | R/W   | R/W   | R/W   | R/W   | R/W   |
| 初始值   | -     | -     | 1     | 1     | 1     | 1     | 1     | 1     |

BIT[5:0] P1nPU – P1n 端口上拉电阻控制位 (n=5-0)

- 0: 端口内部上拉电阻有效；
- 1: 端口内部上拉电阻无效；

### 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 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>PDCON</b> | -     | P12PD | P11PD | P10PD | -     | -     | -     | -     |
| <b>R/W</b>   | -     | R/W   | R/W   | R/W   | -     | -     | -     | -     |
| <b>初始值</b>   | -     | 1     | 1     | 1     | -     | -     | -     | -     |

BIT[6:4] **P1nPD** – P1n 端口下拉电阻控制位 (n=2-0)

0: 端口内部下拉电阻有效;

1: 端口内部下拉电阻无效;

## 6.3 端口模式控制

用作数字输出口时，除 P13 固定为开漏输出外，P1 其余端口可选择推挽输出或开漏输出。

### 端口输出模式寄存器

|              | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>ODCON</b> | -     | -     | P15OD | P14OD | -     | P12OD | P11OD | P10OD |
| <b>R/W</b>   | -     | -     | R/W   | R/W   | -     | R/W   | R/W   | R/W   |
| <b>初始值</b>   | -     | -     | 0     | 0     | -     | 0     | 0     | 0     |

BIT[5:4,2:0] **P1nOD** – P1n 端口输出模式选择位 (n=5-4,2-0)

0: 输出时端口为推挽输出;

1: 输出时端口为开漏输出;

**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。**

## 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，可通过预分频器选择不同的计数时钟频率。WDT 计数器溢出将复位芯片或唤醒 CPU。

可通过配置字 WDTM 和寄存器位 WDTEN 共同决定是否开启 WDT。当 WDTM 选择始终关闭或 WDTEN 为 0 时 WDT 定时器被关闭；当 WDTM 选择始终开启且 WDTEN 为 1 时 WDT 定时器才开启。若 WDT 定时器开启，则在休眠模式下 WDT 依然工作，溢出时将唤醒 CPU；而在 CPU 运行时 WDT 溢出，则将复位芯片。

WDT 和定时器 T0 共用一个预分频器，并通过寄存器位决定预分频器的分配。当预分频器分配给 T0 时，WDT 时钟不分频；而预分频器分配给 WDT 时，T0 时钟不分频。执行 CLRWDT 指令或 STOP 指令将清零 WDT 计数器，若预分频器分配给 WDT，则还将清零预分频计数器（预分频比不变）。

WDT 的基本溢出时间（即无预分频的时间）可配置为 4.5ms/18ms/72ms/288ms。

**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。**

#### 杂项控制寄存器

|     | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|
| MCR | WDTEN | EIS   | -     | -     | -     | -     | -     | -     |
| R/W | R/W   | R/W   | -     | -     | -     | -     | -     | -     |
| 初始值 | 1     | 0     | -     | -     | -     | -     | -     | -     |

BIT[7] **WDTEN** – 看门狗使能位

0: 关闭 WDT;

1: 开启 WDT;

BIT[6] **EIS** – INT 中断外部输入使能位

0: INT 中断外部输入无效，端口用作其他功能;

1: INT 中断外部输入有效，端口需设为输入;

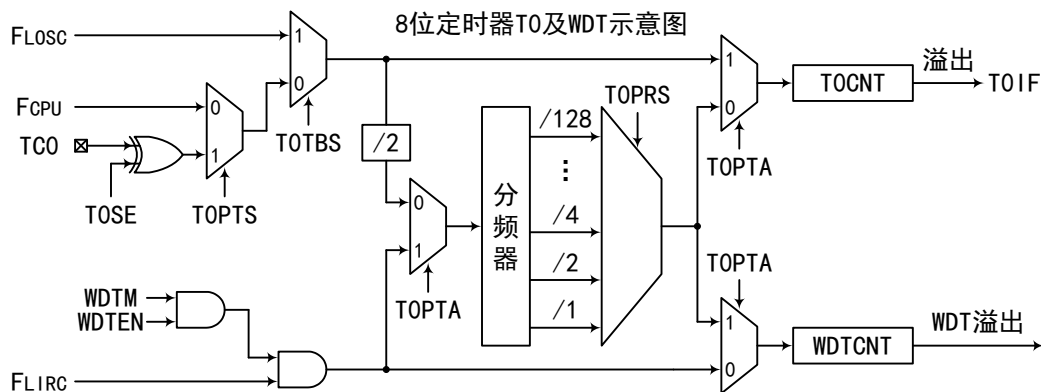
### 7.2 定时器 T0

定时器 T0 为 8 位定时/计数器，包含 1 个 8 位递增计数器、可编程预分频器、控制寄存器。

✧ 可通过预分频比设置计数频率;

✧ 时钟源可选：FLOSC、FCPU、外部时钟（TC0 输入）;

✧ 支持溢出中断和溢出唤醒功能;



TOCNT 为 8 位可读写的递增计数器，计数溢出到 0 时产生溢出信号并触发中断，中断标志 T0IF 将被置 1。预分频器为 T0 与 WDT 共用，通过寄存器位 T0PTA 控制预分频器的分配。T0PTA=0 时，预分频器分配给 T0 使用；T0PTA=1 时，预分频器分配给 WDT 使用。不管预分频器是否分配给 T0，写 TOCNT 都将清零预分频计数器，预分频比保持不变。T0 计数周期 = 预分频比 / T0 计数时钟频率。

通过 T0TBS 选择 FLOSC 作为 T0 时钟，则在低功耗模式下 T0 将继续工作，溢出可唤醒。

#### 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>TOCR</b> | -     | INTM  | T0PTS | T0SE  | T0PTA | T0PRS2 | T0PRS1 | T0PRS0 |
| <b>R/W</b>  | -     | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| <b>初始值</b>  | -     | 0     | 1     | 1     | 1     | 1      | 1      | 1      |

BIT[6] **INTM** – 外部中断 INT 触发方式选择位

- 0: 下降沿触发；
- 1: 上升沿触发；

BIT[5] **T0PTS** – T0 时钟源选择位

- 0: T0 时钟源为 F<sub>CPU</sub>；
- 1: T0 时钟源为 TC0 输入的外部时钟；

BIT[4] **T0SE** – T0 外部时钟计数沿选择位

- 0: 外部时钟上升沿计数；
- 1: 外部时钟下降沿计数；

BIT[3] **T0PTA** – 预分频器分配控制位

- 0: 预分频器分配给 T0；
- 1: 预分频器分配给 WDT；

BIT[2:0] **T0PRS[2:0]** – 预分频比选择位

| <b>T0PRS[2:0]</b> | <b>T0 时钟预分频比<br/>(T0PTA=0)</b> | <b>WDT 时钟预分频比<br/>(T0PTA=1)</b> |
|-------------------|--------------------------------|---------------------------------|
| 000               | 1: 2                           | 1: 1                            |



|     |        |        |
|-----|--------|--------|
| 001 | 1: 4   | 1: 2   |
| 010 | 1: 8   | 1: 4   |
| 011 | 1: 16  | 1: 8   |
| 100 | 1: 32  | 1: 16  |
| 101 | 1: 64  | 1: 32  |
| 110 | 1: 128 | 1: 64  |
| 111 | 1: 256 | 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>   | X      | X      | X      | X      | X      | X      | X      | X      |

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

#### 定时器控制寄存器

|              | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>TM0CR</b> | -     | T0TBS | -     | -     | -     | -     | -     | -     |
| <b>R/W</b>   | -     | R/W   | -     | -     | -     | -     | -     | -     |
| <b>初始值</b>   | -     | 0     | -     | -     | -     | -     | -     | -     |

BIT[6] **T0TBS** – T0 时钟源选择位

- 0: T0 时钟源由 T0PTS 决定;
- 1: T0 时钟源为系统低频时钟 FLOSC;



## 8 中断

芯片的中断源包括外部中断（INT）、定时器中断（T0）和键盘中断等。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 STATUS，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 STATUS，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。

**注：要使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。**

### 8.1 外部中断

芯片有 1 路外部中断源 INT，通过 INTM 可选择上升沿或下降沿等触发方式。外部中断触发时，中断标志 INTIF 将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位 INTIE 为 1，则产生外部中断。

**注：当 P10 端口通过寄存器位 EIS 置 1 复用为 INT 时，端口的键盘中断唤醒功能无效。**

### 8.2 定时器中断

定时器 T0 在计数溢出时触发定时器中断，中断标志 T0IF 将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位 T0IE 为 1，则产生定时器中断。

### 8.3 键盘中断

芯片有 6 路键盘中断源，均可通过寄存器位单独使能或屏蔽，任意一路使能的中断源的输入电平发生变化时，均会触发键盘中断，中断标志 KBIF 将被置 1，若中断总使能位 GIE 为 1 且键盘中断使能位 KBIE 为 1，则产生键盘中断。

## 键盘中断控制寄存器

|             | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| <b>KBIM</b> | -     | -     | KBIM5 | KBIM4 | KBIM3 | KBIM2 | KBIM1 | KBIM0 |
| <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] **KBIMn** – P1n 端口键盘中断使能位 (n=5-0)

0: 屏蔽端口键盘中断功能;

1: 使能端口键盘中断功能;

## 8.4 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> | GIE   | -     | -     | -     | -     | INTIE | KBIE  | TOIE  |
| <b>R/W</b>  | R/W   | -     | -     | -     | -     | R/W   | R/W   | R/W   |
| <b>初始值</b>  | 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[2] **INTIE** – INT 中断使能位

0: 屏蔽 INT 中断;

1: 使能 INT 中断;

BIT[1] **KBIE** – 键盘中断使能位

0: 屏蔽键盘中断;

1: 使能键盘中断;

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> | -     | -     | -     | -     | -     | INTIF | KBIF  | TOIF  |
| <b>R/W</b>  | -     | -     | -     | -     | -     | R/W   | R/W   | R/W   |
| <b>初始值</b>  | -     | -     | -     | -     | -     | 0     | 0     | 0     |

BIT[2] **INTIF** – INT 中断标志位

0: 未触发 INT 中断;

1: 已触发 INT 中断, 需软件清 0;

- BIT[1]      **KBIF** – 键盘中断标志位  
0: 未触发键盘中断;  
1: 已触发键盘中断, 需软件清 0;
- 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 0;

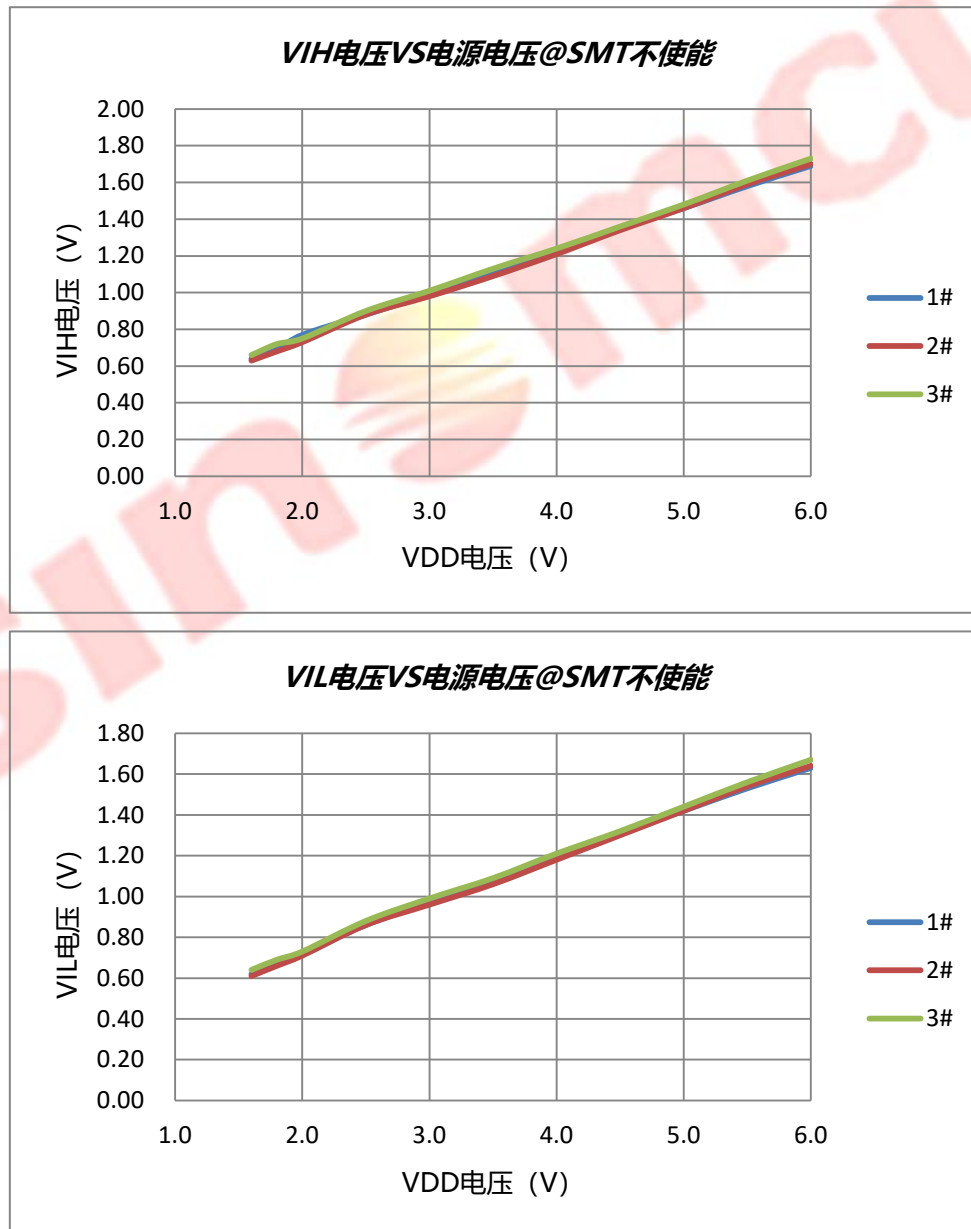
## 9 特性曲线

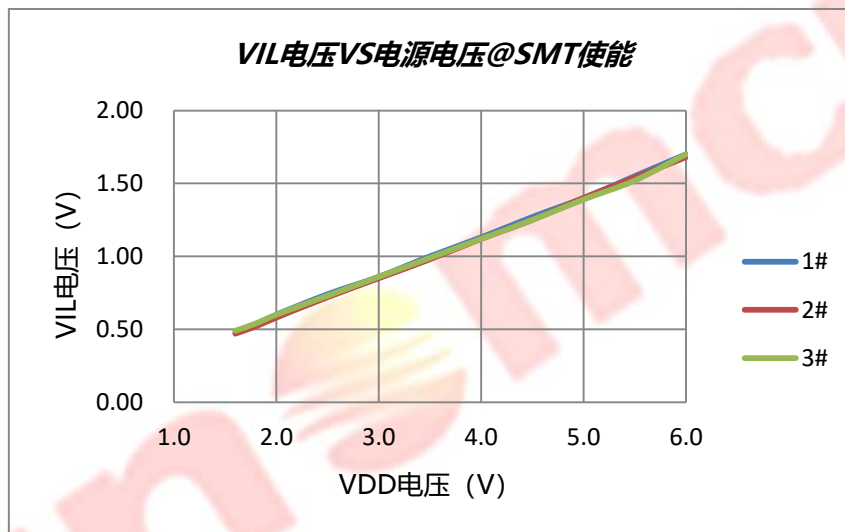
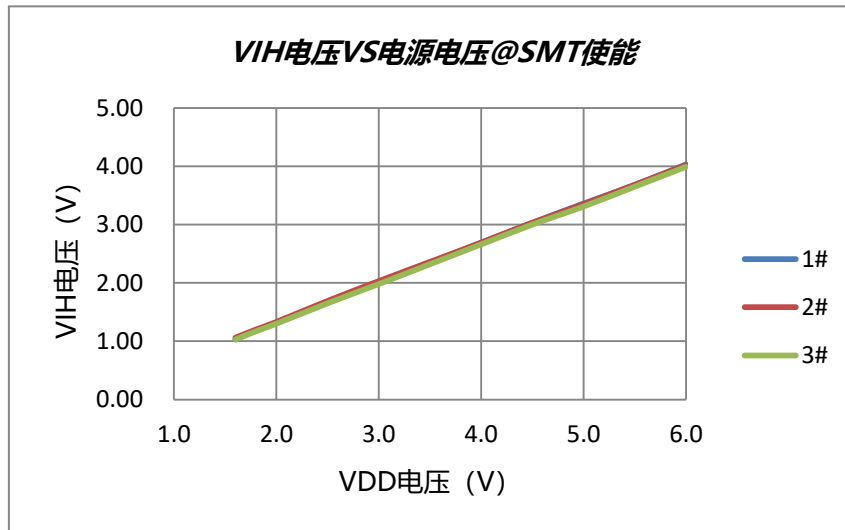
注:

- 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  $V_{DD}=5\text{V}$ ；

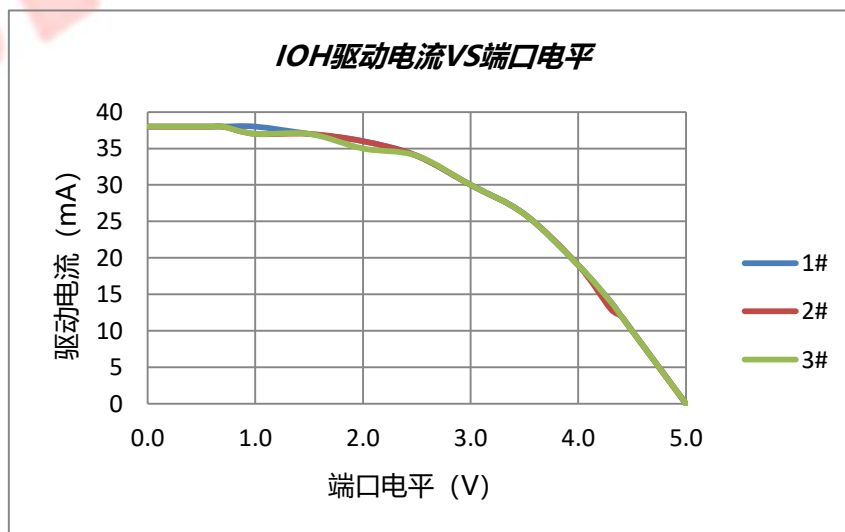
### 9.1 I/O 特性

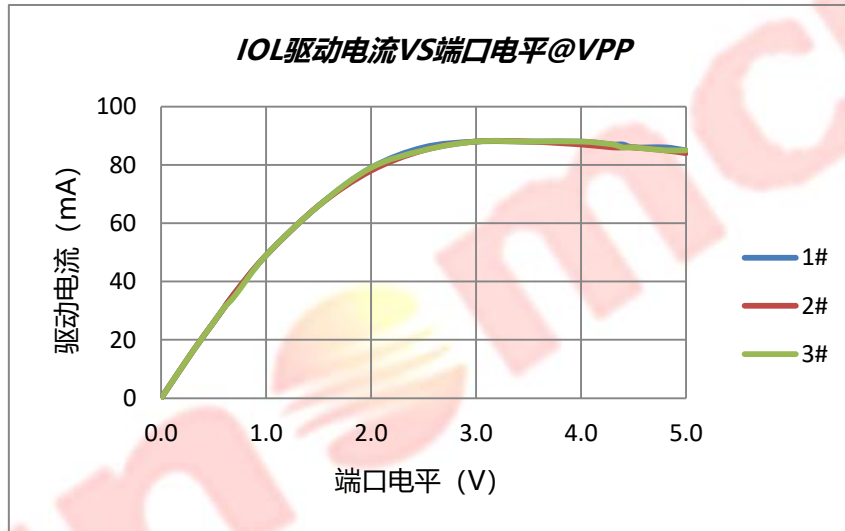
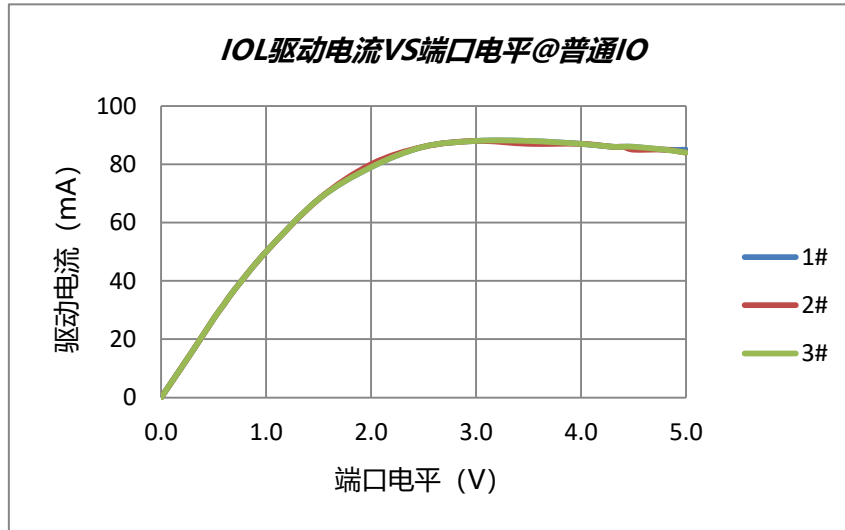
#### 输入 SMT 阈值电压 VS 电源电压



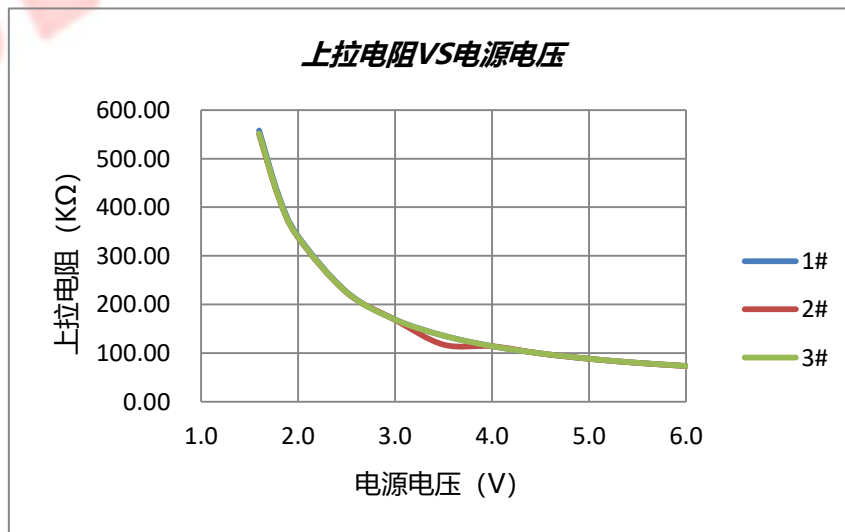


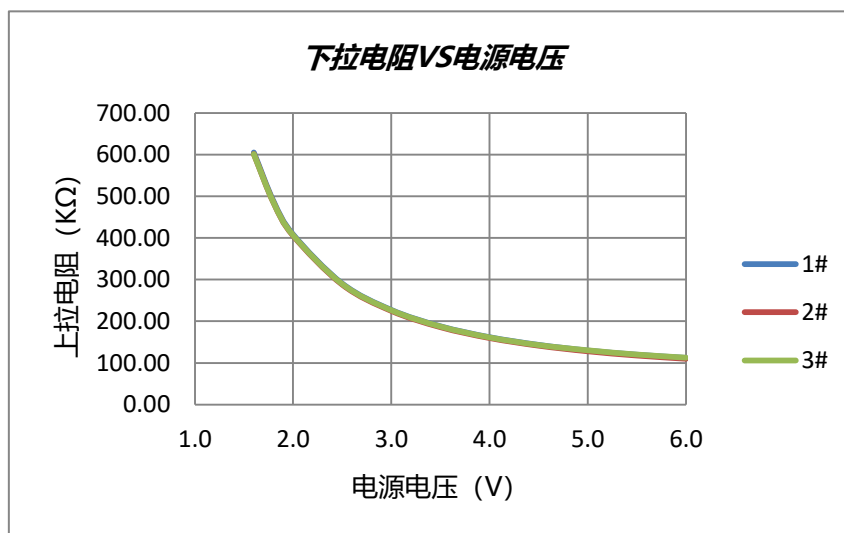
**I/O 输出 驱动电流 VS 端口电压 (VDD=5V)**





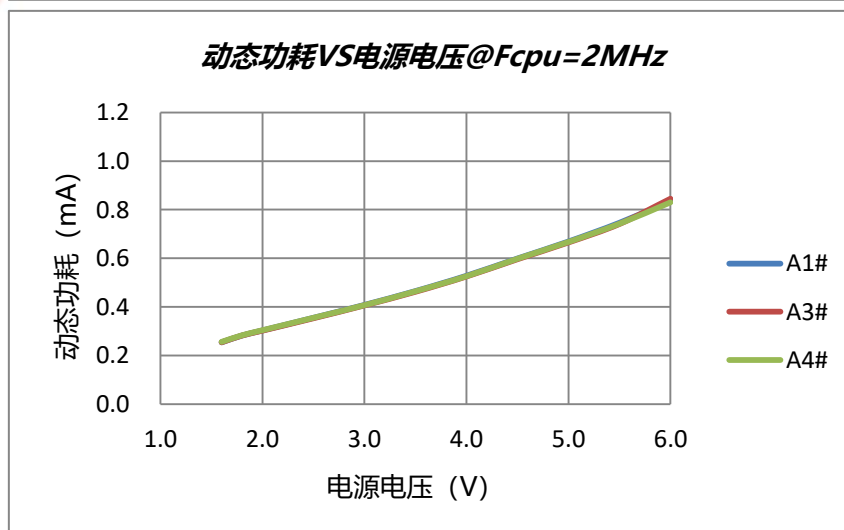
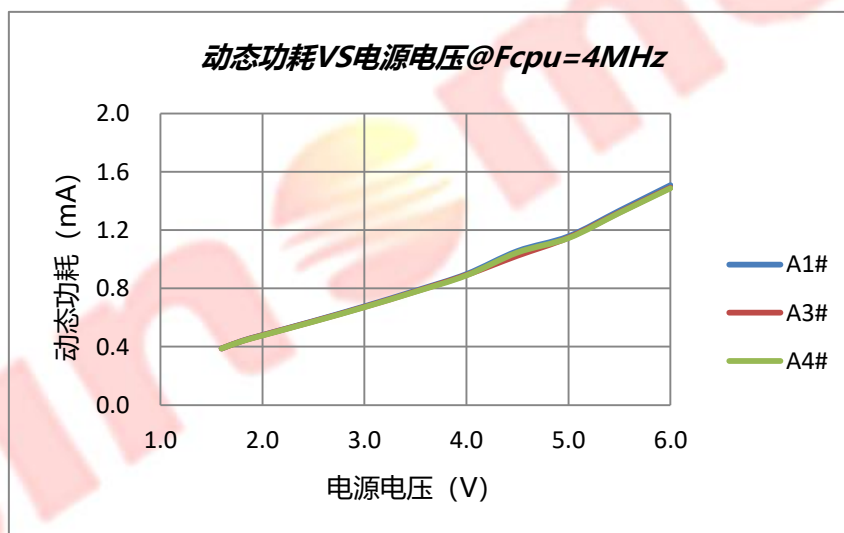
### 上/下拉电阻值 VS 电源电压

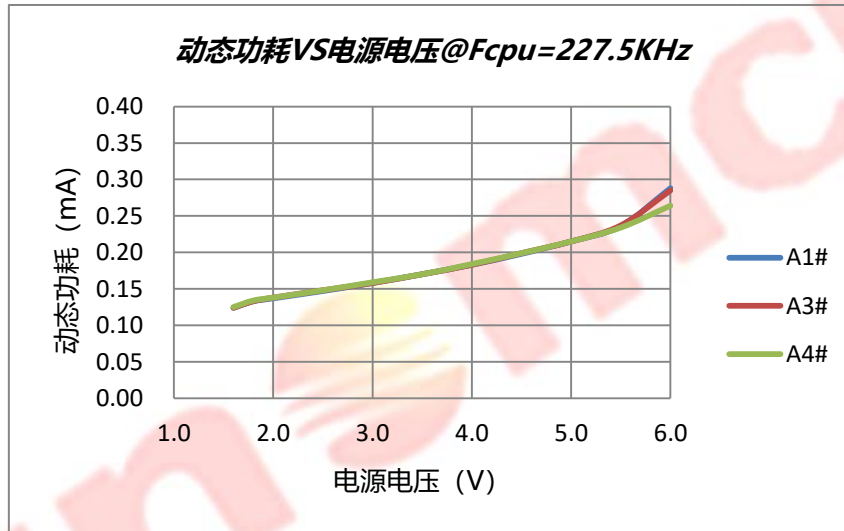
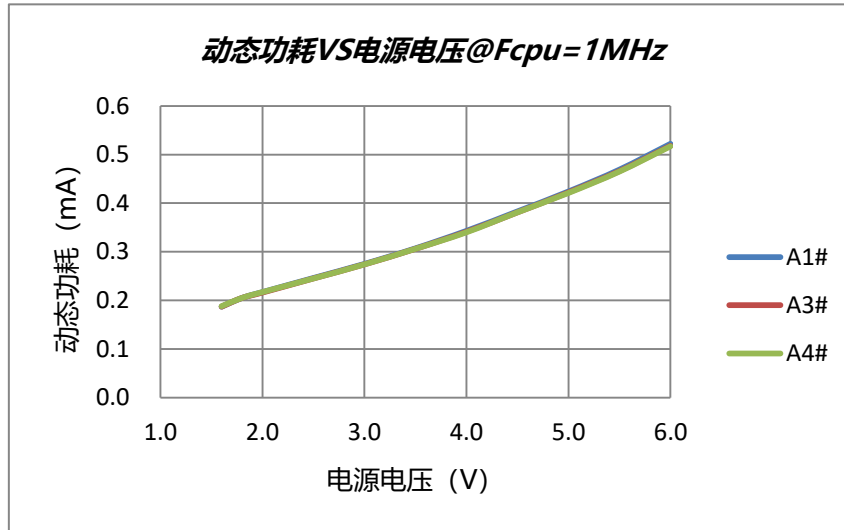




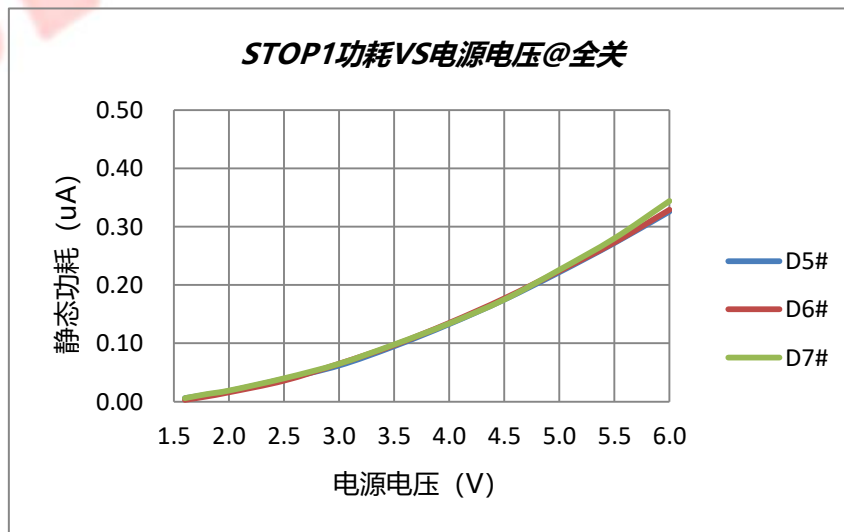
## 9.2 功耗特性

### 运行模式 功耗 VS 电源电压



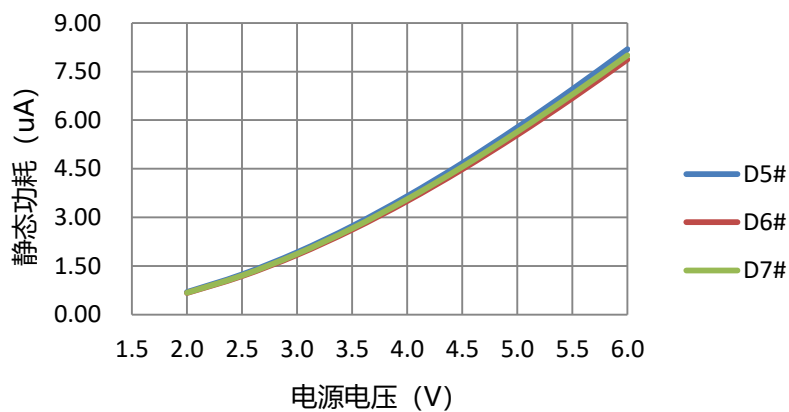


#### 休眠模式 功耗 VS 电源电压/温度

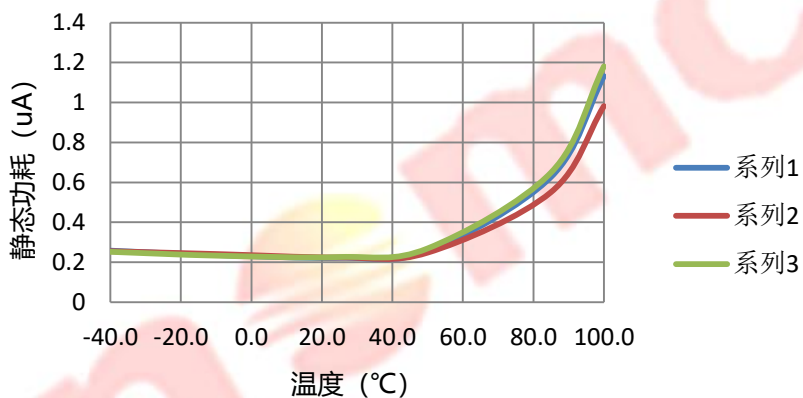




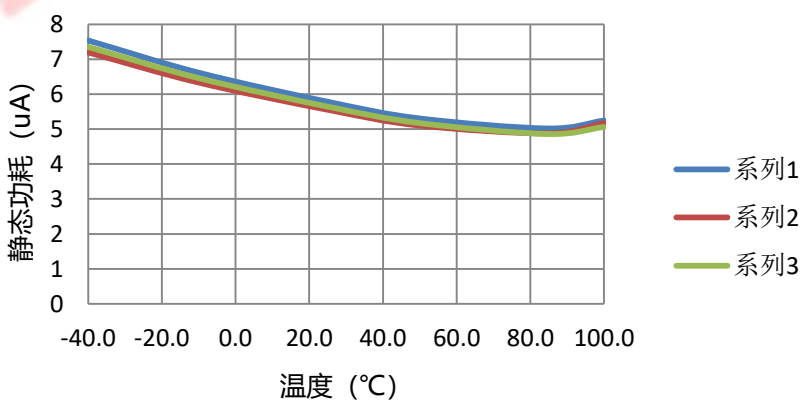
**STOP2功耗VS电源电压@开LVR**



**STOP1功耗VS温度@全关@VDD=5V**

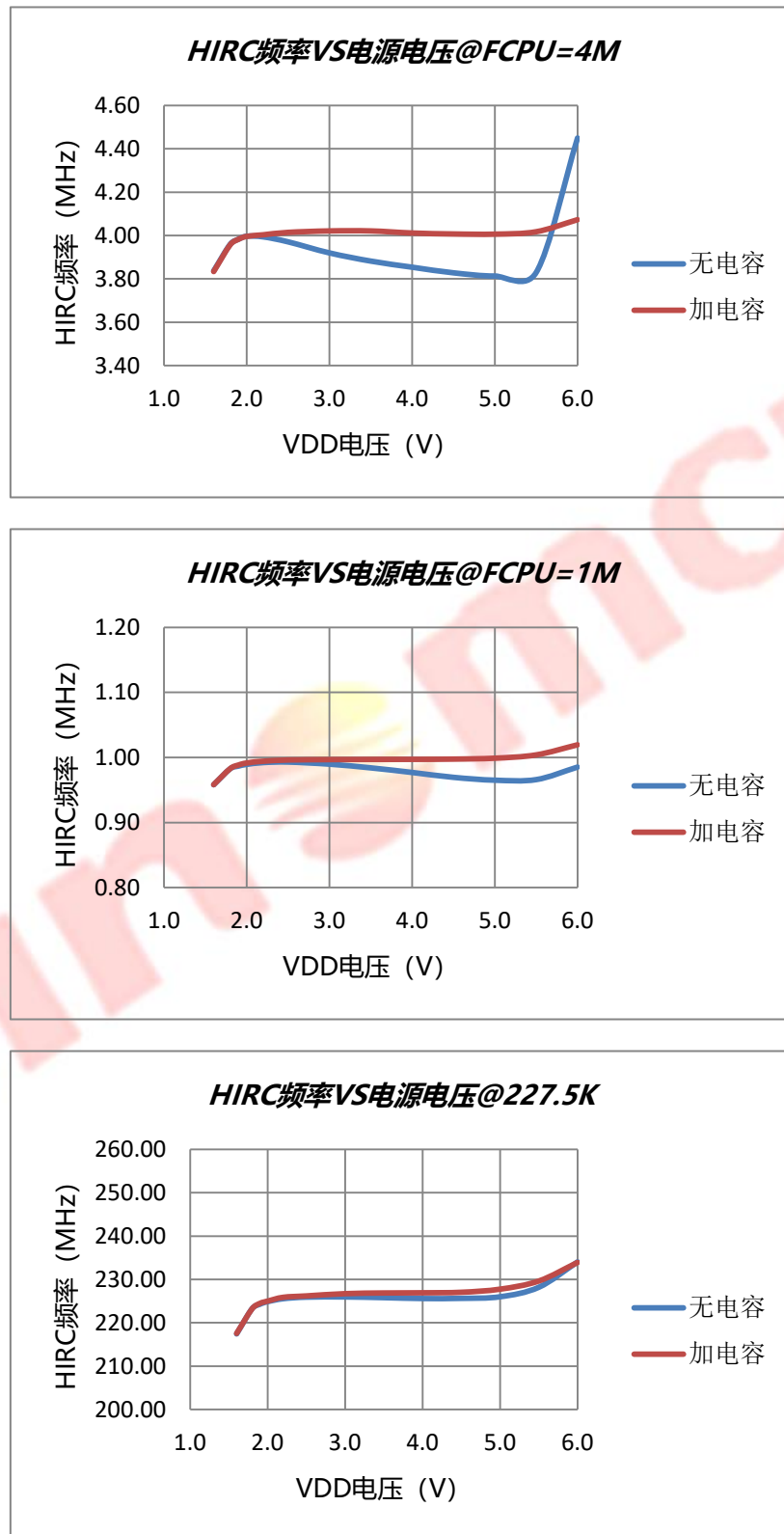


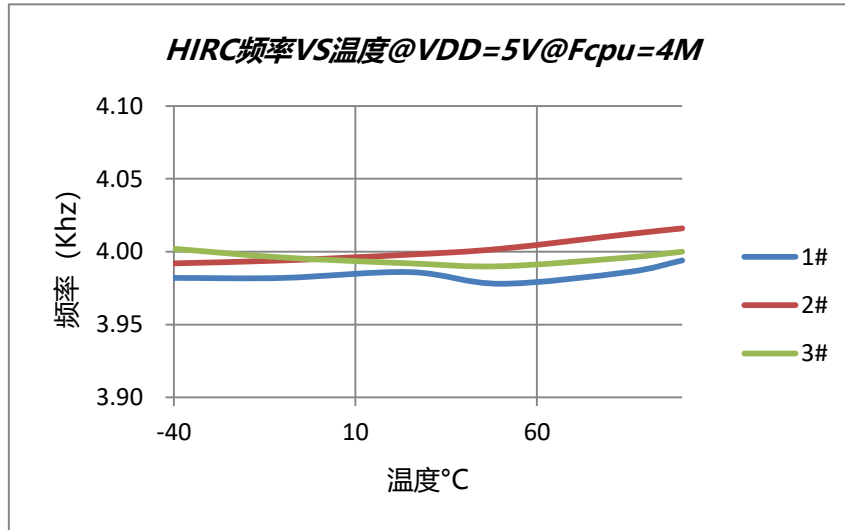
**STOP1功耗VS温度@LVR开@VDD=5V**



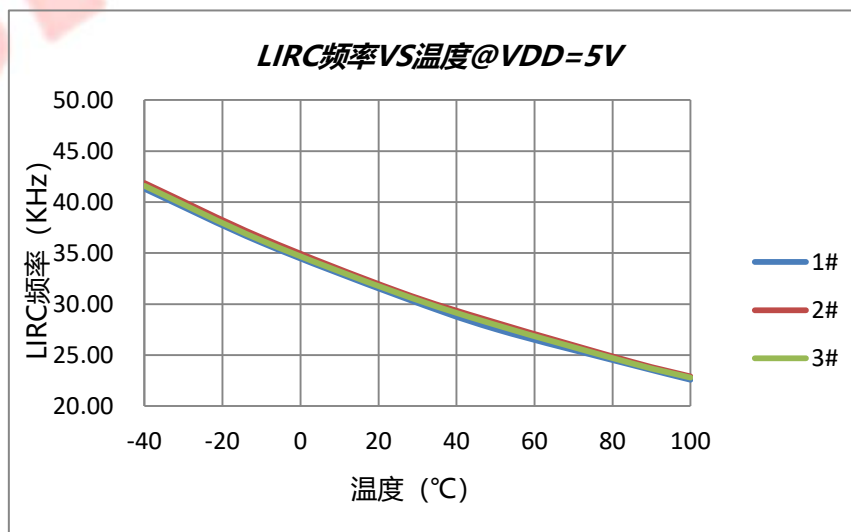
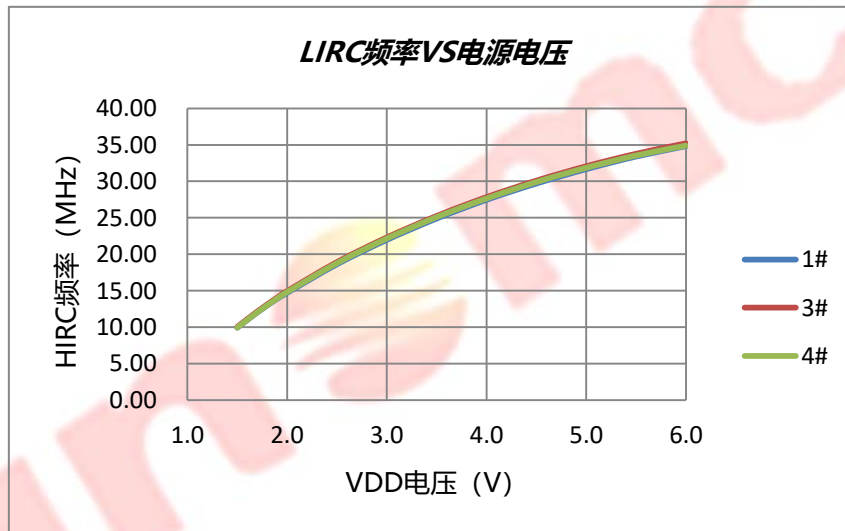
### 9.3 模拟电路特性

#### HIRC 频率 VS 电源电压/温度

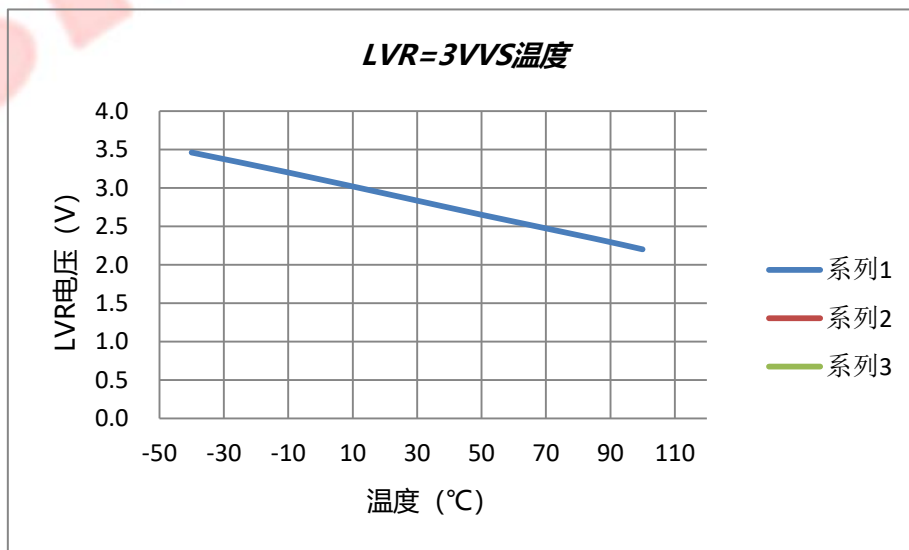
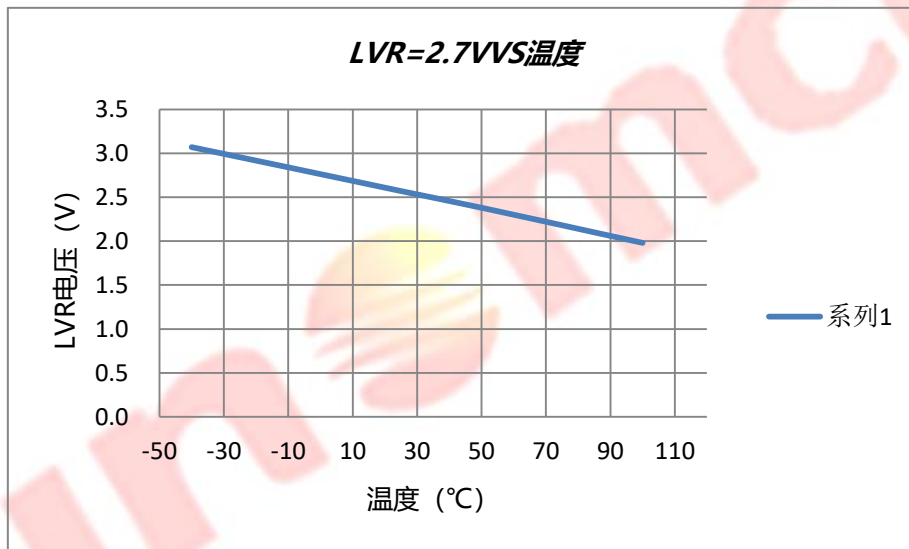
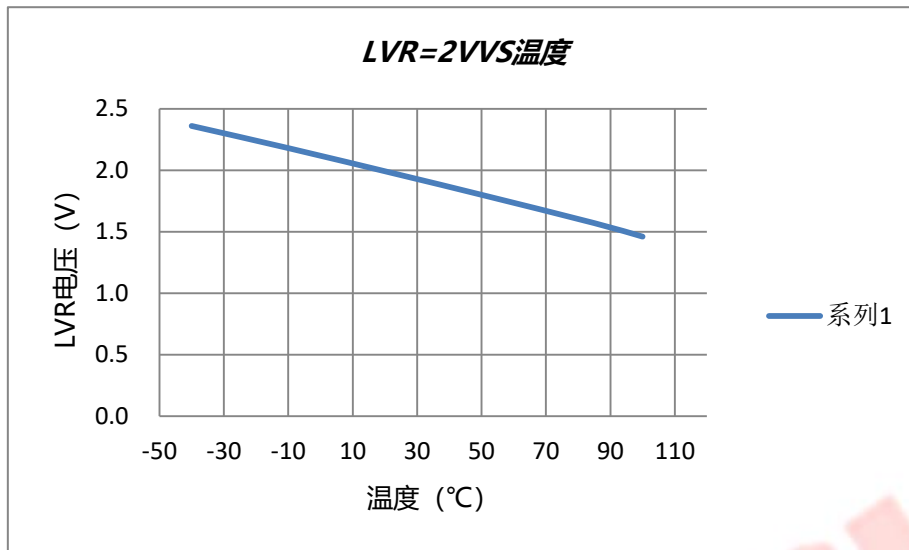


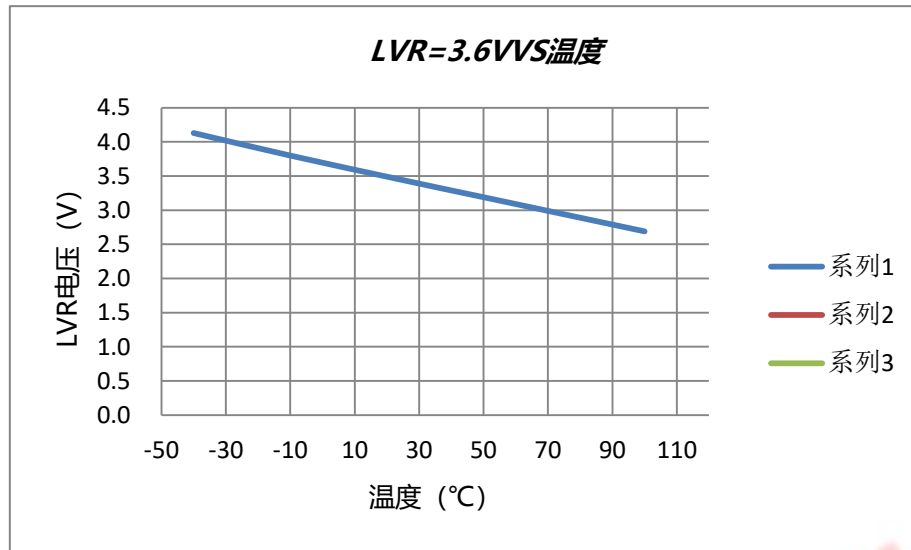


### LIRC 频率 VS 电源电压/温度



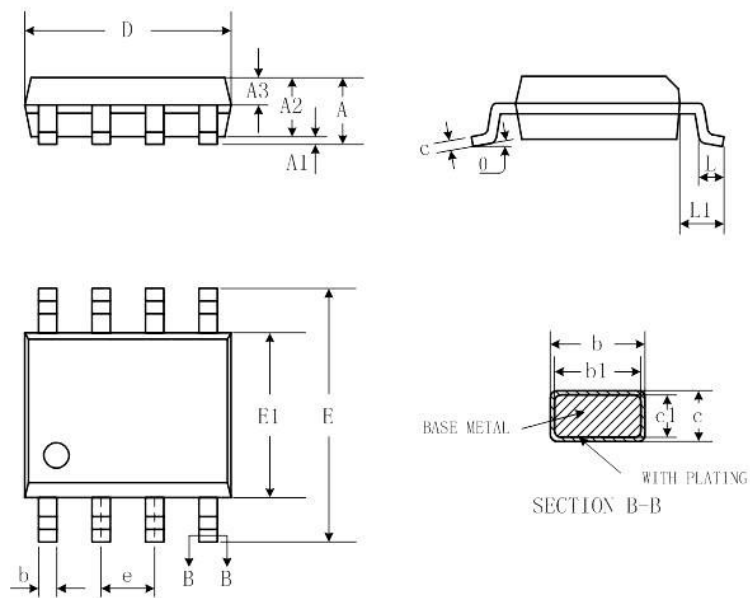
LVR 电压 VS 温度





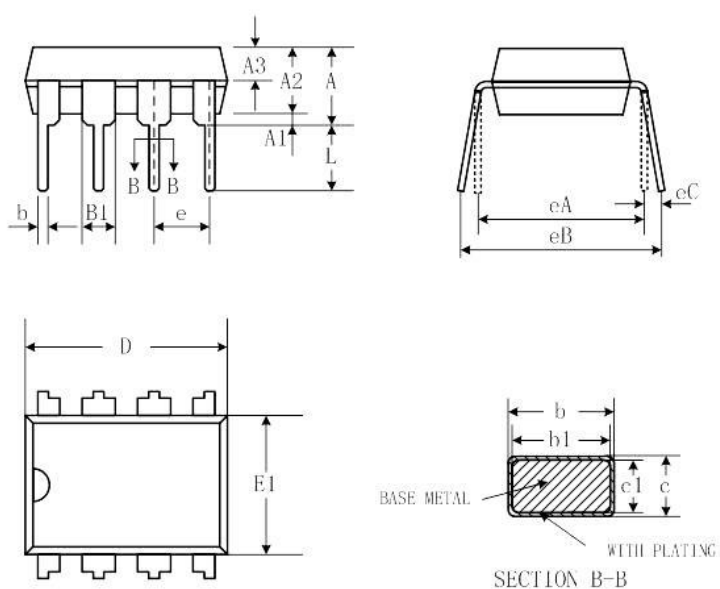
## 10 封装尺寸

### 10.1 SOP8



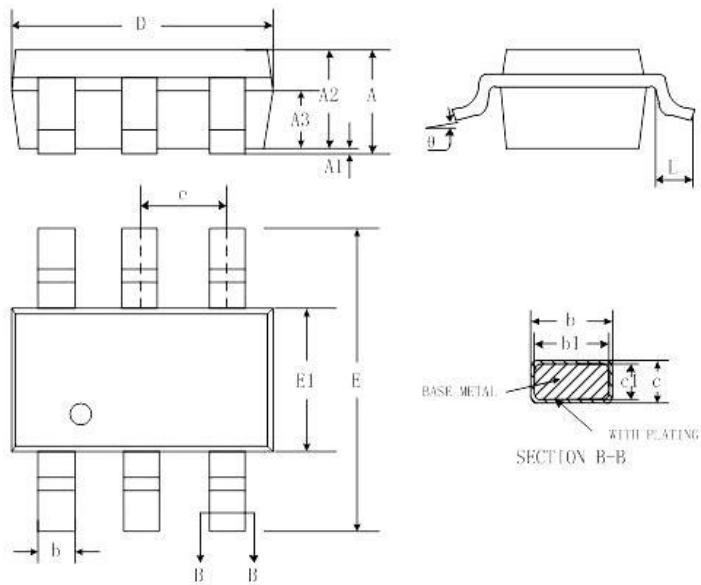
| SYMBOL | MILLIMETER |      |      |
|--------|------------|------|------|
|        | MIN        | TYP  | MAX  |
| A      | -          | -    | 1.77 |
| A1     | 0.08       | 0.18 | 0.28 |
| A2     | 1.20       | 1.40 | 1.60 |
| A3     | 0.55       | 0.65 | 0.75 |
| b      | 0.39       | -    | 0.48 |
| 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°   |

### 10.2 DIP8



| SYMBOL | MILLIMETER |      |      |
|--------|------------|------|------|
|        | MIN        | TYP  | MAX  |
| A      | 3.60       | 3.80 | 4.00 |
| A1     | 0.51       | -    | -    |
| A2     | 3.10       | 3.30 | 3.50 |
| A3     | 1.50       | 1.60 | 1.70 |
| b      | 0.44       | -    | 0.53 |
| b1     | 0.43       | 0.46 | 0.48 |
| B1     | 1.52BSC    |      |      |
| c      | 0.25       | -    | 0.31 |
| c1     | 0.24       | 0.25 | 0.26 |
| D      | 9.05       | 9.25 | 9.45 |
| E1     | 6.15       | 6.35 | 6.55 |
| e      | 2.54BSC    |      |      |
| eA     | 7.62BSC    |      |      |
| eB     | 7.62       | -    | 9.50 |
| eC     | 0          | -    | 0.94 |
| L      | 3.00       | -    | -    |

### 10.3 SOT23-6



| SYMBOL   | MILLIMETER |      |      |
|----------|------------|------|------|
|          | MIN        | TYP  | MAX  |
| A        | -          | -    | 1.35 |
| A1       | 0.04       | -    | 0.15 |
| A2       | 1.00       | 1.10 | 1.20 |
| A3       | 0.55       | 0.65 | 0.75 |
| b        | 0.30       | -    | 0.50 |
| b1       | 0.30       | 0.40 | 0.45 |
| c        | 0.08       | -    | 0.22 |
| c1       | 0.08       | 0.13 | 0.20 |
| D        | 2.72       | 2.92 | 3.12 |
| E        | 2.60       | 2.80 | 3.00 |
| E1       | 1.40       | 1.60 | 1.80 |
| e        | 0.95BSC    |      |      |
| L        | 0.30       | -    | 0.60 |
| $\theta$ | 0          | -    | 8°   |

## 11 修订记录

| 版本   | 修订日期       | 修订内容                                                                  |
|------|------------|-----------------------------------------------------------------------|
| V1.0 | 2018-03-26 | 初版发布；                                                                 |
| V1.1 | 2018-12-17 | 更新 HIRC 温度曲线，新增 LVR 温度曲线；                                             |
| V1.2 | 2019-01-11 | 更新工作温度参数，增加 LVR 低温偏高的注释；                                              |
| V1.3 | 2019-04-26 | 新增 SOT23-6 封装 A0T；                                                    |
| V1.4 | 2020-07-13 | 修改或新增关于直流参数测试条件、开漏 I/O 输出高产生漏电流、低功耗模式未用引脚设置、复位时 VDD 上电慢的应对方法等相关的应用注释； |