

# M8P5170

**8BIT  
MTP MCU**

**Version 1.05**

2026年06月



**磐芯电子**

本公司保留对产品可靠性、功能和设计方面的改进作进一步说明的权利  
数据手册的更改,恕不另行通知

<http://www.masses-chip.com/>

本公司不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，本公司的产品不是专门设计来应用于外科植入、生命维持和任何本公司产品的故障会对个体造成伤害甚至死亡的领域。如果将本公司的产品应用于上述领域，即使这些是由本公司在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证本公司及其雇员、子公司、分支机构和销售商与上述事宜无关。

## 目录

应用注意事项.....	5
<b>1 产品简述 .....</b>	<b>6</b>
1.1 特性.....	6
1.2 引脚图 .....	7
1.2.1 SOT23-6 .....	7
1.2.2 SOP8 .....	7
1.3 引脚描述.....	8
<b>2 中央处理器 (CPU) .....</b>	<b>9</b>
2.1 程序存储器.....	9
2.1.1 复位向量.....	9
2.1.2 中断向量 (0008H) .....	10
2.1.3 查表.....	10
2.2 数据存储器.....	12
2.2.1 数据存储器结构 .....	12
2.2.2 数据存储器寻址模式.....	12
2.2.3 系统寄存器定义 .....	13
2.2.4 INDF0 间接寻址寄存器 0 .....	13
2.2.5 INDF1 间接寻址寄存器 1 .....	13
2.2.6 FSR0 间接寻址指针 0 .....	13
2.2.7 FSR1 间接寻址指针 1 .....	14
2.2.8 HBUF 查表数据高 8 位 .....	14
2.2.9 PCL 程序计数器指针低位 .....	14
2.2.10 STATUS 状态寄存器.....	14
<b>3 复位 .....</b>	<b>15</b>
3.1 复位方式.....	15
<b>4 系统时钟 .....</b>	<b>16</b>
4.1 概述 .....	16
4.2 OSCM 寄存器 .....	16
4.3 系统时钟工作模式 .....	17
4.4 IRCCAL 寄存器.....	18
4.5 IRCLOCK 寄存器 .....	18
4.6 系统时钟结构框图 .....	19
<b>5 中断 .....</b>	<b>20</b>
5.1 概述 .....	20
5.2 OPTION 配置寄存器 .....	20
5.3 IOAICR 变化中断使能寄存器.....	20
5.4 INTCR0 中断控制寄存器 0 .....	21
5.5 INTF0 中断标志寄存器 0 .....	22
<b>6 端口 .....</b>	<b>23</b>
6.1 IOA 数据寄存器 .....	23
6.2 OEA 方向寄存器 .....	23
6.3 PUA 上拉使能寄存器 .....	24
6.4 IOAICR 变化中断使能寄存器.....	24

<b>7 定时器 0(TCO) .....</b>	<b>25</b>
7.1 概述 .....	25
7.2 TOCR0 控制寄存器 .....	27
7.3 TOCR1 控制寄存器 .....	28
7.4 TCOCL TCO 计数器低 8 位/周期寄存器 .....	28
7.5 TCOCH TCO 计数器高位 .....	29
7.6 PWMOCR 控制寄存器 .....	29
7.7 PWM0xD 数据位(x=0,1,2,3) .....	29
7.8 PWMx 波形图(x=0,1) .....	30
7.9 串行 LED 模式功能 .....	30
<b>8 定时器 1(TC1) .....</b>	<b>31</b>
8.1 概述 .....	31
8.2 T1CR 控制寄存器 .....	31
<b>9 触摸按键 (CDC) .....</b>	<b>32</b>
9.1 概述 .....	32
<b>10 低电压检测(LVD).....</b>	<b>33</b>
10.1 概述 .....	33
10.2 LVDCR 低电压检测控制寄存器 .....	33
<b>11 看门狗 (WDT) .....</b>	<b>34</b>
11.1 概述 .....	34
11.2 OPTION 配置寄存器 .....	34
<b>12 仿真烧录.....</b>	<b>35</b>
12.1 仿真连线示意图 .....	35
12.2 校准模式连线示意图 .....	35
12.3 默认 (普通) 连线示意图 .....	36
12.4 带电烧录连线示意图 .....	37
12.5 带电烧录软件配置说明 .....	38
<b>13 芯片配置字 (OPTION BIT) .....</b>	<b>39</b>
<b>14 电性参数.....</b>	<b>40</b>
14.1 极限参数 .....	40
14.2 直流特性 .....	40
14.3 IO 口拉灌电流特性 .....	42
14.4 系统时钟特性 .....	43
<b>15 封装信息.....</b>	<b>45</b>
15.1 SOT23-6 .....	45
15.2 SOP8 .....	46
<b>16 指令集简述.....</b>	<b>47</b>
16.1 概述 .....	47
16.2 符号说明 .....	47
16.3 M8Pxxx 指令集表 .....	48
16.4 M8Pxxx 指令说明 .....	50
<b>17 修改记录.....</b>	<b>51</b>

## 应用注意事项

### 1、IO 引脚下拉电阻应用

- 1、IOA5 上电默认下拉，驱动 NMOS 管可以节省外接电阻。
- 2、下拉电阻在仿真模式下不使能。

### 2、烧录口相关注意事项

- 1、仿真模式下 LDSA3 位需保持为 0。

# 1 产品简述

M8P5170 采用高速低功耗 CMOS 工艺设计开发的 8 位高性能精简指令单片机，内部有 1K\*16 多次可编程存储器 (MTP)，64\*8 位的数据存储器 (RAM)，6 个双向 I/O 口，1 个 64us 定时器，1 个 8/16 位定时器/计数器，4 路 8 位分辨率 PWM，5/6 路触摸按键，高精度 LVD 功能，支持多种系统工作模式和多个中断源。

## 1.1 特性

### ■ CPU 特性

- 高性能精简指令
- 1K\*16位的MTP程序存储器
- 64\*8位的数据存储器
- 5级堆栈缓存器
- 支持查表指令

### ■ I/O 口

- 6个双向I/O口
- 可编程弱上拉口IOA
- 可编程弱下拉口IOA3, IOA5(IOA5默认复位下拉)
- IOA口变化中断
- IOA2, IOA3可设小驱动 (直推LED)

### ■ 64us 定时器

### ■ 1 个定时器/计数器 (16Bit)

- TC0: 具有周期寄存器功能的定时/计数器
  - 16位模式下, 周期寄存器用作低8位计数器
  - 8位模式下, 可用作PWM0/PWM1的时基
  - 8位PWM00-PWM03输出  
PWM01、PWM03极性单独可设  
PWM00可映射在IOA0/IOA4  
PWM01可映射在IOA1/IOA5  
可复用做WS2812驱动器

### ■ 振荡器

- 内部高速RC振荡器 (16MHz, 用于系统时钟, 定时器)
- 内部低速RC振荡器 (64KHz, 用于看门狗, 定时器)

### ■ 高精度 LVD 模块

- 检测电压 1: VDD/2
- 检测电压 2: 外部端口输入
- 内部参考 1.15V-2.7V, 50mV/档

### ■ 系统工作模式

- 普通模式
- 绿色模式
- 休眠模式

### ■ 多路中断源

### ■ 触摸按键

- 双模式
  - 外接电容模式有更高的抗干扰性能
  - 无电容模式, 可以最多支持 6 通道

### ■ 看门狗定时器

### ■ 特殊功能

- 可编程代码保护
- 带电烧录

### ■ 封装形式

- SOT23-6
- SOP8

## 1.2 引脚图

注：烧录接口说明

1、仿真器烧录：

仿真器烧录时，烧录口分别是VDD、SWP（原PSDA）、VPP、GND。

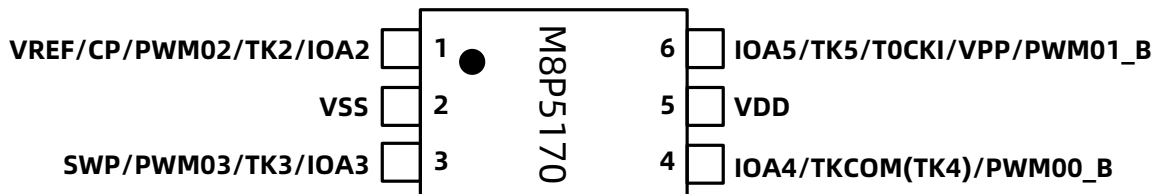
2、烧录器烧录：

烧录器烧录时，根据烧录参数设置不同，烧录接口有两种方式

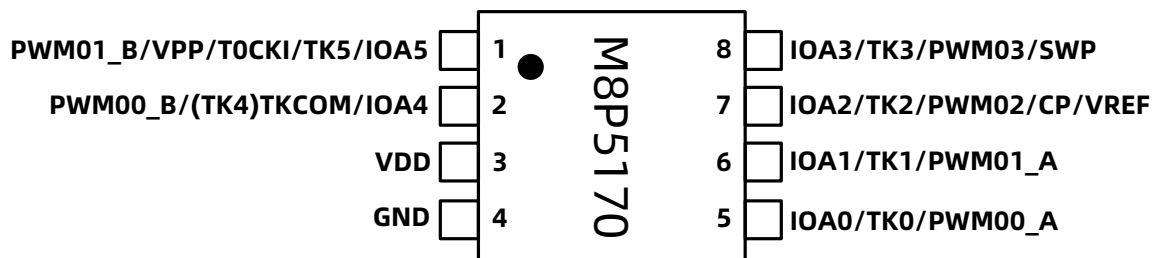
(1)校准模式：烧录口分别是VDD、SWP（原SDA）、VPP、VREF(CLK)、GND。

(2)默认普通：烧录口分别是VDD、SWP（原SDA）、VPP、GND。

### 1.2.1 SOT23-6



### 1.2.2 SOP8



注：TK4在无电容模式下使用。

### 1.3 引脚描述

名称	类型	说明
VDD, GND	P	电源输入端
IOA[0]	I/O	输入/输出 IO, SMT, 上拉电阻
TK0	A	触摸按键通道 0
PWM00_A	O	PWM00_A 输出端口
IOA[1]	I/O	输入/输出 IO、SMT、上拉电阻
TK1	A	触摸按键通道 1
PWM01_A	O	PWM01_A 输出端口
IOA[2]	I/O	输入/输出 IO、SMT、上拉电阻
TK2	A	触摸按键通道 2
PWM02	O	PWM02 输出端口
CP	A	LVD 外部电压输入口
VREF	A	AD 外部参考电压输入
IOA[3]	I/O	输入/输出 IO、SMT、上拉电阻
TK3	A	触摸按键通道 3
PWM03	O	PWM03 输出端口
SWP	I/O	编程用
IOA[4]	I/O	输入/输出 IO、SMT、上拉电阻
TK4	A	触摸按键通道 4
TKCOM	A	触摸按键灵敏度电容口
PWM00_B	O	PWM00_B 输出端口
IOA[5]	I/O	输入/输出 IO、SMT、上拉电阻
TK5	I	触摸按键通道 5
TOCKI	I	TCO 外部时钟输入
VPP	P	编程高压电源
PWM01_B	O	PWM01_A 输出端口

注：I = 输入 O = 输出 I/O = 输入/输出 P = 电源 A = 模拟端口

# 2 中央处理器 (CPU)

## 2.1 程序存储器

- 1K 的程序存储器 (MTP) 空间
- 复位向量为 0000H
- 硬件中断向量为 0008H
- 支持数据表

地址	说明
0x0000	复位向量
0x0001 ~ 0x0007	用户区
0x0008	中断向量
0x0009 ~ 0x03FF	用户区

### 2.1.1 复位向量

M8P5170有以下三种复位方式：

- 上电复位
- 看门狗复位
- 欠压复位

发生上述任一种复位后，程序将从0000H处重新开始执行，系统寄存器也将都恢复为初始默认值。

#### 例：定义复位向量

```

ORG      0000H
GOTO     Main_Program      ;//跳转至用户程序开始
...
Main_Program:              ;//用户程序开始
...
Main:
...
GOTO     Main              ;//用户主程序循环
    
```

## 2.1.2 中断向量 (0008H)

M8P5170中断向量地址为0008H。一旦有中断响应，程序计数器PC的当前值就会存入堆栈缓存器并跳转到0008H处开始执行中断服务程序。

### 例：中断服务程序

```

    ORG      0000H
    GOTO    Main_Program      ;//跳转到程序开始
    ORG      0008H
    GOTO    Interrupt         ;//发生中断后,跳转到中断子程序
Main_Program:
    ...
Main:
    ...
    GOTO    Main              ;//主程序循环

Interrupt:
    PUSH                               ;//压栈、保存A、STATUS
    ...
    POP                                ;//出栈、恢复 A、STATUS
    RETIE

    END

```

## 2.1.3 查表

使用RDT指令可以读取程序区数据，其中读到的16位数据高位放在HBUF中，低位放在A寄存器中。FSR1和FSR0组成12位程序区数据寻址指针。

### 例 1: 查找 ROM 地址为“DTAB”的值

```

    MOVIA   0                    ;//要查的数据在表中的位置
    ADDIA   LOW(DTAB)            ;//获取数据表地址低位
    MOVAR   FSR0                 ;//设置数据表低位指针
    MOVIA   0                    ;//要查的数据在表中的位置
    ADCIA   HIGH(DTAB)          ;//获取数据表地址高位
    MOVAR   FSR1                 ;//设置数据表高位指针
    ;//若需读取表的其它数据,修改指针

    RDT                               ;//读取表的第一个数据0x0102
    MOVAR   TABDL                ;//将低位数据0x02放在TABDL
    MOVR    HBUF,A              ;//高位数据读入累加器A
    MOVAR   TABDH                ;//将高位数据0x01放在TABDH
    ...
DTAB:
    DW      0x0102
    DW      0x1112

```

使用加 PCL 地址来跳转，通过 GOTO 指令可以跳转不同的程序标号。

**例 2:+PCL GOTO 表**

MOVR	ADDRESS,A	;//获取表格地址
ADDAR	PCL,R	
GOTO	TAB1	;//PCL +0 处理程序
GOTO	TAB2	;//PCL +1 处理程序
GOTO	TAB3	;//PCL +2 处理程序
TAB1:		
	处理程序	
TAB2:		
	处理程序	
TAB3:		
	处理程序	

使用加PCL地址来跳转，通过RETIA指令可以读取数据表。

**例 3:+PCL RETIA 表**

MOVR	ADDRESS,A	;//获取地址
ADDAR	PCL,R	;//地址指针加 1
RETIA	0	;//PCL +0
RETIA	1	;//PCL +1
RETIA	2	;//PCL +2
...		

## 2.2 数据存储器

### 2.2.1 数据存储器结构

寄存器可通过直接寻址或INDF0、INDF2间接寻址；特殊功能寄存器可通过直接寻址或INDF1、INDF2间接寻址。具体地址分配参照下表。

地址	间接寻址 INDF0	间接寻址 INDF1	间接寻址 INDF2	直接寻址
0x00 ~ 0xFF	YES	YES	YES	YES

注：此芯片间接寻址INDF1直接映射到00-FFH。

### 2.2.2 数据存储器寻址模式

☆ 直接寻址模式

地址 来自指令低9位

如：MOVAR 0x001； A 中的值传送给地址为 0x001 的 RAM 中

☆ 间接寻址模式 0

地址 0                      FSR0

如：MOVAR INDF0； A 中的值传送给 FSR0 指向的 RAM 中

☆ 间接寻址模式 1

地址 1                      FSR1

如：MOVAR INDF1； A 中的值传送给 FSR1 指向的 RAM 中

☆ 间接寻址模式 2

地址 FSR1                      FSR0

如：MOVAR INDF2； A 中的值传送给(FSR1:FSR0)指向的 RAM 中

### 2.2.3 系统寄存器定义

数据寄存器映射表								
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0x000 ~ 0x03F	GPR							
0x040 ~ 0x0AF	RESERVE							
0x0B0	INDF0	FSR0	TKCTR0	TKCTR1	TKCNTL	TKCNTH	TKCHS	-
0x0B8	INDF1	FSR1	PCL	STATUS	OPTION	OSCM	WDTC	-
0x0C0	INDF2	HBUF	-	-	INTCR0	INTF0		
0x0C8	IOA	OEA	PUA	IOAICR		-	-	-
0x0D0	-	-	-	-	-	-	-	-
0x0D8	PWM0CR	PWM00D	PWM01D	PWM02D	PWM03D	-	-	-
0x0E0	-	-	-	-	-	-	-	-
0x0E8	TOCR0	TOCR1	TCOCL	TCOCH	T1CR	-	-	-
0x0F0	LVDCR	-	-	-	-	-	-	-
0x0F8	-	-	-	-	-	-	-	-

注：GPR 为通用寄存器。

### 2.2.4 INDF0 间接寻址寄存器 0

访问INDF0寄存器时，实现间接寻址模式0，访问到的是FSR0寄存器所指向的寄存器内容，间接寻址模式0仅可寻址通用寄存器区0x0000~0x00FF空间。

### 2.2.5 INDF1 间接寻址寄存器 1

访问INDF1寄存器时，实现间接寻址模式1，访问到的是FSR1寄存器所指向的寄存器内容，间接寻址模式1仅可寻址通用寄存器区0x0100~0x01FF（映射到0x0000~0x00FF）空间。

### 2.2.6 FSR0 间接寻址指针 0

使用间接寻址模式0访问通用寄存器时，FSR0为地址指针；当以间接寻址模式2访问通用寄存器时，FSR0作为地址指针的低位。

## 2.2.7 FSR1 间接寻址指针 1

使用间接寻址模式1访问通用寄存器时，FSR1为地址指针；当以间接寻址模式2访问通用寄存器时，FSR1作为地址指针的高位。

## 2.2.8 HBUF 查表数据高 8 位

使用RDT指令读取程序区数据时，读到的16位数据高8位放在HBUF中。

## 2.2.9 PCL 程序计数器指针低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCL	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:0] **PCL[7:0]**: 程序计数器指针低位

用户将该PCL作为目的操作数做加法运算时（ADDAR PCL、ADCAR PCL），13位PC值参与运算，运算结果写入PC，实现程序的相对跳转；加法运算外的其它运算时，仅PCL参与运算，PCH保持不变。PCH不可寻址。

## 2.2.10 STATUS 状态寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
STATUS	-	-	-	-	-	Z	DC	C
读/写	-	-	-	-	-	R/W	R/W	R/W
复位后	-	-	-	-	-	X	X	X

Bit 2 **Z**: 零标志

0 = 算术/逻辑运算的结果非零

1 = 算术/逻辑运算的结果为零

Bit 1 **DC**: 辅助进位标志

0 = 加法运算时低四位没有进位，或减法运算后有向高四位借位

1 = 加法运算时低四位有进位，或减法运算后没有向高四位借位

Bit 0 **C**: 进位标志

0 = 加法运算后没有进位、减法运算有借位发生或移位后移出逻辑“0”

1 = 加法运算后有进位、减法运算没有借位发生或移位后移出逻辑“1”

# 3 复位

## 3.1 复位方式

- 上电复位 (POR)
- 欠压复位 (BOR)
- 看门狗定时器复位 (WDT Reset)

M8P5170有3种复位方式，任何一种复位都会使PC程序计数器清零，让程序从0000H处开始运行，并且使系统寄存器值复位。

# 4 系统时钟

## 4.1 概述

M8P5170 由内置的 16MHz RC 振荡电路 (HIRC 16MHz) 作为系统时钟源 Fosc, 内置低速时钟仅可作为定时器时钟源与 WDT 时钟源。

## 4.2 OSCM 寄存器

工作模式控制寄存器 OSCM

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OSCM	-	-	-	STOP	-	STPH	-	STPL
读/写	-	-	-	R/W	-	R/W	-	R/W
复位后	-	-	-	0	-	1	-	1

- Bit 4      **STOP:** CPU工作状态标志位  
           0 = CPU正常工作, 所有复位唤醒  
           1 = CPU停止工作
- Bit 2      **STPH:** 高频振荡器控制  
           0 = 休眠状态或低速模式下高速振荡器仍然工作  
           1 = 休眠状态或低速模式下关闭高频振荡器
- Bit 0      **STPL:** 低频振荡器控制  
           0 = 休眠状态下低频振荡器仍然工作  
           1 = 休眠状态下低频振荡器停止工作

## 4.3 系统时钟工作模式

**普通模式：**普通模式只有一种是：1.高频时钟工作，低频时钟工作，不进 STOP  
(电流特性参考电性参数表 I<sub>DD1</sub>)

**绿色模式：**绿色模式有两种分别是：1.高频时钟工作，低频时钟工作，进 STOP  
(电流特性参考电性参数表 I<sub>SP1</sub>)  
2.高频时钟停止，低频时钟工作，进 STOP  
(电流特性参考电性参数表 I<sub>SP2</sub>)

绿色模式可以由所有中断或 WDT 唤醒。

**休眠模式：**休眠模式只有一种是：高频时钟停止，低频时钟停止，进 STOP  
(电流特性参考电性参数表 I<sub>SP3</sub>)

休眠模式可以由外部中断、IO 变化中断或 WDT 唤醒。

**注：**(1) 省电建议，程序运行时跑高频，快速跑完程序然后进休眠，此时休眠下需设置高频时钟停止工作。  
(2) 各工作模式的工作电流参考电性参数表。  
(3) 绿色和休眠模式下，如果总中断不开启，所有中断唤醒能唤醒芯片但是不会进中断。

## 4.4 IRCCAL 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCCAL	-	IRCCAL[6:0]						
读/写	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	X	X	X	X	X	X	X

内置的高频 RC 振荡电路在芯片上电后频率为校准过的 16MHz，但程序中可以通过特殊的流程来调整此频率以满足特定应用需求。

## 4.5 IRCLOCK 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IRCLOCK	-	-	-	-	-	-	-	-
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	X	X	X	X	X	X	X	X

### 例：调整 IRC 频率

TASK\_IRCCAL:

```

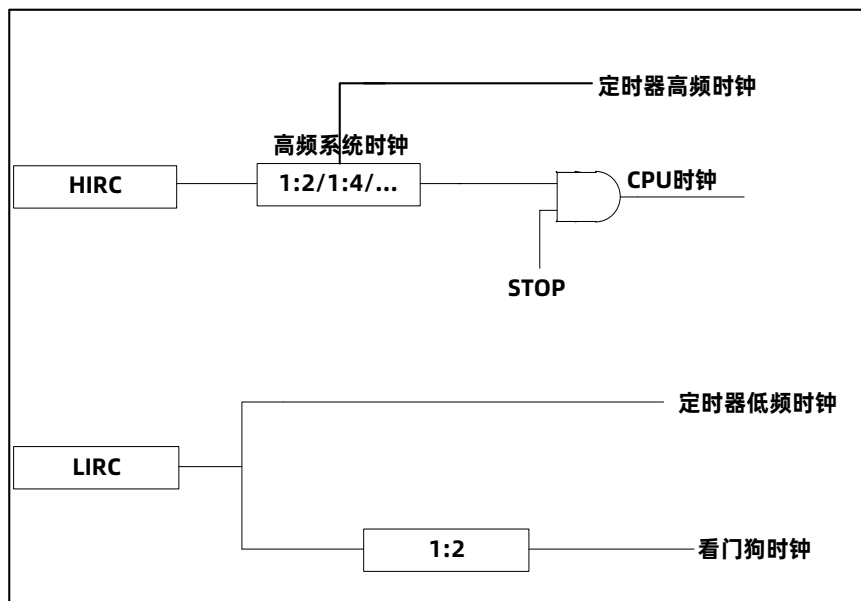
MOVIA    0x55
MOVAR    IRCLOCK    ;// IRCLOCK写入55H
MOVIA    0xAA
MOVAR    IRCLOCK    ;// IRCLOCK写入AAH
MOVIA    VALUE
MOVAR    IRCCAL     ;//写入IRCCAL

```

...

;//若需继续在IRCCAL寄存器内写入其他值需要重复以上所有步骤

## 4.6 系统时钟结构框图



	高速运行模式	休眠模式 (STOP=1)
高频振荡器	运行	由 STPH 决定
低频振荡器	运行	由 STPL 决定
WDT	配置字决定	由配置字决定
TC0/TC1	TCxEN	高速时钟源&STPH=0 低速时钟源&STPL=0

# 5 中断

## 5.1 概述

M8P5170有多路中断源: TC0定时器, TC1定时器, IO变化中断, 低电压检测与触摸转换完成中断。中断可以将系统从休眠模式或绿色模式中唤醒, 在唤醒前, 中断请求被锁定。一旦程序进入中断, 寄存器OPTION的位GIE被硬件自动清零以避免响应其它中断。系统退出中断后, RETIE指令将自动将GIE置“1”, 以响应下一个中断。

设置 GIE 和中断控制寄存器 INTCR0/INTCR1 来使能中断, 查询 INTF0/INTF1 中断标志寄存器判断中断是否发生。

## 5.2 OPTION 配置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIE	-	TO	PD	-	-	-	-
读/写	R/W	-	R	R	-	-	-	-
复位后	0	-	1	1	-	-	-	-

Bit 7      **GIE:** 全局中断控制位  
 0 = 屏蔽所有中断 (响应中断后自动清零)  
 1 = 总中断使能 (RETIE指令会将该位置1)

## 5.3 IOAICR 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAICR	-	-	IOAICR5	IOAICR4	IOAICR3	IOAICR2	IOAICR1	IOAICR0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	0	0	0	0	0	0

Bit [5:0]      **IOAICRx:** IOx端口变化中断使能(x=0-5)  
 0 = 屏蔽IOAx口电平变化中断  
 1 = 使能IOAx口电平变化中断

## 5.4 INTCR0 中断控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR0	TKIE	-	-	-	LVDIE	IOAIE	TC1IE	TC0IE
读/写	R/W	-	-	-	R/W	R/W	R/W	R/W
复位后	0	-	-	-	0	0	0	0

Bit 7      **TKIE:** 触摸转换完成中断  
           0 = 屏蔽触摸转换完成中断  
           1 = 使能触摸转换完成中断

Bit 3      **LVDIE:** 低电压检测中断  
           0 = 屏蔽低电压检测中断  
           1 = 使能低电压检测中断

Bit 2      **IOAIE:** 外部端口IOA中断  
           0 = 屏蔽外部端口IOA中断  
           1 = 使能外部端口IOA中断

Bit 1      **TC1IE:** TC1溢出中断  
           0 = 屏蔽TC1溢出中断  
           1 = 使能TC1溢出中断

Bit 0      **TC0IE:** TC0溢出中断  
           0 = 屏蔽TC0溢出中断  
           1 = 使能TC0溢出中断

## 5.5 INTF0 中断标志寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF0	TKIF	-	-	-	LVDIF	IOAIF	TC1IF	TC0IF
读/写	R/W	-	-	-	R/W	R/W	R/W	R/W
复位后	0	-	-	-	0	0	0	0

- Bit 7      **TKIF**: 触摸转换完成中断标志  
0 = 未产生触摸转换完成中断  
1 = 产生触摸转换完成中断
- Bit 3      **LVDIF**: 低压检测中断标志  
0 = 未产生低电压检测状态变化中断  
1 = 产生低电压检测状态变化中断, 只要VDD电压相比VLVDS设置电压发生状态改变时, 此标志就会被置“1”
- Bit 2      **IOAIF**: 外部端口IOA中断标志  
0 = 对应输入端口状态未发生变化  
1 = 对应输入端口状态发生变化
- Bit 1      **TC1IF**: TC1溢出中断标志  
0 = 未产生TC1溢出中断  
1 = 产生TC1溢出中断
- Bit 0      **TC0IF**: TC0溢出中断标志  
0 = 未产生TC0溢出中断  
1 = 产生TC0溢出中断

# 6 端口

## 6.1 IOA 数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOA	-	-	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	X	X	X	X	X	X

## 6.2 OEA 方向寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEA	LDSA3	LDSA2	OEA5	OEA4	OEA3	OEA2	OEA1	OEA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7:6] **LDSAx**: A2和A3输出能力选择 (x=2,3)

0 = 普通驱动(IoL1/ IoH1)

1 = 小驱动(IoL2/ IoH2)

Bit [5:0] **OEAx**: A口输出使能 (x=0-5)

0 = 输入

1 = 输出

**注：仿真模式下 LDSA3 位必须保持为 0。**

## 6.3 PUA 上拉使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUA	PDA5	PDA3	PUA5	PUA4	PUA3	PUA2	PUA1	PUA0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	1	0	0	0	0	0	0	0

Bit[7:6] **PDAX:** A3和A5口下拉使能 (x=3, 5)

0 = 下拉关闭

1 = 下拉使能

Bit [5:0] **PUAx:** A口上拉使能 (x=0-5)

0 = 上拉关闭

1 = 上拉使能

**注: IOA5 上电后默认下拉, 下拉电阻 5.1K±30%。**

## 6.4 IOAICR 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOAICR	VINS	-	IOAICR5	IOAICR4	IOAICR3	IOAICR2	IOAICR1	IOAICR0
读/写	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7] **VINS:** 端口输入电平选择

0 = VDD/2

1 = SMT

**注: 仿真模式下, 输入电平选择为 VDD/2, VINS 无效。**

Bit [6] **Reserved:** 保留, 必需保持为0

Bit [5:0] **IOAICRx:** IO端口变化中断使能(x=0-5)

0 = 屏蔽IOA口电平变化中断

1 = 使能IOA口电平变化中断

# 7 定时器 0(TCO)

## 7.1 概述

TCO 为带有可设置 1:128 预分频器, 1:16 后分频及周期寄存器的 8 位/16 位定时计数器, 具有绿色模式下唤醒功能。

在 8 位模式下, TCOCL 作为 TCO 的周期寄存器, TCO 使能后, TCOCH 递加, 当 TCOCH 与 TCOCL 数值相等时, TCO 溢出, 将 TCOCH 清零重新开始计数, 同时将中断标志位 TCOIF 置 1。

在 16 位模式下, [TCOCH,TCOCL]作为 16 位的计数器, TCO 使能后, 16 位计数器递加, 当计数值等于 0xFFFF 时, 16 位计数器将清零重新开始计数, 同时将中断标志位 TCOIF 置 1。

- 可选择时钟源: 高频系统时钟 Fosch、低频系统时钟 Foscl 和指令时钟 Fcpu 、外部时钟 T0CKI/触摸按键
- 可选择 8 位模式和 16 位模式
  - 8 位模式下, 通过设置周期寄存器, 可任意设置 TCO 的周期
- 预分频比多级可选, 最大可选择 1:128
- 溢出中断功能
- 溢出中断唤醒功能 (当输入频率选择 Foscl 或 Fosch 时, 若所选择的时钟源振荡器一直工作, 此时 TCO 在绿色模式下依然工作, 溢出中断可唤醒 CPU)

PWM0 信号的占空比计算如下:

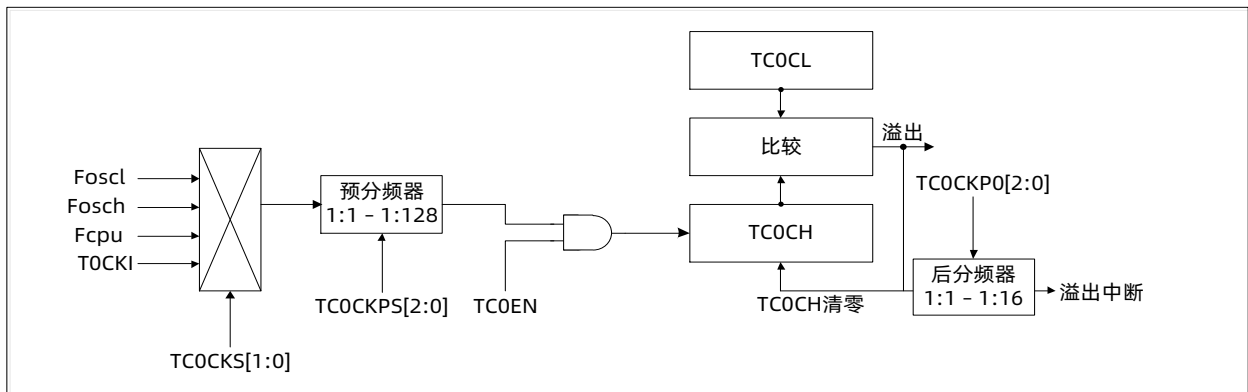
- ◇ 高电平时间 =  $PWM0xD \times (1 / (TCO \text{ 时钟源} / \text{预分频比}))$
- ◇ 周期 =  $(TCOCL+1) \times (1 / (TCO \text{ 时钟源} / \text{预分频比}))$
- ◇ 占空比:

正相输出 (PWMPOL = 0): 占空比 =  $PWM0xD / (TCOCL+1)$

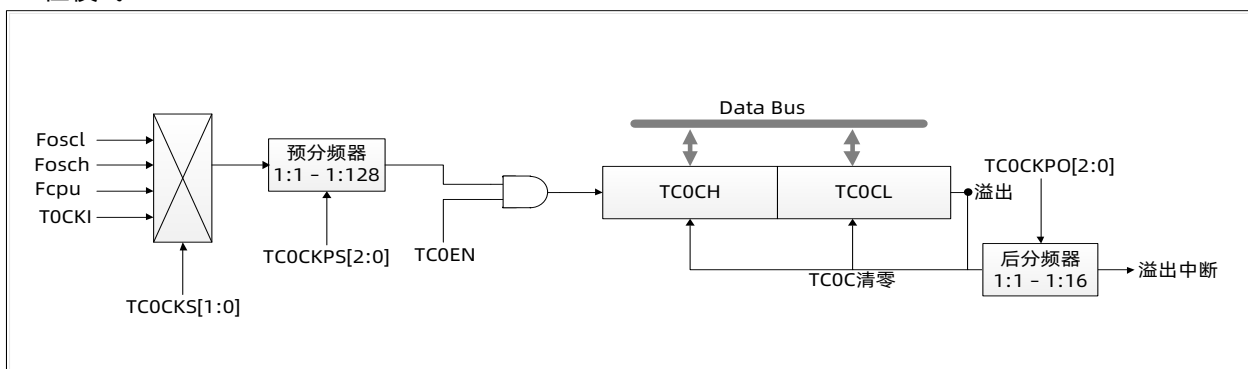
反相输出 (PWMPOL = 1): 占空比 =  $1 - (PWM0xD / (TCOCL+1))$

TCO 框图

8 位模式



16 位模式



## 7.2 TOCRO 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOCRO	TCOEN	TCOMOD	LEDM	TCOCKS[1:0]		TCOCKPS[2:0]		
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **TCOEN**: TCO模块使能位

0 = 关闭TCO

1 = 使能TCO

Bit 6 **TCOMOD**: TCO模式选择位

0 = 8位模式

1 = 16位模式

Bit 5 **LEDM**: PWM/串行LED模式选择

0 = PWM模式

1 = 串行LED模式

Bit [4:3] **TCOCKS**: TCO时钟源选择

TCOCKS[1:0]	TCO 时钟源选择
00	Foscl(低频系统时钟)
01	Fosch(高频系统时钟)
10	Fcpu
11	TOCKI

Bit [2:0] **TCOCKPS[2:0]**: TCO预分频比选择

TCOCKPS[2:0]	TCO 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

## 7.3 TOCR1 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TOCR1	LEDBUSY	PWMPOL	MPPWM01	MPPWM00	TCOCKPO[3:0]			
读/写	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit 7 **LEDBUSY**: 数据发送标志位

0 = 空闲, 可以写入数据

1 = 正在发送数据, 不允许写入

Bit 6 **PWMPOL**: PWM01/PWM03输出电平设置

0 = PWM01, PWM03正相输出

1 = PWM01, PWM03反相输出

Bit 5 **MPPWM01**: PWM01管脚映射

0 = 从PWM01A (IOA1) 输出

1 = 从PWM01B (IOA5) 输出

Bit 4 **MPPWM00**: PWM00管脚映射

0 = 从PWM00A (IOA0) 输出

1 = 从PWM00B (IOA4) 输出

Bit [3:0] **TCOCKPO[3:0]**: TC0后分频

TCOCKPO[3:0]	TC0 后分频比	TCOCKPO[3:0]	TC0 后分频比
0000	1:1	1000	1:9
0001	1:2	1001	1:10
0010	1:3	1010	1:11
0011	1:4	1011	1:12
0100	1:5	1100	1:13
0101	1:6	1101	1:14
0110	1:7	1110	1:15
0111	1:8	1111	1:16

## 7.4 TC0CL TC0 计数器低 8 位/周期寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TC0CL	TC0CL7	TC0CL6	TC0CL5	TC0CL4	TC0CL3	TC0CL2	TC0CL1	TC0CL0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	X	X	X	X	X	X	X	X

## 7.5 TCOCH TCO 计数器高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCOCH	TCOCH7	TCOCH6	TCOCH5	TCOCH4	TCOCH3	TCOCH2	TCOCH1	TCOCH0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	X	X	X	X	X	X	X	X

## 7.6 PWMOCR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMOCR	PWM03EN	PWM03OE	PWM02EN	PWM02OE	PWM01EN	PWM01OE	PWM00EN	PWM00OE
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

Bit [7,5,3,1] **PWM0xEN**: PWM0x模块使能位 (x=0-3)

0 = 关闭PWM0x

1 = 使能PWM0x

Bit [6,4,2,0] **PWM0xOE**: PWM0x波形输出使能位 (x=0-3)

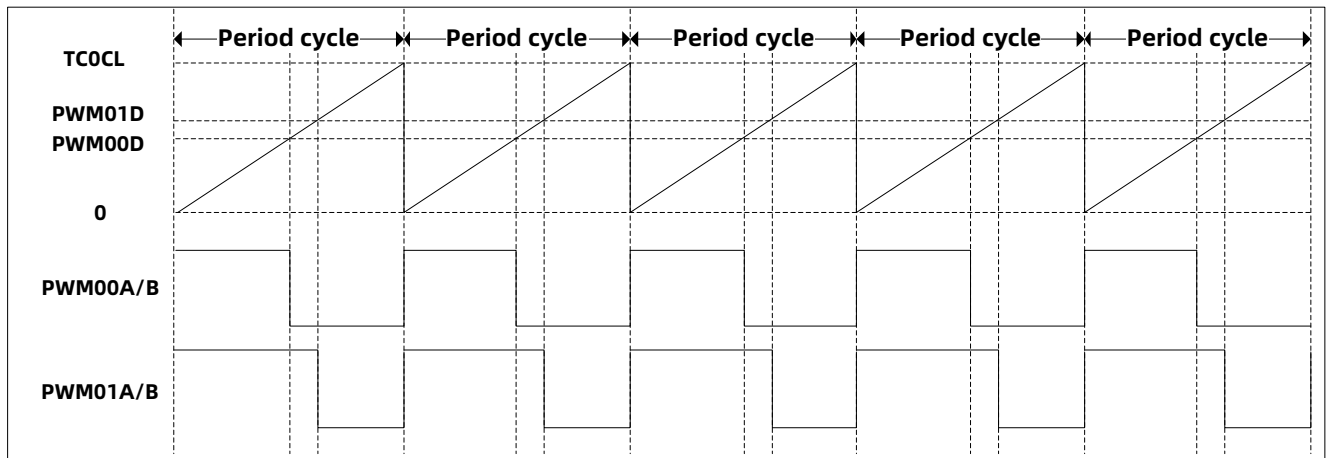
0 = 端口用作IO

1 = 端口输出PWM0x波形

## 7.7 PWM0xD 数据位(x =0,1,2,3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0xD	PWM0xD[7:0]							
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	0	0	0	0	0	0	0

## 7.8 PWMx 波形图(x=0,1)



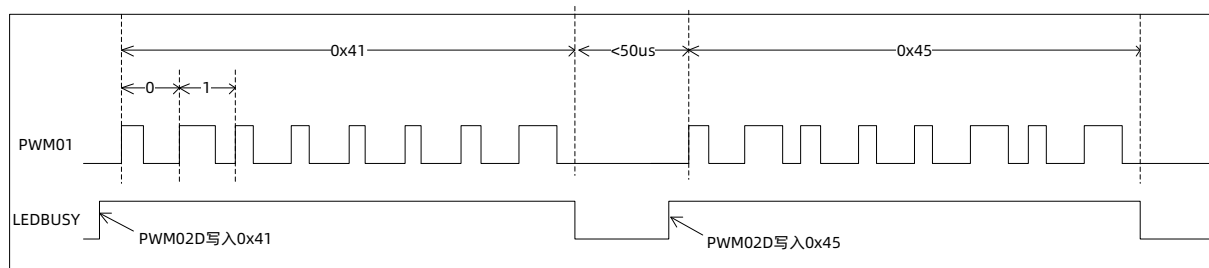
由上图可知 PWM00A/B&PWM01A/B 可以输出同周期，不同占空比的 PWM 波形。

## 7.9 串行 LED 模式功能

LEDM=1 时，使能 LED 功能，此时 TCO 后分频需要设置成 1:8 分频（单字节发送）；位周期由 TC0CL 控制，0 码的高电平宽度由 PWM00 控制，1 码的高电平宽度由 PWM01 控制，发送数据寄存器为 PWM02D，LED 输出端口由 PWM01 管脚映射控制，LEDBUSY=0 时，可以写入数据。

参考操作步骤：

- STEP1: TOCR0 写入 0xA8
- STEP2: TOCR1 写入 0x87 (1Byte 发送)
- STEP3: TC0CL 写入 19, 设置码周期为 1.25us
- STEP4: PWM00D 写入 5 (0 码高电平设置为 312.5ns)
- STEP5: PWM01D 写入 12 (1 码高电平设置为 750ns)
- STEP6: 写入 PWM02D 启动发送, PWM01 端口输出
- STEP7: 判断 LEDBUSY, 为 0 时写入 PWM02D 循环发送



- 注：**
- (1) 串行 LED 模式和 PWM 模式不能同时使用。
  - (2) 串行 LED 输出端口由 PWM01 管脚映射控制。
  - (3) 关于级联 LED 驱动应用请参照 DEMO 相关例程。

# 8 定时器 1(TC1)

## 8.1 概述

TC1 为 64us 定时器

- 时钟源：高频时钟 Fosch
- 溢出中断功能

## 8.2 T1CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T1CR	-	-	-	-	-	-	-	TC1EN
R/W	-	-	-	-	-	-	-	R/W
POR	-	-	-	-	-	-	-	0

Bit 0      **TC1EN**: TC1 模块使能

0 = 定时器关闭

1 = 定时器开启

# 9 触摸按键 (CDC)

## 9.1 概述

M8P5170 最多有 6 路触摸按键通道，灵敏度可通过外接电容调节，可替代机械式触摸按键，实现防水防尘，简单易用的操作接口。

**注：CDC 功能请使用官网对应的库。**

# 10 低电压检测(LVD)

## 10.1 概述

高精度低电压检测模块，可以精准检测供电电压(VDD)和外部输入电压。当 VDD 电压或外部输入电压相比 VLVDs 设置电压发生状态改变时，可产生 LVDIF 中断标志，查询 LVDF 即可了解当前 VDD 电压或外部输入电压与设置电压大小关系。

## 10.2 LVDCR 低电压检测控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCR	LVDEN	LVDF	CMPCHS	VLVDS[4:0]				
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **LVDEN**: 低压检测使能

0 = 关闭低压检测

1 = 使能低压检测

Bit 6 **LVDF**: 比较器输出位

0 = 检测电压高于设置电压

1 = 检测电压低于设置电压

Bit 5 **CMPCHS**: 检测电压选择位

0 = 检测VDD/2电压

1 = 检测外部输入口CP脚电压 (IOA2)

Bit [4:0] **VLVDS[4:0]**: LVD电压设置

VLVDS[4:0]	电压	VLVDS[4:0]	电压	VLVDS[4:0]	电压	VLVDS[4:0]	电压
00000	1.15V	01000	1.55V	10000	1.95V	11000	2.35V
00001	1.20V	01001	1.60V	10001	2.00V	11001	2.40V
00010	1.25V	01010	1.65V	10010	2.05V	11010	2.45V
00011	1.30V	01011	1.70V	10011	2.10V	11011	2.50V
00100	1.35V	01100	1.75V	10100	2.15V	11100	2.55V
00101	1.40V	01101	1.80V	10101	2.20V	11101	2.60V
00110	1.45V	01110	1.85V	10110	2.25V	11110	2.65V
00111	1.50V	01111	1.90V	10111	2.30V	11111	2.70V

# 11 看门狗 (WDT)

## 11.1 概述

看门狗定时器的时钟为内部独立 RC 时钟。

配置字 WDTEN 设置看门狗定时器的三种工作状态：

- (1) 始终开启 WDT 功能,即在 STOP 模式下仍然工作, 溢出可唤醒 STOP
- (2) 使能: 绿色或休眠模式下关闭, 即 STOP 下关闭
- (3) 屏蔽 WDT 功能, 即始终关闭

配置字 TWDTEN 设置看门狗的四种溢出时间: 4.5ms、18ms、72ms 或 288ms。

**注: 看门狗正常溢出后, 程序复位到 0000H, 但是在休眠模式下看门狗溢出程序是继续往下运行。**

## 11.2 OPTION 配置寄存器

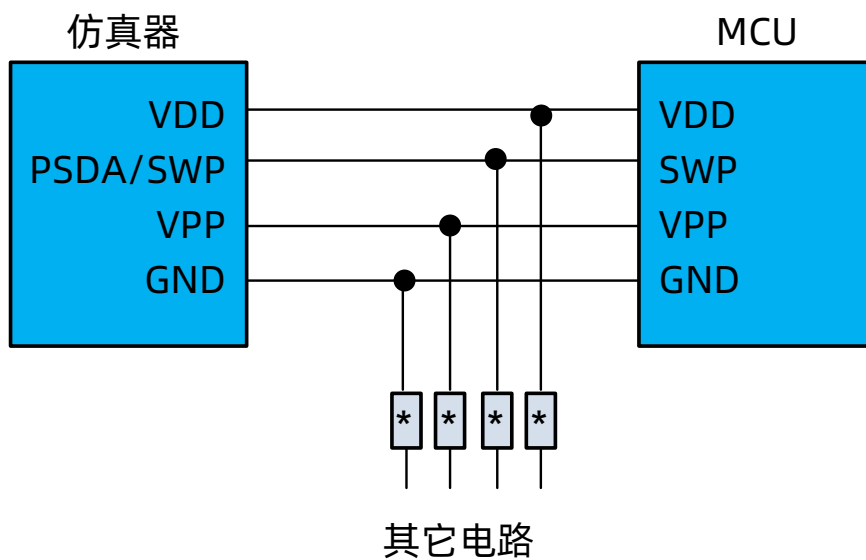
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIE	-	TO	PD	-	-	-	-
读/写	R/W	-	R	R	-	-	-	-
复位后	0	-	1	1	-	-	-	-

- Bit 5      **TO:** 超时位  
           0 = WDT发生溢出  
           1 = 上电复位或清除WDT
- Bit 4      **PD:** 掉电位  
           0 = 进入休眠模式  
           1 = 上电复位或清除WDT

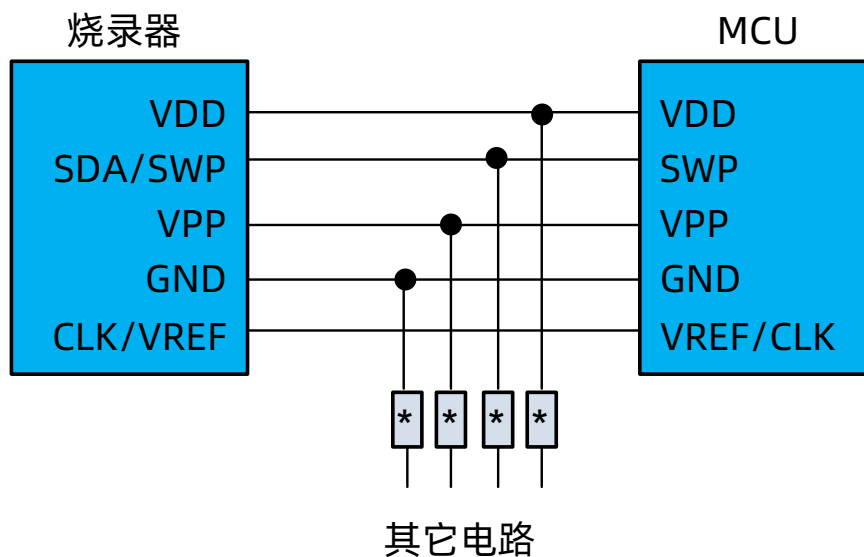
**注: CLRWDT 指令可清除 WDT 定时器。**

# 12 仿真烧录

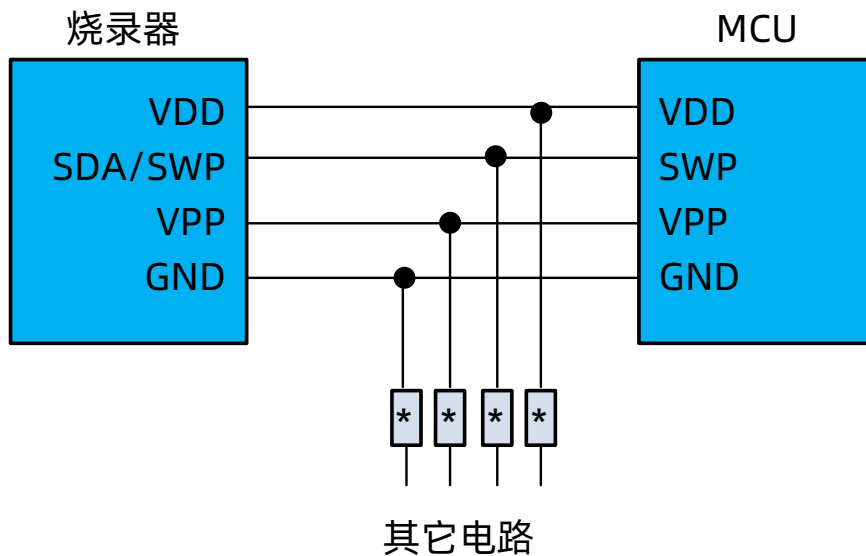
## 12.1 仿真连线示意图



## 12.2 校准模式连线示意图



## 12.3 默认（普通）连线示意图



注：（1）“\*”表示隔离或分离。隔离如采用电阻，应 $>3.3K\Omega$ 。分离是指和其它电路可以断开。

（2）VPP在烧录时会有10V左右的高压。

（3）烧录接口说明

### 1、仿真器烧录：

仿真器烧录时，烧录口分别是VDD、SWP（原PSDA）、VPP、GND。

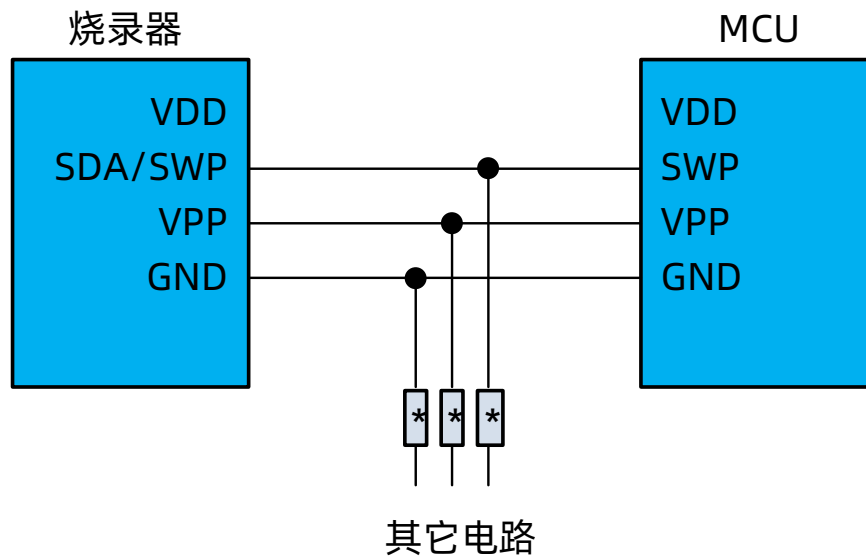
### 2、烧录器烧录：

烧录器烧录时，根据烧录参数设置不同，烧录接口有两种方式

（1）校准模式：烧录口分别是VDD、SWP（原SDA）、VPP、VREF(CLK)、GND。

（2）默认普通：烧录口分别是VDD、SWP（原SDA）、VPP、GND。

## 12.4 带电烧录连线示意图



注：

- 1、“\*”表示隔离或分离。隔离如采用电阻，应 $>3.3K\Omega$ 。分离是指和其它电路可以断开。
- 2、烧录器和芯片的VDD不需要连接，烧录信号由烧录器提供逻辑电平，具体参照烧录软件配置说明。
- 3、VPP在烧录时会有10V左右的高压。
- 4、带电烧录时只能配置在板烧录选项。

## 12.5 带电烧录软件配置说明

针对一些带电板(带电池产品, 烧录时电池不方便拆卸)需要在板烧录时, 可以通过设置代码配置字和烧录器参数设置来完成烧录动作。

操作步骤如下:

- 1、烧录代码必须在配置字中配置 SWP 端口设置为: 任何情况下有效。

CPU运行速度选择	8T(Vlvr>1.8V)
WDT使能选择	屏蔽WDT功能
WDT溢出时间	72ms
复位电压选择	Vlvr=2.4V
<b>SWP 端口设置</b>	<b>任何情况下有效</b>
输出端口读取	从输出寄存器读取
外部复位端口选择	作为IO端口
芯片代码加密	不使能

**注: SWP 端口设置作为任何情况下有效, 该口需设成数字输入模式或通过软件模式设置可保证在烧录时为数字输入模式。**

- 2、烧录参数配置相关配置项: 芯片管脚检测控制、编程电源电压、进烧录模式方式和烧录完成动作。



烧录参数设置窗口截图，显示了以下配置项：

- 文件存储空间选择: 自动存储
- 文件存储空间擦除: 全部擦除
- 芯片型号显示设置: 显示芯片型号
- 芯片管脚检测控制: 不检测芯片管脚 (1)
- 机台信号设置: 机台信号1(默认)
- 烧录启动方式选择: 等待按键启动
- 自动检测启动间隔: 0S
- HIRC校准电压: 固件默认
- 编程电源电压: 外部3.3V (2)
- 进烧录模式方式: MCU RST管脚复位 (3)
- 烧录完成动作: 复位运行 (4)

- ①、芯片管脚检测选择: 不检测芯片管脚
- ②、编程电源电压: 外部 3.3V/外部 5.0V(指烧录信号电压), 目标板是其它供电电压则就近选择对应的电压。
- ③、进烧录模式方式: MCU RST 管脚复位
- ④、烧录完成动作: 一般默认--复位运行, 如果选择--无动作, 则需要客户自行复位目标板
- ⑤、如果按上述操作有异常可联系我方工程师寻求帮助

# 13 芯片配置字 (OPTION BIT)

烧录选项	内容	说明
CPU 运行速度选择	4T (LVR>2.1V)	高频模式下 CPU 速度选择
	8T	
	16T	
	32T	
SWP 端口设置	仅复位过程	
	任何情况下 (A3 口需设为输入)	
输出 IO 读取设置	读输出寄存器	
	读端口电平	
复位电压选择	LVR=1.8V (FCPU<4T)	系统高速运行时, 请选择相应较高的 LVR 电压, 以保证系统的可靠性
	LVR=2.0V (FCPU<4T)	
	LVR=2.2V	
	LVR=2.4V	
	LVR=2.6V	
WDT 使能选择	始终开启 WDT 功能	
	使能: 绿色或休眠模式下关闭	
	屏蔽 WDT 功能	
WDT 溢出时间	WDT 溢出时间=4.5ms	VDD=5V 典型值
	WDT 溢出时间=18ms	
	WDT 溢出时间=72ms	
	WDT 溢出时间=288ms	
仿真电压选择	VDD 5.0V (<200mA)	
	VDD 3.3V (<300mA)	
	外供电源	

# 14 电性参数

## 14.1 极限参数

储存温度.....-50°C~125°C  
 工作温度.....-40°C~85°C  
 电源供应电压.....0V~5.5V  
 端口输入电压.....GND-0.3V~VDD+0.3V

**注：如果器件工作条件超出上述极限参数，将造成器件永久性破坏。如果在极限参数最大值上长时间工作，器件稳定性会受到影响。为保障器件稳定运行请在规定范围内工作。**

## 14.2 直流特性

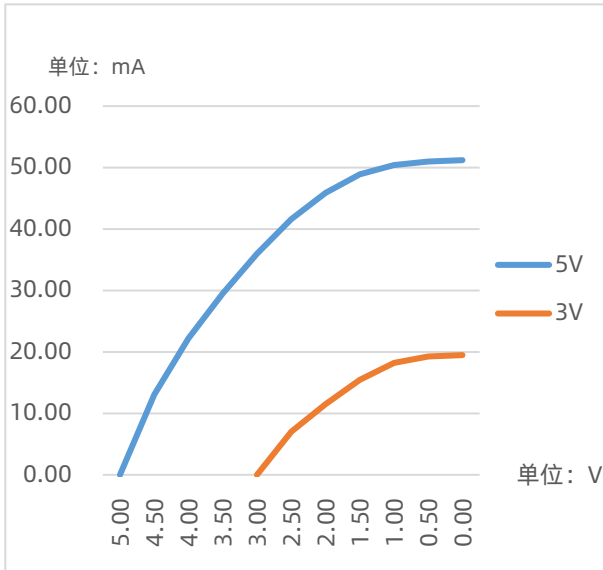
符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件 (常温 25°C)				
V <sub>DD</sub>	工作电压	—	F <sub>osc</sub> = 16MHz, 16T	2.0	-	5.5	V
I <sub>DD1</sub>	工作电流 1	3V	高频运行 (HIRC=16M) 低频运行 (LIRC=64K) F <sub>CPU</sub> =HIRC/16T	-	0.6	-	mA
		5V	全速工作	-	0.9	-	
I <sub>SP1</sub>	静态电流 1	3V	高频运行 (HIRC=16M), 低频运行 (LIRC=64K) STOP =1	-	120	-	uA
		5V	无唤醒源	-	200	-	
I <sub>SP2</sub>	静态电流 2	3V	高频停止 低频运行 (LIRC=64K)	-	4.5	-	
		5V	STOP =1 无唤醒源	-	7	-	
I <sub>SP3</sub>	静态电流 3	3V	高频停止 低频停止	-	0.5	-	
		5V	STOP =1 无唤醒源	-	0.7	-	
I <sub>SP4</sub>	静态电流 4	3V	高频停止 (HIRC=16M) 低频运行 (LIRC=64K), F <sub>CPU</sub> =HIRC/16T, STOP =1,	-	4.5	-	uA
		5V	WDT 唤醒 (72ms)	-	7	-	

符号	参数	测试条件		最小值	典型值	最大值	单位
		VDD	条件 (常温 25°C)				
R <sub>PH</sub>	上拉	5V	VIN = GND	-	21	-	kΩ
		3V		-	39	-	
R <sub>PD</sub>	下拉	5V	VIN=VDD	-	5.9	-	kΩ
		3V		-	6.5	-	
I <sub>OH1</sub>	输出拉电流	5V	输出口, V <sub>out</sub> =VDD-0.6V	-	16	-	mA
		3V		-	8	-	
I <sub>OL1</sub>	输出灌电流	5V	输出口, V <sub>out</sub> =GND+0.6V	-	37	-	
		3V		-	21	-	
I <sub>OH2</sub>	输出拉电流	5V	输出口, V <sub>out</sub> =VDD-0.6V	-	0.35	-	
		3V		-	0.22	-	
I <sub>OL2</sub>	输出灌电流	5V	输出口, V <sub>out</sub> =GND+0.6V	-	1.50	-	
		3V		-	1.15	-	

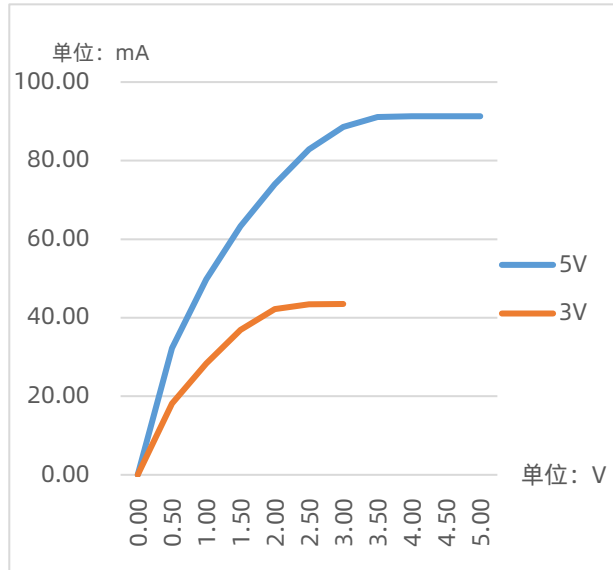
**注：具体值不做设计保证。**

## 14.3 IO 口拉灌电流特性

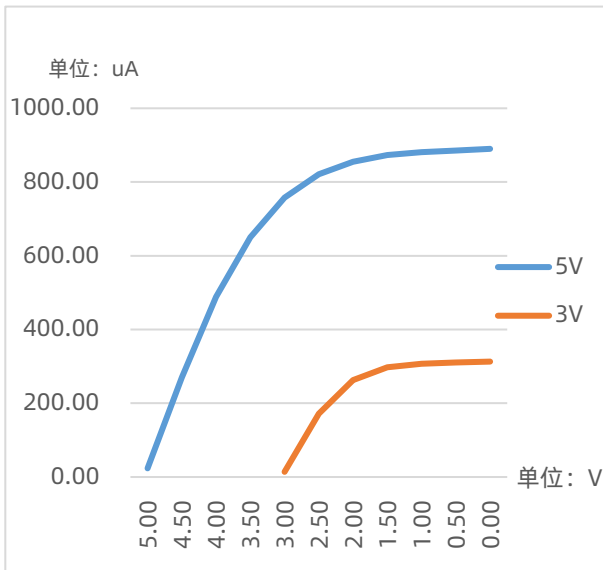
IOH<sub>1</sub>



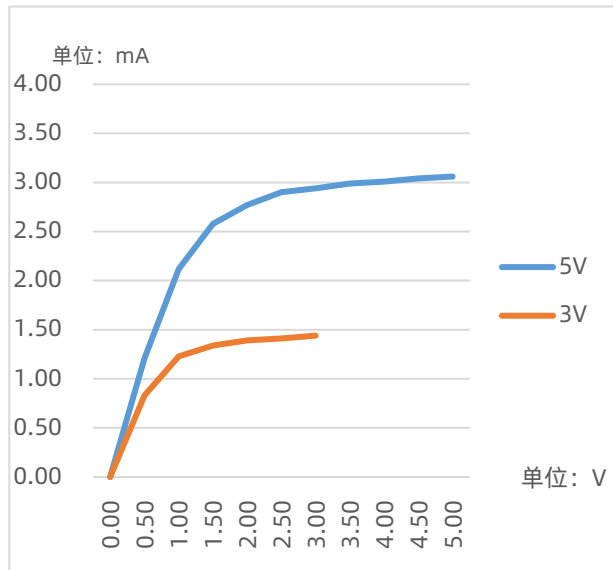
IOL<sub>1</sub>



IOH<sub>2</sub>



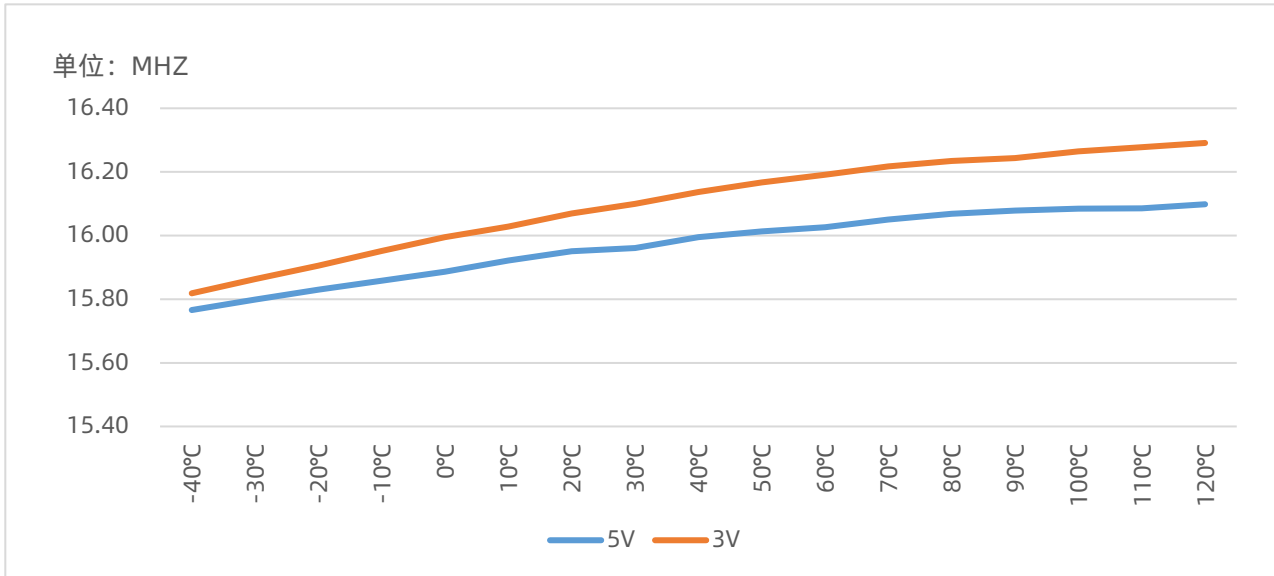
IOL<sub>2</sub>



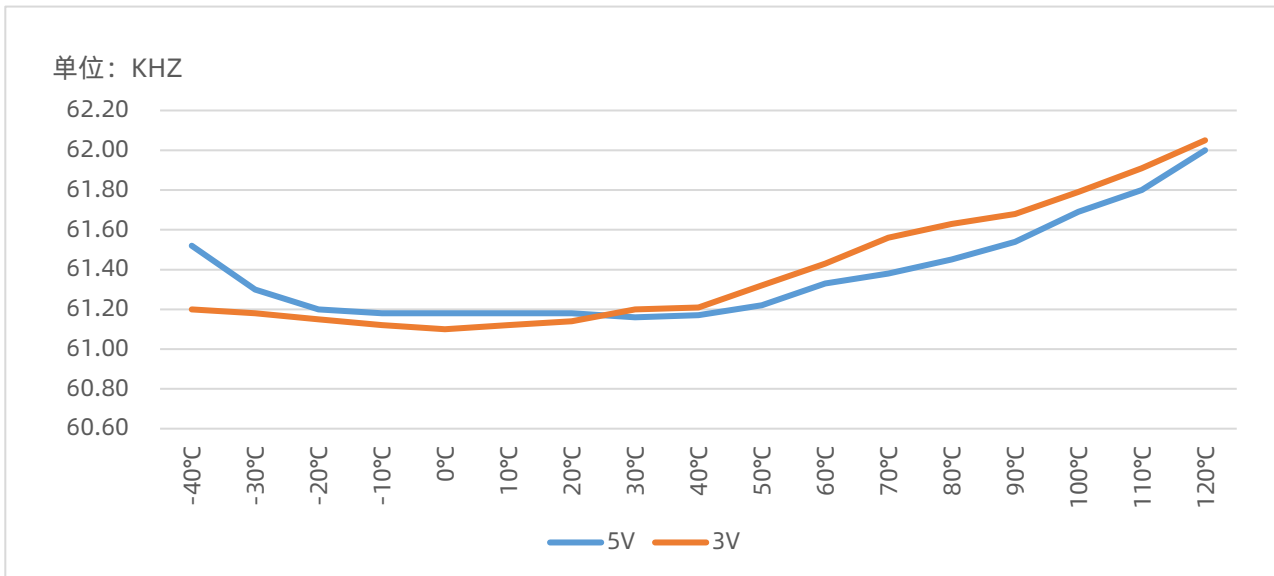
注: 具体值不做设计保证。

## 14.4 系统时钟特性

高频时钟（16MHZ）随温度变化曲线

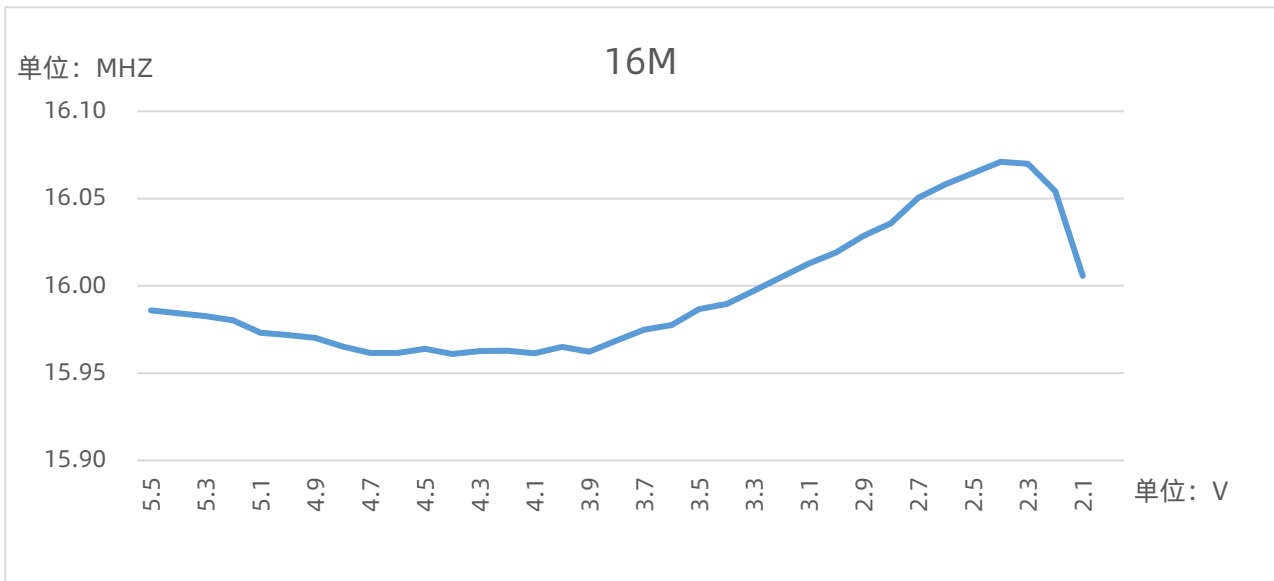


低频时钟（64KHZ）随温度变化曲线

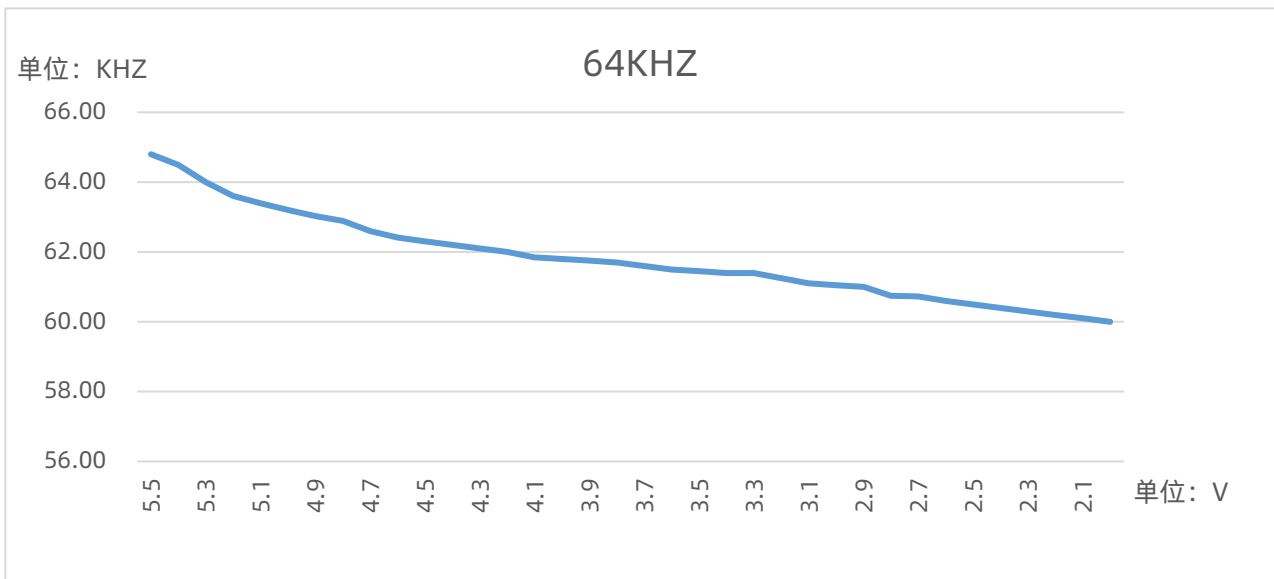


注：具体值不做设计保证。

常温下 (25°C) , 高频时钟 (16MHZ) 随电压变化曲线



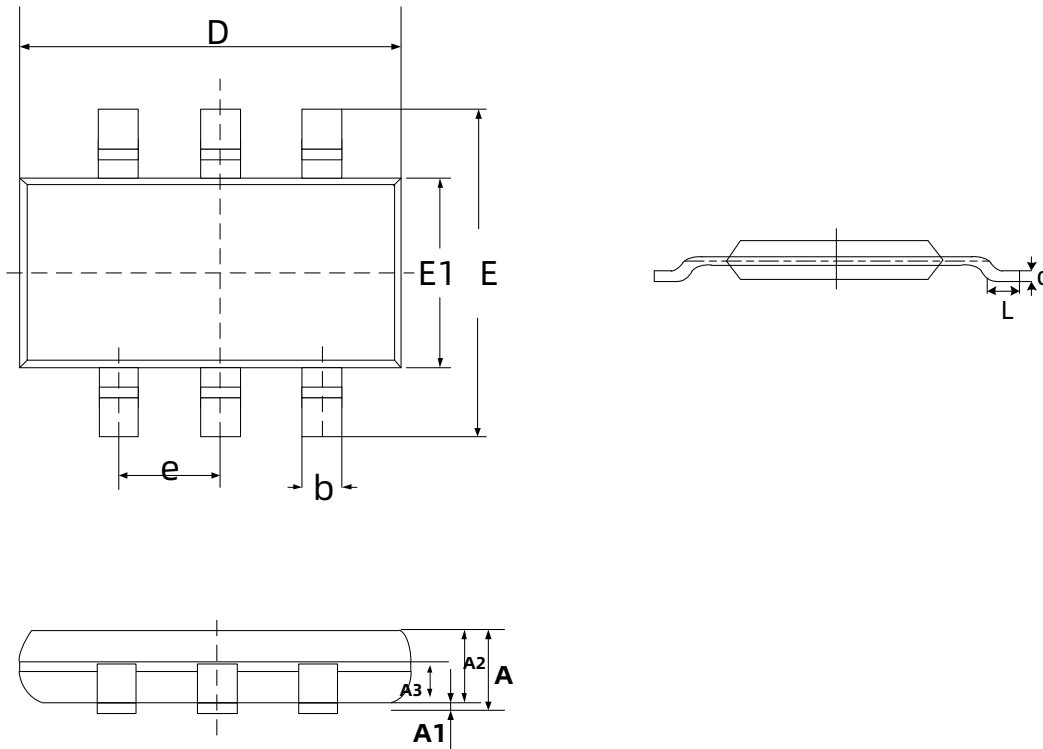
常温下 (25°C) , 低频时钟 (64KHZ) 随电压变化曲线



注: 具体值不做设计保证。

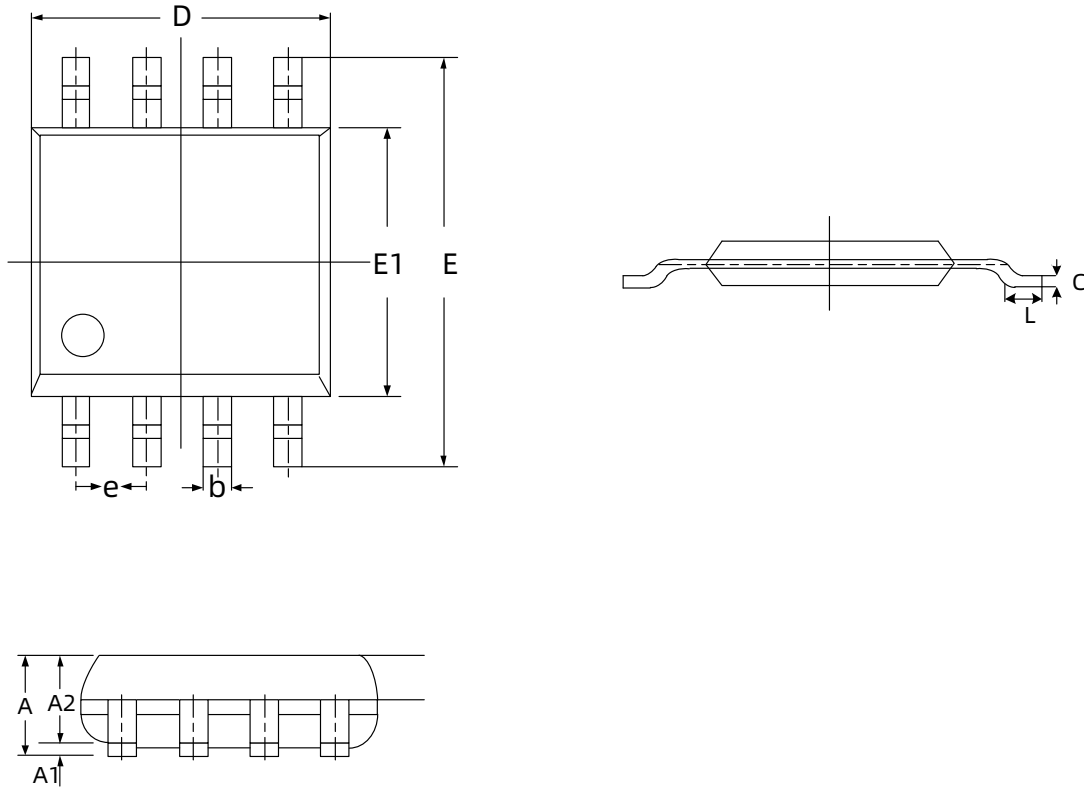
# 15 封装信息

## 15.1 SOT23-6



符号	单位 (mm)		
	最小	正常	最大
A	0.90	-	1.45
A1	0.00	-	0.15
A2	0.90	1.10	1.30
A3	0.50	0.65	0.80
D	2.80	2.95	3.1
E1	1.50	1.60	1.75
E	2.60	2.80	3.00
b	0.30	0.40	0.50
e	0.95BSC		
L	0.30	0.45	0.60
c	0.10	0.15	0.20

## 15.2 SOP8



符号	单位 (mm)		
	最小	正常	最大
A	1.35	-	1.75
A1	-	-	0.25
A2	1.35	1.40	1.50
b	0.30	-	0.51
D	4.70	4.90	5.10
e	1.27BSC		
E1	3.70	3.90	4.10
E	5.75	6.00	6.25
L	0.40	0.60	0.80
c	-	0.20	-

# 16 指令集简述

## 16.1 概述

M8Pxxx系列指令集是一种精简指令集 (RISC)，指令宽度为16位，由操作码和0~2个操作数组成。指令按照功能可分为5类，即字节操作指令、位操作指令、立即数指令、分支指令、特殊控制指令。

一个指令周期由1个系统时钟周期组成，除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特征情况，指令执行需要两个指令周期。

任何一条指定文件寄存器作为指令一部分的指令都进行读-修改-写操作。读寄存器、修改数据并根据指令或目标标识符“d”存储结果。即使是写寄存器的指令也将先对改寄存器进行读操作。

## 16.2 符号说明

符号	范围	说明	符号	范围	说明
R/r	0-0x1ff	寄存器地址	C	-	进位标志
A	-	ACC 寄存器	DC	-	半进位标志
B/b	0-7	位地址	Z	-	零标志
I/i	0-0xff	立即数	d	0-1	目的操作数定义
K/k	0-0x1fff	标号	GIE	-	总中断使能位
TOS	-	栈顶	stkp	-	堆栈指针
PC	-	PC 指针			

## 16.3 M8Pxxx 指令集表

指令集表中, d=1, 目的操作数为 R; d=0, 目的操作数为 A

指令类型	助记符	指令说明	周期数	影响标志位	备注
寄存器操作指令	ADDAR R,d	$R+A \rightarrow d$	1	Z,DC,C	
	ADCAR R,d	$R+A+C \rightarrow d$	1	Z,DC,C	
	SUBAR R,d	$A-R \rightarrow d$	1	Z,DC,C	
	SBCAR R,d	$A-R-C \rightarrow d$	1	Z,DC,C	
	SUBRA R,d	$R-A \rightarrow d$	1	Z,DC,C	
	SBCRA R,d	$R-A-C \rightarrow d$	1	Z,DC,C	
	ANDAR R,d	$R \& A \rightarrow d$	1	Z	
	ORAR R,d	$R A \rightarrow d$	1	Z	
	XORAR R,d	$R \wedge A \rightarrow d$	1	Z	
	COMR R,d	$R \rightarrow d$	1	Z	
	MOVR R,d	$R \rightarrow d$	1	Z	
	MOVAR R	$A \rightarrow R$	1	-	
	CLRR R	$0 \rightarrow R$	1	Z	
	SWAPR R,d	R 半字节交换 $\rightarrow d$	1	-	
	RLR R,d	$R[7] \rightarrow C, \{R[6:0], C\} \rightarrow d$	1	C	
	RLRNC R,d	$\{R[6:0], 0\} \rightarrow d$	1	-	
	RRRR,d	$R[0] \rightarrow C, \{C, R[7:1]\} \rightarrow d$	1	C	
	RRRNC R,d	$\{0, R[7:1]\} \rightarrow d$	1	-	
	DECR R,d	$R-1 \rightarrow d$	1	Z	
	DJZR R,d	$R-1 \rightarrow d, \text{SKIP if } 0$	1(2)	-	
	INCR R,d	$R+1 \rightarrow d$	1	Z	
	JZR R,d	$R+1 \rightarrow d, \text{SKIP if } 0$	1(2)	-	
	JNZR R,d	$R+1 \rightarrow d, \text{SKIP if } !0$	1(2)	-	
	DJNZR R,d	$R-1 \rightarrow d, \text{SKIP if } !0$	1(2)	-	
	JCMPEAR R	SKIP if $A=R$	1(2)	Z,C	
	JNCMPAR R	SKIP if $A \neq R$	1(2)	Z,C	
	JGAR R	SKIP if $A \geq R$	1(2)	Z,C	
	JLAR R	SKIP if $A < R$	1(2)	Z,C	
XCHAR R	$A \leftrightarrow R$	1	-		
位操作指令	JBTS0 R,b	SKIP if $R[b]=0$	1(2)	-	
	JBTS1 R,b	SKIP if $R[b]=1$	1(2)	-	
	BCLR R,b	$0 \rightarrow R[b]$	1	-	
	BSET R,b	$1 \rightarrow R[b]$	1	-	

指令类型	助记符	指令说明	周期数	影响标志位	备注
立即数操作指令	ADDIA I	I+A → A	1	Z,DC,C	
	ADCIA I	I+A+C → A	1	Z,DC,C	
	SUBIA I	I-A → A	1	Z,DC,C	
	SBCIA I	I-A-C → A	1	Z,DC,C	
	SUBAI I	A-I → A	1	Z,DC,C	
	SBCAI I	A-I-C → A	1	Z,DC,C	
	ANDIA I	A&I → A	1	Z	
	ORIA I	A I → A	1	Z	
	XORIA I	A^I → A	1	Z	
	MOVIA I	I → A	1	-	
	RETIA I	Stack → PC, I → A	2	-	
	JCMPAI I	SKIP if A=I	1(2)	Z,C	
	JNCPAII	SKIP if A≠I	1(2)	Z,C	
	特殊操作指令	RLA	A[7] → C, {A[6:0],C} → A	1	C
RLANC		{A[6:0],0} → A	1	-	
RRA		A[0] → C, {C,A[7:1]} → A	1	C	
RRANC		{0,A[7:1]} → A	1	-	
DECA		A-1 → A	1	Z	
DJZA		A-1 → A, SKIP if 0	1(2)	-	
INCA		A+1 → A	1	-	
JZA		A+1 → A, SKIP if 0	1(2)	-	
RETIE		Stack → PC, 1 → GIE	2	-	
RETURN		Stack → PC	2	-	
NOP		None Operation	1	-	
RDT		ROM[{{fsr1,fsr0}}] → {HBUF, A}	3	-	
DAA		加法后十进制调整	1	DC, C	
DSA		减法后十进制调整	1	DC, C	
PUSH		A, STATUS 压栈	1	-	
POP		A, STATUS 出栈	1	Z, DC, C	
CLRWDT		清除 WDT 寄存器	1	PD, TO	
分支指令	CALL I	I → PC, PC → Stack	2	-	
	GOTO I	I → PC	2	-	

## 16.4 M8Pxxx 指令说明

指令集详细说明请到官网下载：

[M8Pxxx 指令说明](#)

## 17 修改记录

版本	日期	描述
V1.00	2025-07-15	初版
---	---	---
V1.05	2026-06-06	修改烧录方式配置名