

M8P615

**8BIT
AD+TK 型
MTP MCU**

Version 2.00

2020 年 11 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。

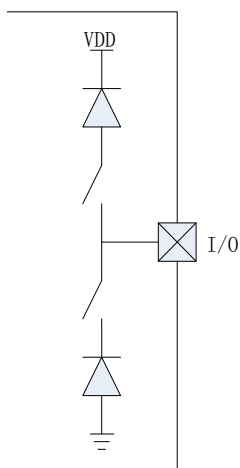
注意

■ 端口上/下拉特性

- 问题描述

M8P615的I/O口上/下拉特性类似用一个二极管接到VDD/GND(如下图), 所以实际拉到的电平是与VDD/GND相差一个二极管的开启电压。

在一些应用场合, 如工作电压较低时的按键判断, 上拉电平有可能达不到端口内部Schmitt输入的高翻转点, 而造成判断错误。



- 解决方案

1. 根据端口特性进行外部应用电路设计。

2. 如做按键判断, 可以:

- (1)使用IOB5, 这个端口的上拉是电阻型的。

- (2)使用IOB0或IOA3, 并设置寄存器IOHDS中的Bit7[LTS1]/Bit6[LTS0]为1, 则对应端口为非Schmitt的单一低翻转电平特性。

■ 电源口处理

- 问题描述

此单片机AD采集时, 易受电源噪声干扰。

- 解决方案

MCU电源口VDD&GND口并联104电容, 104电容位置应紧靠IC, 电源走线也应先进入104电容再进入MCU。

目录

1 产品简述	7
1.1 特性	7
1.2 引脚图	8
1.2.1 16Pin	8
1.2.2 8Pin	8
1.2.3 DFN8	8
1.3 引脚描述	9
2 中央处理器（CPU）	11
2.1 程序存储器	11
2.1.1 复位向量（0000H）	11
2.1.2 中断向量（0008H）	12
2.1.3 查表	12
2.2 数据存储器	14
2.2.1 数据存储器结构	14
2.2.2 数据存储器寻址模式	14
2.2.3 系统寄存器定义	15
2.2.4 INDF0 间接寻址寄存器 0	15
2.2.5 INDF1 间接寻址寄存器 1	15
2.2.6 FSR0 间接寻址指针 0	15
2.2.7 FSR1 间接寻址指针 1	15
2.2.8 HBUF 查表数据高 8 位	16
2.2.9 PCL 程序计数器指针低位	16
2.2.10 STATUS 状态寄存器	16
3 复位	17
3.1 复位方式	17
4 系统时钟	18
4.1 概述	18
4.2 OSCM 寄存器	18
4.3 系统时钟工作模式	19
4.4 IRCCAL 寄存器	20
4.5 系统时钟结构框图	21
5 中断	22
5.1 概述	22
5.2 OPTION 配置寄存器	22
5.3 IO 变化中断使能寄存器	23
5.4 INTCR0 中断控制寄存器 0	23
5.5 INTF0 中断标志寄存器 0	24
5.6 INTCR1 中断控制寄存器 1	25

5.7 INTF1 中断标志寄存器 1	25
5.8 中断范例	26
6 端口	28
6.1 IOA	28
6.2 IOB	30
6.3 IO 口下拉使能	32
6.4 IO 驱动选择	33
7 定时器 0/1(TC0/TC1)	34
7.1 概述	34
7.2 TxCR 控制寄存器 (x=0,1)	35
7.3 TCxCL TCx 计数器低 8 位/周期寄存器 (x=0,1)	36
7.4 TCxCH TCx 计数器高位 (x=0,1)	36
7.5 定时器范例	37
8 脉宽调制模块 PWM0	38
8.1 概述	38
8.2 PWM0CR 控制寄存器	38
8.3 PWM0D 数据寄存器	38
8.4 PWM0 范例	39
9 脉宽调制模块 PWM1	40
9.1 概述	40
9.2 PWM1CR 控制寄存器	40
9.3 PWM1DH 数据高位	41
9.4 PWM1DL 数据低位	41
9.5 PWMDEADT PWM1 死区控制寄存器	41
9.6 PWM 输出波形示例	42
9.6.1 互补 PWM 输出	42
9.6.2 带死区的互补 PWM 输出	43
9.6.3 8+4 位分辨率模式	43
9.6.4 有效电平选择	44
9.7 PWM1 范例	44
10 模数转换器(ADC)	46
10.1 概述	46
10.2 ADCON0 寄存器	46
10.3 ADCON1 寄存器	47
10.4 ADCON2 寄存器	48
10.5 ADH ADC 数据高字节	49
10.6 ADL ADC 数据低字节	49
10.7 ADC 范例	50
11 触摸按键 (CDC)	52

11.1 概述	52
11.2 原理框图	52
11.3 TKCTR0 控制寄存器 0	53
11.4 操作说明	53
12 看门狗 (WDT)	54
12.1 概述	54
12.2 OPTION 配置寄存器	54
12.3 WDTIC 看门狗控制寄存器	54
13 芯片配置字	55
14 电性参数	57
14.1 极限参数	57
14.2 直流特性	57
14.3 温度对系统时钟的影响	59
14.4 IO 口拉灌电流特性曲线	60
15 封装信息	62
15.1 SOP16	62
15.2 DIP16	64
15.3 SOP8	66
15.4 DIP8	68
15.5 DFN8	70
16 指令集简述	71
16.1 概述	71
16.2 符号说明	71
16.3 M8Pxxx 指令集表	72
16.4 M8Pxxx 指令说明	74
17 修正记录	75

1 产品简述

M8P615 是一颗采用高速低功耗 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 位多次擦写编程存储器（MTP，擦写次数 1000），64*8 位的数据存储器（RAM），14 个双向 I/O 口，2 个 8 位(带自动重载)/16 位定时器/计数器，2 路 PWM，8+3 路 12 位 AD 转换器，7 路触摸按键，支持多种系统工作模式和多个中断源。

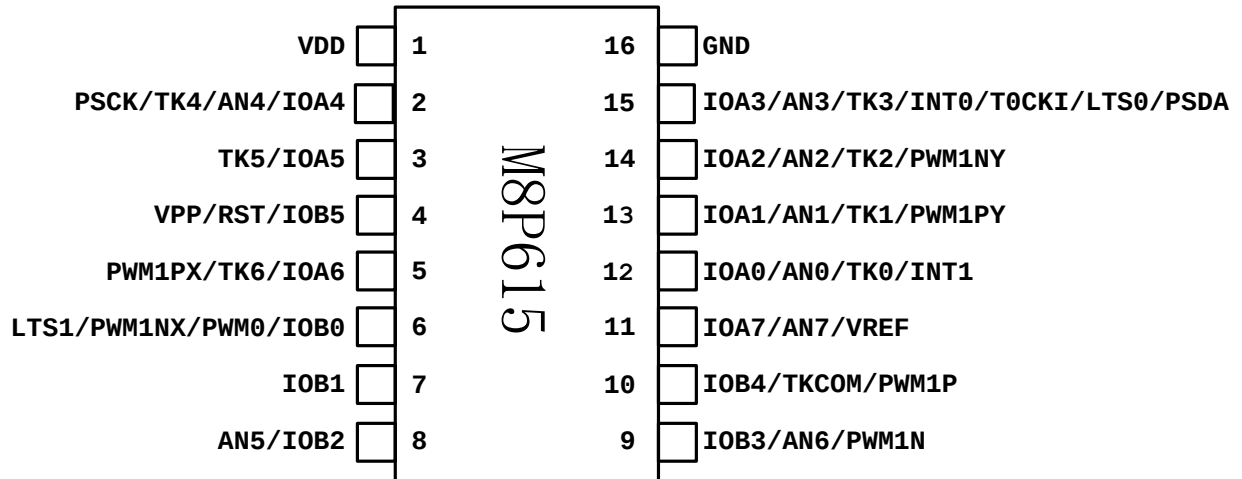
1.1 特性

- 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
 - 最多14个双向I/O口（IOB5开漏输出）
 - 可编程弱上拉/弱下拉口IOA/IOB
 - 支持IO口电平变化中断
 - 2个可变翻转点IO
- 2 个定时器/计数器
 - TC0/TC1: 8位(带自动重载)/16位定时器/计数器
- 系统时钟
 - 内部高速RC振荡器: 16MHz
 - 内部低速RC振荡器: 64KHz (5V)
- 系统工作模式
 - 普通模式
 - 绿色模式
 - 休眠模式
- 2 路 PWM
 - 时基可独立选择 TC0/TC1
 - 互补输出及死区控制
 - 8+4 位高分辨率
- 8+3 路 12 位 ADC
 - 内嵌参考电压2V、3V、4V、VDD
 - 8路外部输入
 - 1路内部电源电压检测VDD/4
 - 1路内部GND电压检测
 - 1路内部参考电压检测
- 7 路触摸按键扫描
- 多路中断源
 - 定时器中断: TC0/TC1
 - 外部中断: INT0/INT1
 - IO口电平变化中断
 - ADC转换中断
 - 触摸按键中断
- 看门狗定时器
- 特殊功能
 - 可编程代码保护
 - 多级LVR低压复位
- 封装形式
 - DIP16/SOP16
 - DIP8/SOP8
 - DFN

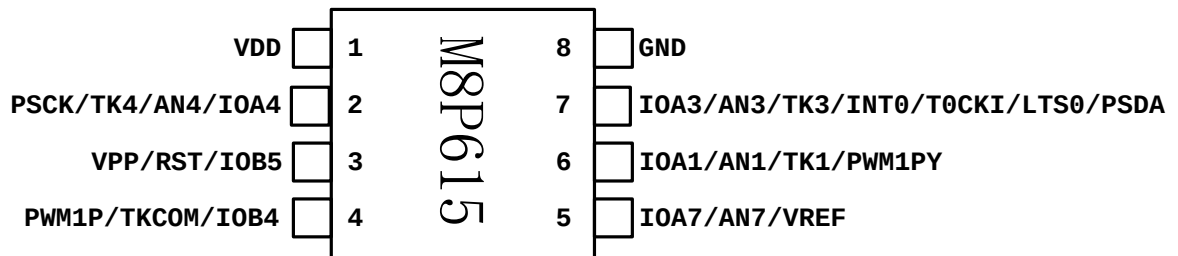
1.2 引脚图

注：芯片仿真烧录口分别是VDD、PSDA、PSCK、VPP、GND。

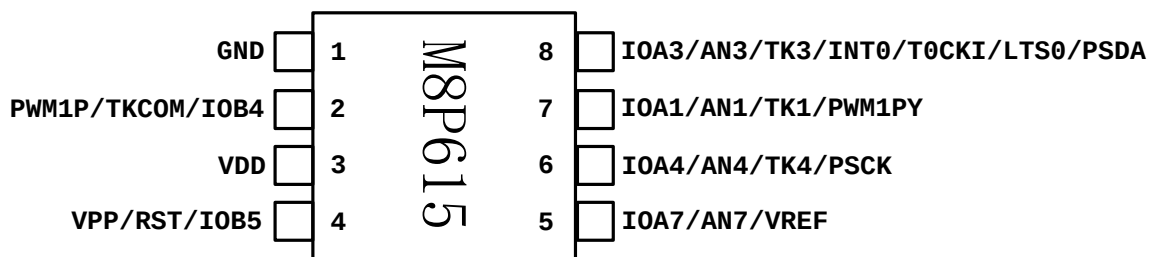
1.2.1 16Pin



1.2.2 8Pin



1.2.3 DFN8



1.3 引脚描述

名称	类型	说明
VDD, GND	P	电源输入端
IOA0	I/O	输入/输出 IO, SMT, 上拉电阻
AN[0]	A	AD 通道 0
TK[0]	A	触摸按键通道 0
INT1	I	外部中断
IOA1	I/O	输入/输出 IO, SMT, 上拉电阻
AN[1]	A	AD 通道 1
TK[1]	A	触摸按键通道 1
PWM1PY	O	PWM1P 输出端口
IOA2	I/O	输入/输出 IO, SMT, 上拉电阻
AN[2]	A	AD 通道 2
TK[2]	A	触摸按键通道 2
PWM1NY	O	PWM1N 输出端口
IOA3	I/O	输入/输出 IO, SMT, 上拉电阻
AN[3]	A	AD 通道 3
TK[3]	A	触摸按键通道 3
INT0	I	外部中断
LT_S0	I	可编程低翻转点输入
T0CKI	I	外部时钟输入
PSDA	I/O	编程用
IOA4	I/O	输入/输出 IO, SMT, 上拉电阻
AN[4]	A	AD 通道 4
TK[4]	A	触摸按键通道 4
PSCK	I/O	编程用
IOA5	I/O	输入/输出 IO, SMT, 上拉电阻
TK[5]	A	触摸按键通道 5
IOA6	I/O	输入/输出 IO, SMT, 上拉电阻
TK[6]	A	触摸按键通道 6
PWM1PX	O	PWM1P 输出端口
IOA7	I/O	输入/输出 IO, SMT, 上拉电阻
AN[7]	A	AD 通道 7
VREF	A	AD 外部参考电压输入
IOB0	I/O	输入/输出 IO, SMT, 上拉电阻, 电平变化中断
PWM0	O	PWM0 输出端口
PWM1NX	O	PWM1N 输出端口
LT_S1	I	可编程低翻转点输入
IOB1	I/O	输入/输出 IO, SMT, 上拉电阻, 电平变化中断
IOB2	I/O	输入/输出 IO, SMT, 上拉电阻, 电平变化中断
AN[5]	A	AD 通道 5



名称	类型	说明
IOB3	I/O	输入/输出 IO, SMT, 上拉电阻, 电平变化中断
AN[6]	A	AD 通道 6
PWM1N	O	PWM1N 输出端口
IOB4	I/O	输入/输出 IO, SMT, 上拉电阻, 电平变化中断
TKCOM	A	触摸按键灵敏度电容端口
PWM1P	O	PWM1P 输出端口
IOB5	I/O	输入/开漏输出 IO, SMT, 上拉电阻, 电平变化中断
VPP	P	编程高压电源

注: I = 输入 O = 输出 I/O = 输入/输出 P = 电源 A = 模拟信号
IOB5口是开漏口, 输出1时可开内部上拉使端口输出弱“1”。

2 中央处理器（CPU）

2.1 程序存储器

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（0000H）

M8P615有以下四种复位方式

- 上电复位
- 看门狗复位
- 外部复位
- 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   MainProgram    ;//跳转至用户程序开始
...
MainProgram:           ;//用户程序开始
...
Main:
...
GOTO   MAIN            ;//用户主程序循环
```

2.1.2 中断向量（0008H）

M8P615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     MainProgram      ;//跳转到程序开始
      ORG      0008H
      GOTO     Interrupt        ;//发生中断后，跳转到中断子程序
MainProgram:
      ...
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

查表有三种模式：

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11位程序区数据寻址指针。

例 1：查找 ROM 地址为“DTAB”的值

```

      MOVIA     0              ;//赋给A的值就是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

地址	间接寻址 INDF0	间接寻址 INDF1	间接寻址 INDF2	直接寻址
0x000 ~ 0x0FF	YES	YES	YES	YES

注：此芯片间接寻址INDF1直接映射到间接寻址INDF0。

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

✧ 间接选址模式 0

地址	0	FSR0
----	---	------

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

✧ 间接选址模式 1

地址	1	FSR1
----	---	------

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

注：本系列 RAM 只在 0x00~0xFF 存在，其它区域数据全部映射到 0x00~0xFF

✧ 间接选址模式 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 ~ 0x038	GPR							
0x040 ~ 0x0A8	RESERVE							
0x0B0	INDF0	FSR0	-	-	-	-	-	-
0x0B8	INDF1	FSR1	PCL	STATUS	OPTION	OSCM	WDTC	IOICR
0x0C0	INDF2	HBUF	-	-	INTCR0	INTF0	INTCR1	INTF1
0x0C8	IOA	OEA	PUA	ANSA	IOB	OEB	PUB	ANSB
0x0D0	-	-	-	-	-	IOLDS	IOHDS	PDAB
0x0D8	PWM0CR	PWM0D	-	-	PWM1CR	PWM1DH	PWM1DL	PWMDEADT
0x0E0	-	-	-	-	-	-	-	-
0x0E8	T0CR	TC0CL	TC0CH	-	T1CR	TC1CL	TC1CH	-
0x0F0	-	-	-	ADCON2	ADCON0	ADCON1	ADH	ADL
0x0F8	TKCTR0	-	-	-	-	-	-	-

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空间。

注：此芯片间接寻址INDF1直接映射到间接寻址INDF0。

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)
- 外部复位 (MCLR Reset)
- 欠压复位 (BOR)
- 看门狗定时器复位 (WDT Reset)

M8P615 有以上4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

M8P615由内置的16MHz RC振荡电路，Fosch（HIRC 16MHz）作为系统时钟源，内置低速时钟Foscl（LIRC 64KHz）仅作为定时器时钟源。

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：普通模式有两种分别是：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_{DD1}）

2.高频时钟停止，低频时钟工作，不进 STOP

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_{SP1}）

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_{SP2}）

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_{SP3}）

休眠模式可以由外部中断或 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	-	IRCAL6	IRCCAL5	IRCCAL4	IRCCAL3	IRCCAL2	IRCCAL1	IRCCAL0
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。

例：调整 IRC 频率

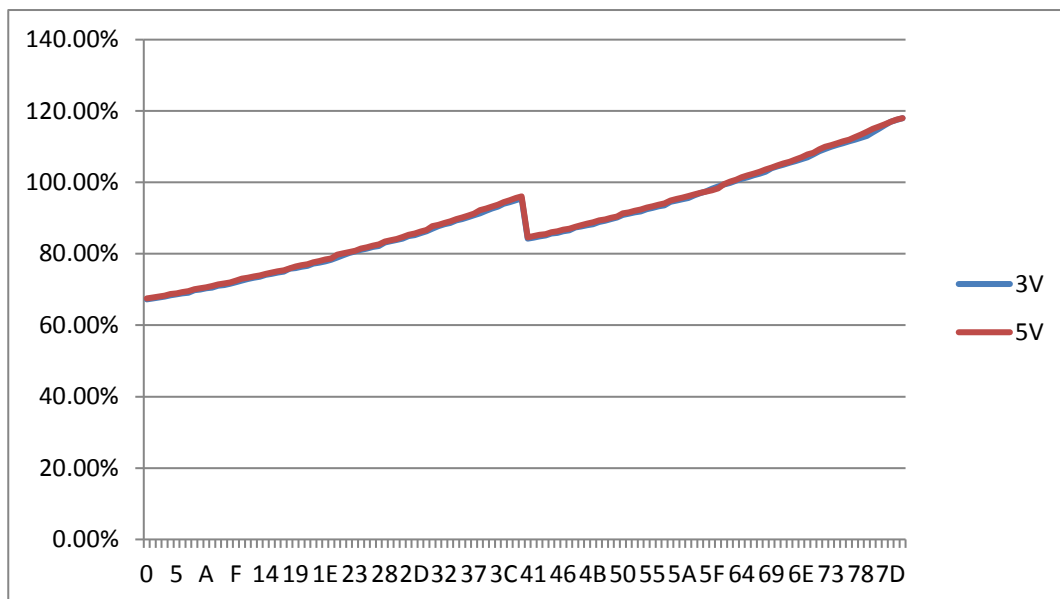
TASK_IRCCAL:

```

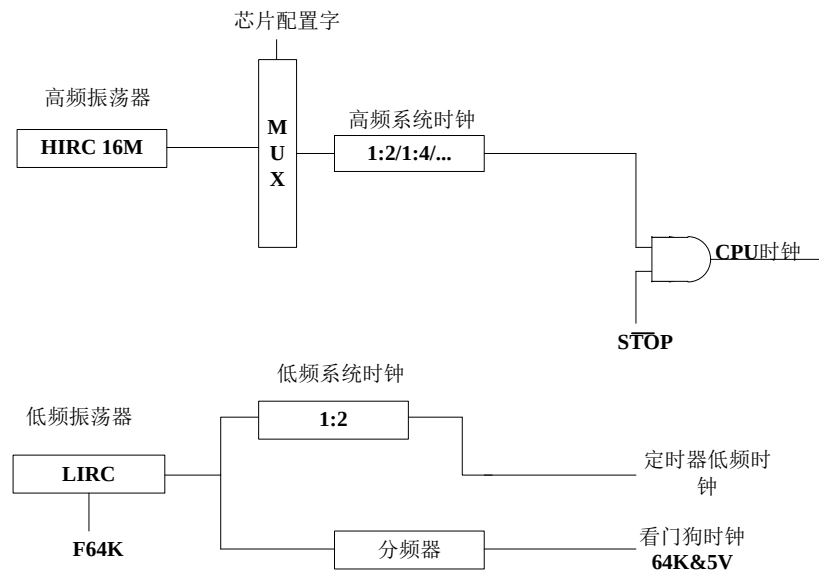
MOVIA    0x55
MOVAR    0x1F9          ;//1F9H地址写入055H
MOVIA    0xAA
MOVAR    0x1F9          ;//1F9H地址写入0AAH
MOVIA    VALUE          ;//VALUE值可选择为（0-127）
MOVAR    IRCCAL         ;//写入IRCCAL
...
;//若需继续在IRCCAL寄存器内写入其他值需要重复以上所有步骤
  
```

注：（1）IRCCAL 寄存器地址为 1FCH，写 0FCH 是无效的.进入修改模式所用的寄存器地址为 1F9H，写 0F9H 是无效的。
 （2）VALUE 可选为（0-127）具体值不做设计保证。

调整 IRCCAL 寄存器的值使 IRC 频率变化的趋势如下图：



4.5 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

M8P615有多路中断源：TC0/TC1，IOB口电平变化，INT0/INT1，ADC中断可以将系统从睡眠模式中唤醒，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，硬件自动将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。

注：使用外部中断 INT0、INT1 要注意，当中断触发和进休眠的操作（即 STOP 从 0 到 1）同时发生时，会导致外部中断 INT0、INT1 无效且无法唤醒休眠；
解决方法：如需使用外部中断尽量使用 IO 变化中断，如果必须使用外部中断 INT0、INT1，必须要开启 WDT 作为唤醒源。

5.2 OPTION 配置寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPTION	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
读/写	R/W	-	R	R	R/W	R/W	R/W	R/W
复位后	0	-	1	1	0	0	0	0

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）

Bit [3:2] **MINT1[1:0]:** INT1中断模式选择

MINT1[1:0]	INT1 中断模式选择
00	上升沿中断
01	下降沿中断
1X	变化中断

Bit [1:0] **MINT0[1:0]:** INT0中断模式选择

MINT0[1:0]	INT0 中断模式选择
00	上升沿中断
01	下降沿中断
1X	变化中断

5.3 IO 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOICR	IOA6ICR	IOA5ICR	IOB5ICR	IOB4ICR	IOB3ICR	IOB2ICR	IOB1ICR	IOB0ICR
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] **IOAxICR:** A口变化中断使能 (x=5,6)

0 = 屏蔽IOA口变化中断

1 = 使能IOA口变化中断

Bit [5:0] **IOBxICR:** B口变化中断使能 (x=0-5)

0 = 屏蔽IOB口变化中断

1 = 使能IOB口变化中断

5.4 INTCR0 中断控制寄存器 0

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

Bit 7 **TKIE:**

0 = 屏蔽触摸按键中断

1 = 使能触摸按键中断

Bit 6 **ADIE:**

0 = 屏蔽ADC转换中断

1 = 使能ADC转换中断

Bit 1 **TC1IE:**

0 = 屏蔽TC1溢出中断

1 = 使能TC1溢出中断

Bit 0 **TC0IE:**

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	ADIF	-	-	-	-	TC1IF	TC0IF
读/写	R/W	R/W	-	-	-	-	R/W	R/W
复位后	0	0	-	-	-	-	0	0

注：所有中断标志位需软件清零。

- Bit 7 **TKIF:**
 0 = 未产生触摸按键中断
 1 = 产生触摸按键中断
- Bit 6 **ADIF:**
 0 = 未产生ADC转换中断
 1 = 产生ADC转换中断
- Bit 1 **TC1IF:**
 0 = 未产生TC1溢出中断
 1 = 产生TC1溢出中断
- Bit 0 **TC0IF:**
 0 = 未产生TC0溢出中断
 1 = 产生TC0溢出中断

5.6 INTCR1 中断控制寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTCR1	-	-	-	-	INT1IE	INT0IE	-	IOCHIE
读/写	-	-	-	-	R/W	R/W	-	R/W
复位后	-	-	-	-	0	0	-	0

Bit 3 **INT1IE:**

0 = 屏蔽外部中断1

1 = 使能外部中断1

Bit 2 **INT0IE:**

0 = 屏蔽外部中断0

1 = 使能外部中断0

Bit 0 **IOCHIE:**

0 = 屏蔽端口变化中断

1 = 使能端口变化中断

5.7 INTF1 中断标志寄存器 1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INTF1	-	-	-	-	INT1IF	INT0IF	-	IOCHIF
读/写	-	-	-	-	R/W	R/W	-	R/W
复位后	-	-	-	-	0	0	-	0

注：所有中断标志位需软件清零。

Bit 3 **INT1IF:**

0 = 未产生外部中断1

1 = 产生外部中断1

Bit 2 **INT0IF:**

0 = 未产生外部中断0

1 = 产生外部中断0

Bit 0 **IOCHIF:**

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

5.8 中断范例

例:IO 变化中断范例

```

        ORG          0000H
        GOTO         MainProgram
        ORG          0008H
        GOTO         Interrupt

MainProgram:
IO_INTERRUPT_INIT:
; //以 IOB0 口为例
; //1、端口设置
        BSET        PUB, 0           ; //IOB0 端口设置为上拉
        BCLR        OEB, 0          ; //IOB0 端口设置为输入
; //2、允许 B 口变化中断设置
        BSET        IOICR, 0        ; //IOB 口变化中断使能
; //3、中断使能
        BSET        INTCR1, IOCHIE
        BCLR        INTF1, IOCHIF
; //4、总中断使能
        BSET        OPTION, GIE     ; //总中断使能

Main:
        ...
        GOTO        Main            ; //主程序循环

Interrupt:
; //中断进来
        PUSH        ; //压栈、保存 A、STATUS
; //中断处理程序
        JBTS0       INTF1, IOCHIF   ; //检测 IO 中断标志位
        GOTO        Interrupt_IO

Interrupt_End:
; //中断结束
        POP         ; //出栈、恢复 A、STATUS
        RETIE

Interrupt_IO:
        BCLR        INTF1, IOCHIF
; //IO 中断处理程序
        ...
        GOTO        Interrupt_End

        END
  
```

例: INT0 外部中断范例

```

        ORG            0000H
        GOTO           MainProgram
        ORG            0008H
        GOTO           Interrupt

MainProgram:
INT0_Init:
; //1、端口设置
        BSET           PUA, 0           ; //INT0端口设置为上拉
        BCLR           OEA, 0          ; //INT0端口设置为输入
; //2、中断模式选择
        MOVIA          0x34            ; //INT0下降沿中断
        MOVAR          OPTION
; //3、中断使能
        BSET           INTCR1, INT0IE   ; //INT0中断使能
        BCLR           INTF1, INT0IF
; //4、总中断使能
        BSET           OPTION, GIE      ; //总中断使能

Main:
        ...
        GOTO           Main            ; //主程序循环

Interrupt:
; //中断进来
        PUSH           ; //压栈、保存 A、STATUS
; //中断处理程序
        JBTS0          INTF1, INT0IF    ; //检测 INT0 标志位
        GOTO           Interrupt_INT0

Interrupt_End:
; //中断结束
        POP            ; //出栈、恢复 A、STATUS
        RETIE

Interrupt_INT0:
        BCLR           INTF1, INT0IF    ; //清 INT0 标志位
; //INT0 中断处理程序
        ...
        GOTO           Interrupt_End

        END
  
```

6 端口

注：（1）使用IOB5，这个端口的上拉是电阻型的(开漏输出)。
 （2）使用IOB0或IOA3，并设置寄存器IOHDS中的Bit7[LTS1]/Bit6[LTS0]为1，
 则对应端口为非SMT的单一低翻转电平特性。

6.1 IOA

IOA 数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
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

IOA 方向寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEA	OEA7	OEA6	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 [7:0] **OEAx:** A口输出使能 (x=0-7)
 0 = 输入
 1 = 输出

IOA 上拉使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUA	PUA7	PUA6	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	0	0	0	0	0	0	0	0

Bit [7:0] **PUAx:** A口上拉使能 (x=0-7)
 0 = 上拉关闭
 1 = 上拉使能

IOA 端口模式控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSA	ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
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] **ANSAx:** A口模式控制 (x=0-7)
 0 = 作为数字IO口
 1 = 作为模拟端口 (IO输入功能屏蔽)

IOA 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOICR	IOA6ICR	IOA5ICR	IOB5ICR	IOB4ICR	IOB3ICR	IOB2ICR	IOB1ICR	IOB0ICR
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] **IOAxICR:** A口变化中断使能 (x=5,6)

0 = 屏蔽IOA口变化中断

1 = 使能IOA口变化中断

6.2 IOB

IOB 数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOB	-	-	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
读/写	-	-	R/W	R/W	R/W	R/W	R/W	R/W
复位后	-	-	X	X	X	X	X	X

IOB 方向寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OEB	-	-	OEB5	OEB4	OEB3	OEB2	OEB1	OEB0
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] **OEBx:** B口输出使能 (x=0-5)

0 = 输入

1 = 输出

注: IOB[5]作为输出口的注意事项

(1) 需将 PUB[5]置 1 才能输出高电平。

(2) IOB[5]输出的高电平是由上拉电阻提供的, 所以驱动能力弱。

(3) IOB[5]输出的低电平驱动能力比其他端口弱, 输出低电平时内部电路会关闭上拉电阻。

IOB 上拉使能/翻转点选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PUB	-	-	PUB5	PUB4	PUB3	PUB2	PUB1	PUB0
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] **PUBx:** B口上拉使能 (x=0-5)

0 = 上拉关闭

1 = 上拉使能

IOB 端口模式控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANSB	-	-	-	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
读/写	-	-	-	R/W	R/W	R/W	R/W	R/W
复位后	-	-	-	0	0	0	0	0

Bit [4:0] **ANSBx:** B口模式控制 (x=0-4)

0 = 作为数字IO口

1 = 作为模拟端口

IOB 变化中断使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOICR	IOA6ICR	IOA5ICR	IOB5ICR	IOB4ICR	IOB3ICR	IOB2ICR	IOB1ICR	IOB0ICR
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 [5:0] **IOBxICR:** B口变化中断使能 (x=0-5)

0 = 屏蔽IOB口变化中断

1 = 使能IOB口变化中断

6.3 IO 口下拉使能

IOA/IOB 下拉使能寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PDAB	PDA2	PDA1	PDA0	PDB4	PDB3	PDB2	PDB1	PDB0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位后	0	1	0	0	0	0	0	0

Bit [7:5] **PDAx:** A口下拉使能 (x=0-2)

0 = 下拉关闭

1 = 下拉使能

Bit [4:0] **PDBx:** B口下拉使能 (x=0-4)

0 = 下拉关闭

1 = 下拉使能

6.4 IO 驱动选择

IO 弱驱动选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOLDS	A7LDS	A6LDS	A5LDS	A4LDS	B3LDS	B2LDS	B1LDS	B0LDS
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 [7:4] **A[7:4]LDS:** A[7:4]口弱驱动选择使能
 0 = A口驱动为普通驱动 (IOL2, IOH2)
 1 = A口驱动为弱驱动 (IOL1, IOH1)

Bit [3:0] **B[3:0]LDS:** B[3:0]口弱驱动选择使能
 0 = B口驱动为普通驱动 (IOL2, IOH2)
 1 = B口驱动为弱驱动 (IOL1, IOH1)

IO 强驱动选择寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOHDS	LTS1	LTS0	-	A4HDS	-	-	B4HDS	B3HDS
读/写	R/W	R/W	-	R/W	-	-	R/W	R/W
复位后	0	0	-	0	-	-	0	0

Bit [7] **LTS1:** IOA[3]口翻转点选择
 0 = 普通翻转点
 1 = 低翻转点

Bit [6] **LTS0:** IOB[0]口翻转点选择
 0 = 普通翻转点
 1 = 低翻转点

Bit [4] **A4HDS:** IOA[4]口强驱动选择使能
 0 = 驱动为普通驱动 (IOL2, IOH2)
 1 = 驱动为强驱动 (IOL3, IOH3)

Bit [1] **B4HDS:** IOB[4]口强驱动选择使能
 0 = 驱动为普通驱动 (IOL2, IOH2)
 1 = 驱动为强驱动 (IOL4, IOH4)

Bit [0] **B3HDS:** IOB[3]口强驱动选择使能
 0 = 驱动为普通驱动 (IOL2, IOH2)
 1 = 驱动为强驱动 (IOL4, IOH4)

注：IO 口的各个驱动能力参考电气参数表。

7 定时器0/1(TC0/TC1)

7.1 概述

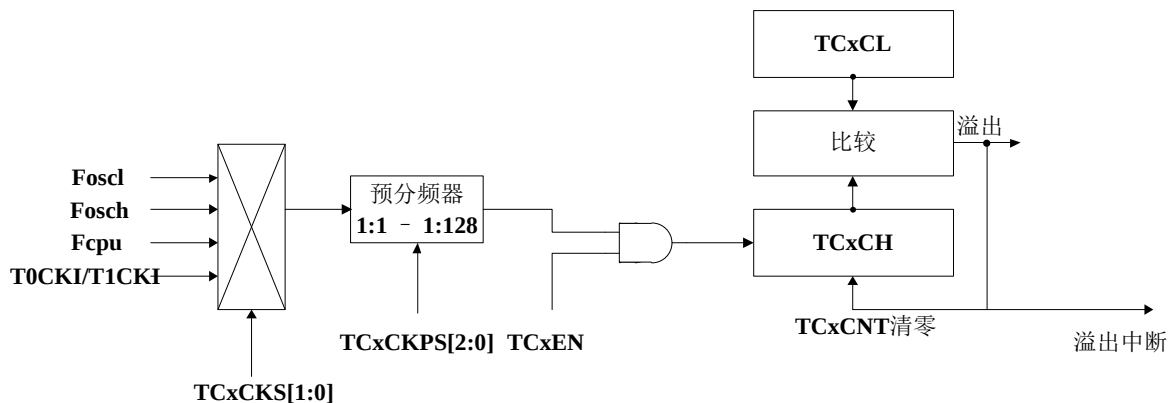
M8P615 TC0/TC1 为带有可设置 1:128 预分频器及周期寄存器的 8 位/16 位定时计数器，具有休眠状态下唤醒功能。

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

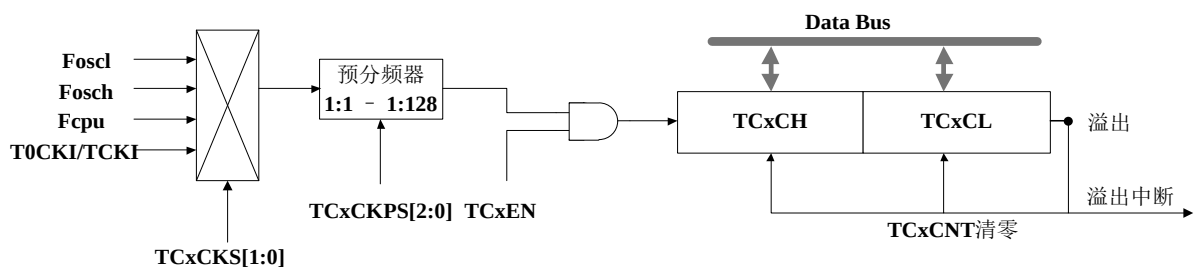
在 16 位模式下，[TCxCH, TCxCL]作为 16 位的计数器，TCx 使能后，16 位计数器递加，当计数值等于 0xFFFF 时，16 位计数器将清零重新开始计数，同时将中断标志位 TCxIF 置 1。

- 可选择时钟源，高频系统时钟 Fosch、低频系统时钟 Foscl、指令时钟 Fcpu 和外部时钟 T0CKI
- 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 TCx 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, Fosch 或 TxCKI 时，若所选择的时钟源振荡器一直工作，此时 TC0/TC1 在休眠状态下依然工作，溢出中断可唤醒 CPU）

8 位模式



16 位模式



7.2 TxCR 控制寄存器 (x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TxCR	TCxEN	TCxMOD	-	TCxCKS1	TCxCKS0	TCxCKPS2	TCxCKPS1	TCxCKPS0
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

Bit 7 **TCxEN:** TCx模块使能位

0 = 关闭TCx

1 = 使能TCx

Bit 6 **TCxMOD:** TCx模式选择位

0 = 8位模式

1 = 16位模式

Bit [4:3] **TCxCKS:** TCx时钟源选择

TCxCKS[1:0]	TCx 时钟源选择
00	Fosc1(低频系统时钟)
01	Fosc1(高频系统时钟)
10	Fcpu
11	T0CKI (TC0) /T1CKI (TC1)

Bit [2:0] **TCxCKPS[2:0]:** TCx预分频比选择

TCxCKPS[2:0]	TCx 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 TCxCL TCx 计数器低 8 位/周期寄存器 (x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCL	TCxCL7	TCxCL6	TCxCL5	TCxCL4	TCxCL3	TCxCL2	TCxCL1	TCxCL0
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

7.4 TCxCH TCx 计数器高位 (x=0,1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCxCH	TCxCH7	TCxCH6	TCxCH5	TCxCH4	TCxCH3	TCxCH2	TCxCH1	TCxCH0
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 定时器范例

```

        ORG          0000H
        GOTO         MainProgram
        ORG          0008H
        GOTO         Interrupt

MainProgram:
TC0_Init:
; //1、时钟源设置(以TC0为例，使其产生125us的波形)
        MOVIA        B'00001100'
        MOVAR        T0CR          ; //定时器T0，选择高频，8位模式，频分比为1: 16
        CLRR         TC0CH
        MOVIA        124
        MOVAR        TC0CL          ; // 0.0625*16*(124+1)=125us
; //2、开启定时器TC0
        BSET         T0CR, 7        ; //开启TC0使能
; //3、开启定时器TC0中断
        BCLR         INTF0, TC0IF   ; //清中断标志位
        BSET         INTCR0, TC0IE  ; //开启定时器TC0溢出中断
        BSET         OPTION, GIE    ; //总中断使能

Main:
        GOTO         Main          ; //主程序循环

Interrupt:
        PUSH                     ; //压栈、保存 A、STATUS
        JBTS0         INTF0, TC0IF ; //判断中断标志位是否置 1
        GOTO         Interrupt_TC0

Interrupt_End:
        POP                     ; //出栈、恢复 A、STATUS
        RETIE

Interrupt_TC0:
        BCLR         INTF0, TC0IF   ; //清中断标志位
; //TC0 中断处理程序
        ...
        GOTO         Interrupt_End
        END
  
```

8 脉宽调制模块PWM0

8.1 概述

M8P615 有 1 路 8 位分辨率的 PWM，时基可选择 TC0 或 TC1。

8.2 PWM0CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0CR	PWM0EN	PWM0OE	-	-	-	PWM0TBS	-	-
读/写	R/W	R/W	-	-	-	R/W	-	-
复位后	0	0	-	-	-	0	-	-

Bit 7 **PWM0EN:** PWM模块使能位

0 = 关闭PWM

1 = 使能PWM

Bit 6 **PWM0OE:** PWM波形输出使能位

0 = 端口用作IO

1 = 端口输出PWM波形

Bit 2 **PWM0TBS:** PWM时基选择

0 = TC0

1 = TC1

8.3 PWM0D 数据寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM0D	PWM0D7	PWM0D6	PWM0D5	PWM0D4	PWM0D3	PWM0D2	PWM0D1	PWM0D0
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

8.4 PWM0 范例

```

        ORG            0000H
        GOTO           MainProgram
        ORG            0008H
        GOTO           Interrupt

MainProgram:
PWM0_Init:
; //1、IO 端口设置
        BSET           OEB, 0                ; //PWM0 的输出端口为 IOB0
; //2、时钟源设置(以 TC0 为例, 使其产生 1KHz 的波形)
        MOVIA          B'00001110'
        MOVAR           T0CR                 ; //TC0, 高频, 8 位模式, 频分比 1:64
        CLRR            TC0CH
        MOVIA           249
        MOVAR           TC0CL                ; //16MHz/64/(249+1)=1KHz
; //3、PWM 设置
        MOVIA           B'01000000'         ; //PWM 时基选择定时器 TC0
        MOVAR           PWM0CR
        MOVIA           (249+1)/2
        MOVAR           PWM0D                ; //产生占空比为 50%的 PWM 波形
; //4、开启 PWM0
        BSET            T0CR, TC0EN          ; //定时器 T0 开启
        BSET            PWM0EN               ; //PWM0 开启

Main:
        ...
        GOTO            Main                 ; //主程序循环

Interrupt:
; //中断进来
        PUSH            ; //压栈、保存 A、STATUS
        中断处理程序
Interrupt_End:
; //中断结束
        POP              ; //出栈、恢复 A、STATUS
        RETIE

        END
  
```

9 脉宽调制模块PWM1

9.1 概述

M8P615 有 1 路带有死区控制的 PWM，分辨率为 8+4 位。

- 8+4 位分辨率模式：设置为 8 位模式的 TCx 做为 PWM 时基，每 16 个 TCx 溢出周期组成一个完整 PWM 周期，4 位扩展位决定相应溢出周期内 PWM 输出波形为(脉宽+1)个计数值，得到等效平均的 12 位 PWM 分辨率效果
- 互补输出
- 死区控制

9.2 PWM1CR 控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1CR	PWM1EN	PWM1POE	PWM1NOE	PWM1PAS	PWM1NAS	PWM1TBS	PWM1S1	PWM1S0
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 **PWM1EN:** PWM模块使能位

0 = 关闭PWM

1 = 使能PWM

Bit 6 **PWM1POE:** PWM正相波形输出使能位

0 = 端口用作IO

1 = 端口输出PWM1P波形

Bit 5 **PWM1NOE:** PWM反相波形输出使能位

0 = 端口用作IO

1 = 端口输出PWM1N波形

Bit 4 **PWM1PAS:** PWM1P波形有效电平选择

0 = 端口输出PWM1P波形有效电平为高电平

1 = 端口输出PWM1P波形有效电平为低电平

Bit 3 **PWM1NAS:** PWM1N波形有效电平选择

0 = 端口输出PWM1N波形有效电平为低电平

1 = 端口输出PWM1N波形有效电平为高电平

Bit 2 **PWM1TBS:** 时基选择

0 = TC0

1 = TC1

Bit [1:0] **PWM1S[1:0]:** PWM输出端口选择

PWM1S[1:0]	PWM 输出端口
00	PWM1P/PWM1N
01	PWM1PX/PWM1NX
10	PWM1PY/PWM1NY
11	保留

9.3 PWM1DH 数据高位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DH	PWM1D11	PWM1D10	PWM1D9	PWM1D8	PWM1D7	PWM1D6	PWM1D5	PWM1D4
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

9.4 PWM1DL 数据低位

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DL	PWM1D3	PWM1D2	PWM1D1	PWM1D0	-	-	-	-
读/写	R/W	R/W	R/W	R/W	-	-	-	-
复位后	0	0	0	0	-	-	-	-

9.5 PWMDEADT PWM1 死区控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDEADT	DEADTF3	DEADTF2	DEADTF1	DEADTF0	DEADTR3	DEADTR2	DEADTR1	DEADTR0
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 [7:4] **DEADTF[3:0]:** 前死区宽度设置

DEADTF[3:0]	前死区时间设定
0000	无前死区
0001	前死区时间为 1*(时基时钟周期/2)
0011	前死区时间为 2*(时基时钟周期/2)
0010	前死区时间为 3*(时基时钟周期/2)
1010	前死区时间为 4*(时基时钟周期/2)
1011	前死区时间为 5*(时基时钟周期/2)
1001	前死区时间为 6*(时基时钟周期/2)
1101	前死区时间为 7*(时基时钟周期/2)
0101	前死区时间为 8*(时基时钟周期/2)
0100	前死区时间为 9*(时基时钟周期/2)
0110	前死区时间为 10*(时基时钟周期/2)
0111	前死区时间为 11*(时基时钟周期/2)
1111	前死区时间为 12*(时基时钟周期/2)
1110	前死区时间为 13*(时基时钟周期/2)
1100	前死区时间为 14*(时基时钟周期/2)
1000	前死区时间为 15*(时基时钟周期/2)

DEADTR[3:0]

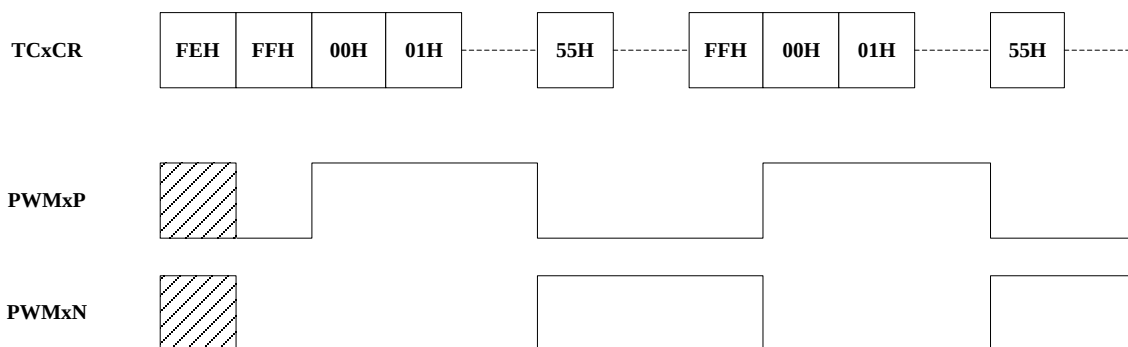
DEADTR[3:0]	后死区时间设定
0000	无后死区
0001	后死区时间为 1*(时基时钟周期/2)
0011	后死区时间为 2*(时基时钟周期/2)
0010	后死区时间为 3*(时基时钟周期/2)
1010	后死区时间为 4*(时基时钟周期/2)
1011	后死区时间为 5*(时基时钟周期/2)
1001	后死区时间为 6*(时基时钟周期/2)
1101	后死区时间为 7*(时基时钟周期/2)
0101	后死区时间为 8*(时基时钟周期/2)
0100	后死区时间为 9*(时基时钟周期/2)
0110	后死区时间为 10*(时基时钟周期/2)
0111	后死区时间为 11*(时基时钟周期/2)
1111	后死区时间为 12*(时基时钟周期/2)
1110	后死区时间为 13*(时基时钟周期/2)
1100	后死区时间为 14*(时基时钟周期/2)
1000	后死区时间为 15*(时基时钟周期/2)

注：时基时钟周期即 PWM 所选择的时钟源经预分频之后的时钟周期。

9.6.1 互补 PWM 输出

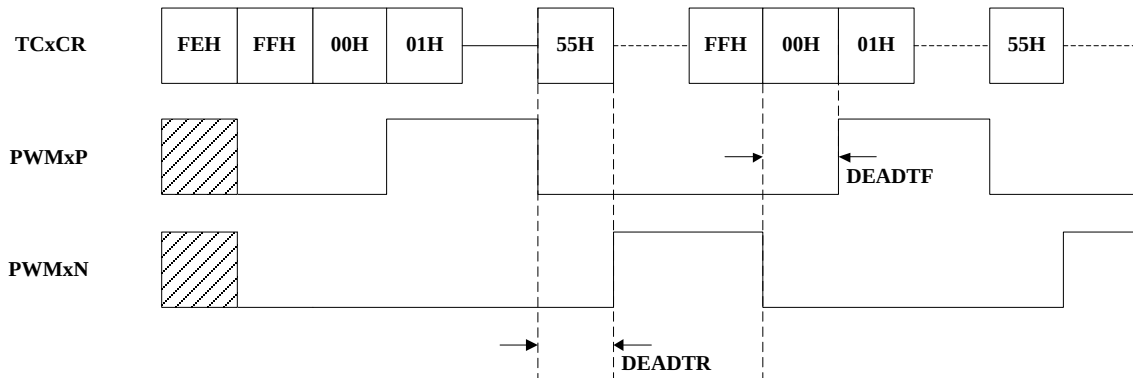
9.6.1 互补 PWM 输出

例: PWM1CR=11100000B, PWM1DH=00H, PWM1DL=55H, TC0CL=FFH



9.6.2 带死区的互补 PWM 输出

例：PWM1CR=11100100B, PWM1DH=55H, PWM1DL=00H, TC0CL=FFH, PWMDEADT=00010001B



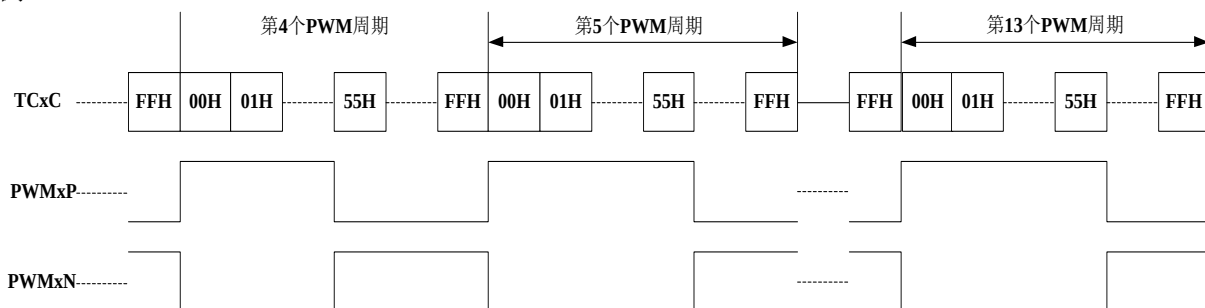
9.6.3 8+4 位分辨率模式

PWM1D[3:0]为4位扩展位, PWMD[11:4]决定 PWM 脉冲基础宽度.在每 16 个 PWM 周期循环中, 扩展位中的有效位对应的 PWM 周期, 输出的 PWM 脉冲宽度为 (PWMD[11:4]+1), 而其余的 PWM 周期, 输出的 PWM 脉冲宽度为 (PWMD[11:4]), 这样得到的 PWM 输出是等效的 12 位 PWM 分辨率效果。

PWM1D[3:0]对应的扩展周期序号:

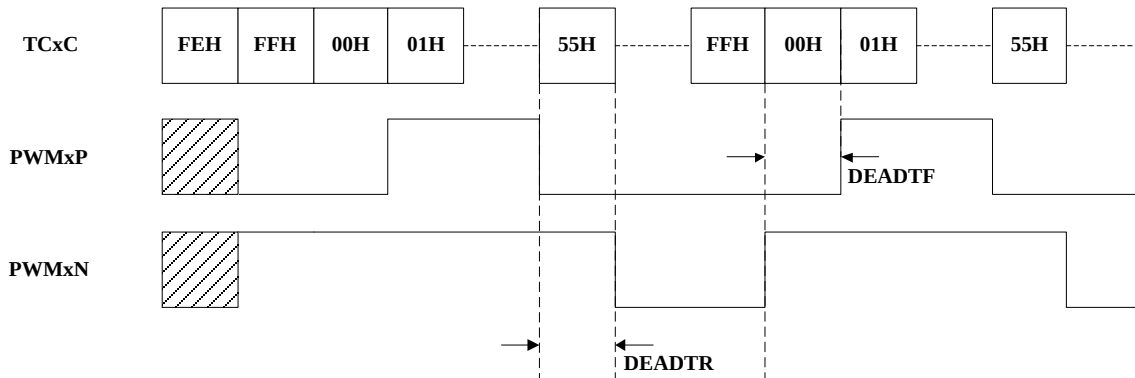
PWM1D[3:0]	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PWM1D3		●		●		●		●		●		●		●		●
PWM1D2			●				●				●				●	
PWM1D1					●								●			
PWM1D0									●							

例：PWM1CR=11100000B, PWM1DH=55H, PWM1DL=02H, TC0CL=FFH



9.6.4 有效电平选择

例：PWM1CR=11101100B, PWM1DH=55H, PWM1DL=00H, TC0CL=FFH, PWMDEADT=00010001B



9.7 PWM1 范例

```

    ORG      0000H
    GOTO     MainProgram
    ORG      0008H
    GOTO     Interrupt

MainProgram:
PWM1_Init:
; //1、IO 端口设置
    BSET     OEB, 3           ; //PWM1N 波形输出口
    BSET     OEB, 4           ; //PWM1P 波形输出口
; //2、时钟源设置(以 TC0 为例，使其产生 1KHz 的波形)
    MOVIA    B'00001110'
    MOVAR    T0CR              ; //TC0, 8 位 Fosch, 分频比 1: 1
    CLRR     TC0CH
    MOVIA    249
    MOVAR    TC0CL              ; //16MHz/64/(249+1)=1KHz
; //3、PWM 设置
    MOVIA    B'01111100'
    MOVAR    PWM1CR              ; //时基 TC0, PWM1P 低电平有效,
PWM1N 高电平有效波形
    MOVIA    0x00
    MOVAR    PWM1DL              ; //8+4 位分辨率模式
    MOVIA    (249+1)/2
    MOVAR    PWM1DH              ; //占空比为 1: 1
    MOVIA    B'00000000'
    MOVAR    PWMDEADT            ; //无前死区和后死区
  
```

```
;;4、开启 PWM1
      BSET      T0CR, TC0EN      ;;定时器 T0 开启
      BSET      PWM1EN          ;;PWM1 开启
Main:
      ...
      GOTO      Main            ;;主程序循环
Interrupt:
;;中断进来
      PUSH                      ;;压栈、保存 A、STATUS
      中断处理程序
Interrupt_End:
;;中断结束
      POP                      ;;出栈、恢复 A、STATUS
      RETIE
      END
```

10 模数转换器(ADC)

10.1 概述

M8P615有一个8路外部通道(AIN0~AIN7)和3路内部通道(VDD/4, VREF, GND)的12位分辨率A/D转换器,可以将模拟信号转换成12位数字信号。进行AD转换时,首先要选择输入通道,然后启动AD转换。转换结束后,系统自动将EOC设置为“1”,并将转换结果存入寄存器ADH和寄存器ADL中。

注: MCU电源口VDD&GND口并联104电容, 104电容位置应紧靠IC, 电源走线也应先进入104电容再进入MCU。

10.2 ADCON0 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADENB	ADS	EOC	ADFM	CHS3	CHS2	CHS1	CHS0
读/写	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
复位后	0	0	1	0	1	0	1	0

- Bit 7 **ADENB:** ADC使能控制位
 0 = 关闭ADC
 1 = 使能ADC
- Bit 6 **ADS:** ADC 启动位
 0 = 停止, 转换完成自动清零
 1 = 开始 (每次写入1将重新启动ADC)
- Bit 5 **EOC:** ADC 状态控制位
 0 = 转换过程中
 1 = 转换结束, ADS 复位
- Bit 4 **ADFM:** 数据格式选择位.
 0 = ADRES = {ADH[7:0], ADL[7:4]}; ADL[3:0] = 0
 1 = ADRES = {ADH[3:0], ADL[7:0]}; ADH[7:4] = 0
- Bit [3:0] **CHS[3:0]:** ADC 输入通道选择位
 [0000] ~ [0111] = AIN0 ~ AIN7
 [1000] = VDD/4
 [1001] = 内建 VREF 基准电平
 [1010] = 未定义
 [1011] = GND

注: (1) 若 ADENB = 1, 用户应设置 IOA.x/AINx 为无上拉电阻的输入模式.系统不会自动设置. 若已经设置了 ANSEL.x, IOA.x/AINx 的数字 I/O 功能。
(2) 休眠模式下, 关掉 ADENB, ADC 输入通道选择位不能为 VDD/4。

10.3 ADCON1 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	-	ADCKS2	ADCKS1	ADCKS0	VREMS1	VREMS0	VHS1	VHS0
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

Bit [6:4] **ADCKS[2:0]:** ADC 时钟源选择位

ADCKS[2:0]	ADC 时钟源选择
000	Fcpu
001	Fcpu/2
010	Fcpu/4
011	Fcpu/8
100	Fcpu/16
101	Fcpu/32
110	Fcpu/64
111	

Bit [3:2] **VREMS[1:0]:** ADC 参考电压模式选择位

VREMS[1:0]	ADC 参考电压模式
00	VDD
01	内部参考电压
10	外部参考电压
11	内部参考与外部参考连接

Bit [1:0] **VHS[1:0]:** ADC 内建基准电平选择位

VHS[1:0]	内建 VREF 基准电平
00	关闭内部参考
01	2.0V
10	3.0V
11	4.0V

- 注：（1）若由 VHS[1:0]控制选择的内部 VREF 电平高于 VDD，内部 VREF 为 VDD。
 例：VHS[1:0] = 11（内部 VREF = 4.0V），VDD = 3.0V，则实际内部 VREF = 3.0V。
 （2）12 位 AD 转换时间 = 16 个 AD 时钟。
 （3）休眠模式下 VHS[1:0]为 00，即关闭内部参考。

10.4 ADCON2 寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	-	-	-	-	ADVOS3	ADVOS2	ADVOS1	ADVOS0
读/写	-	-	-	-	R/W	R/W	R/W	R/W
复位后	-	-	-	-	0	0	0	0

Bit [3:0] **ADVOS[3:0]:** ADC失调补偿寄存器

ADC通道选择内部GND通道，先设置ADCON2为0x00，若ADC的GND通道转换值不为0，就把ADCON2加1，直到ADC的GND通道转换值为0时，ADCON2的值就是调校好的值，ADCON2值最大等于15。

```

Main_Program:                                ;//程序开始
ADC_Init:
; //1、ADCON0 设置
        MOVIA    b'10011011'                ;//选择内部GND通道
        MOVAR    ADCON0
; //2、ADCON1 设置
        MOVIA    b'00000000'
        MOVAR    ADCON1
; //3、ADCON2 设置
        MOVIA    b'00000000'
        MOVAR    ADCON2
; //4、开启转换
ADC_CHANGE:
        BSET     ADCON0, 6
        JBTS0    ADCON0, 6
        GOTO     $-1
        MOVIA    0x00
        JCMPAR   ADH
        GOTO     ADC_CHANGE
        JCMPAR   ADL
        GOTO     $+2
        GOTO     Main                        ;//此时 ADCON2 的值调校好的值
        INCR     ADCON2, R
        GOTO     ADC_CHANGE

Main:
        .....
        GOTO     Main
  
```

10.5 ADH ADC 数据高字节

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

10.6 ADL ADC 数据低字节

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

注：ADH/ADL 的数据格式与 ADM 相关，当 ADFM=1 时，ADH[7:4] =0,ADH[3:0]存放高四位数据
 ADL[7:0]存放低 8 位数据；当 ADFM=0 时，ADH[7:0]存放高 8 位数据，ADL[7:4]存放低 4 位
 数据，AD:[3:0] = 0。

10.7 ADC 范例

例: ADC 模数转换器

将高位的值存在寄存器 **ADC_DATAH**, 低位存在寄存器 **ADC_DATAH**。

```

ADC_DATAH    EQU    '00'                ;//特殊寄存器定义声明
ADC_DATAH    EQU    '01'

                ORG      0000H
                GOTO     MainProgram
                ORG      0008H
                GOTO     Interrupt

MainProgram:
ADC_Init:
; //1、I/O 设置, AIN1 设置为模拟输入
                BCLR     OEA, 1
                BSET     ANSA, 1

; //2、ADCON0 设置---12 位数据左对齐, 通道 1
                MOVIA    0xD1
                MOVAR     ADCON0

; //3、ADCON1 设置---16MHz/16/1, 参考是 VDD
                MOVIA    0x00
                MOVAR     ADCON1

; //4、ADCON2 设置---0 补偿
                MOVIA    0x00
                MOVAR     ADCON2
                NOP
                ...

                NOP                ;//延迟一段时间再采集
                BSET     ADCON0, 7  ;//启动 ADC 转换
                BSET     ADCON0, 6

Main:
                ...

                GOTO     Main        ;//主程序循环

WAIT:
                JBTS1     ADCON0, 5  ;//等待转换完成
                GOTO     WAIT
                MOVR      ADH
                MOVAR     ADC_DATAH  ;//将采集到的高位值送到 ADC_DATAH 寄存器中
                MOVR      ADL
                MOVAR     ADC_DATAH  ;//将采集到的低位值送到 ADC_DATAH 寄存器中
  
```

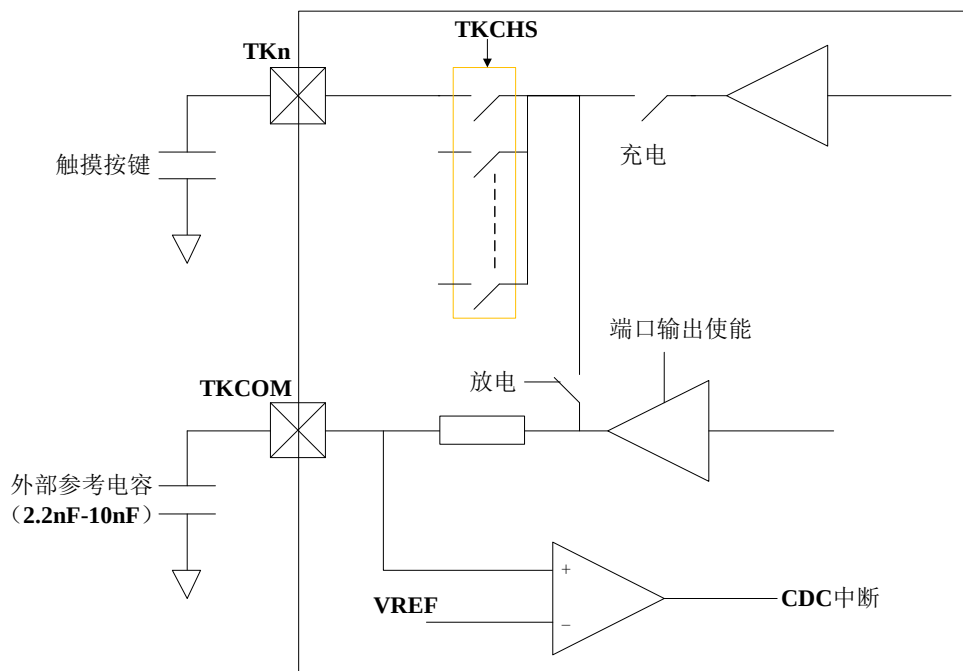
```
Interrupt:
; //中断进来
                PUSH                ; //压栈、保存 A、STATUS
; //中断处理程序
                NOP
Interrupt_End:
; //中断结束
                POP                 ; //出栈、恢复 A、STATUS
                RETIE
                END
```

11 触摸按键（CDC）

11.1 概述

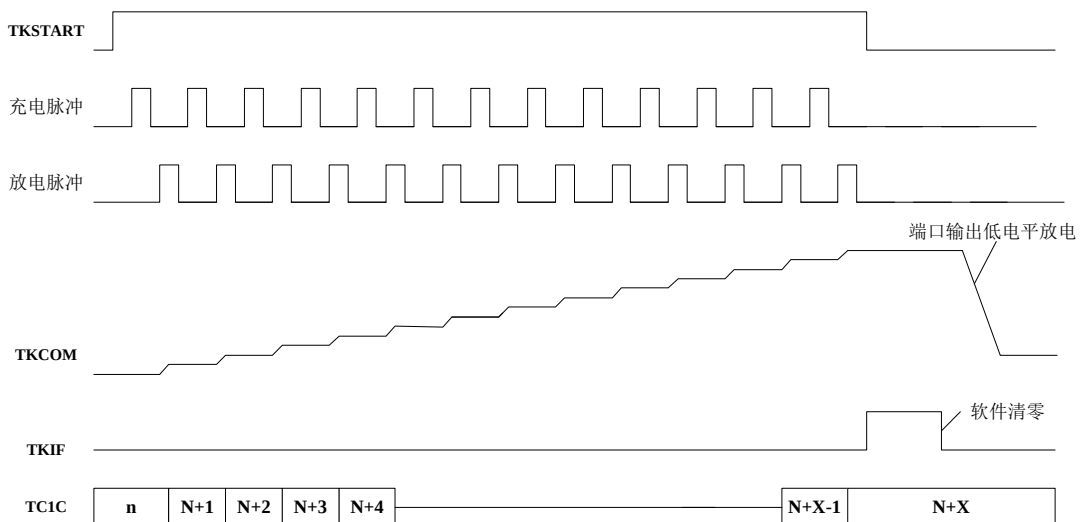
M8P615 有 7 路触摸按键通道，灵敏度可通过外接电容调节，可替代机械式触摸按键，实现防水防尘，简单易用的操作接口。

11.2 原理框图



注：外部参考电容为 NPO 或 X7R 等温度系数较良好的电容。

信号波形示意图:



11.3 TKCTR0 控制寄存器 0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCTR0	TKEN	TKSTART	TKCKS1	TKCKS0	-	TKCHS2	TKCHS1	TKCHS0
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
POR	0	0	0	0	-	0	0	0

Bit 7 **TKEN:** CDC模块使能控制位

0 = 关闭CDC模块

1 = 使能CDC模块

Bit 6 **TKSTART:** 启动通道转换

0 = 通道转换完成, 自动清零

1 = 启动通道转换

Bit [5:4] **TKCKS[1:0]:** CDC时钟分频选择位

TKCKS[1:0]	输入信号选择
00	未定义
01	Fosch/8
10	Fosch/16
11	Fosch/32

Bit [2:0] **TKCHS[2:0]:** CDC通道选择位

TKCHS[2:0]	通道	TKCHS[2:0]	通道
000	0	100	4
001	1	101	5
010	2	110	6
011	3	-	-

11.4 操作说明

- 1- 设置相关通道 IO 方向控制及设置为模拟 PIN
- 2- 定时器 1 时钟源设置为 CDC 输出 TKCLK
- 3- CDC 相关通道/转换时钟设置
- 4- 使能 CDC 模块 TKEN=1
- 5- 设置 TKCOM 管脚输出 0, 对外接电容放电 (保证足够时间放电完全)
- 6- 设置 TKCOM 管脚为输入模式
- 7- 清除定时器 1 TC1CH/TC1CL
- 8- 启动 CDC 转换 (TKSTART 置 1)
- 9- 等待转换完成 (TKSTART=0) /或使用中断模式 (TKIF)
- 10- 读取定时器 1 的计数值, 判断是否有按键发生
- 11- 重复 3-10 对不同通道进行扫描

注: 触摸库函数函数请到公司网站查阅。

12 看门狗 (WDT)

12.1 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) 始终使能：在 STOP 模式下仍然工作，溢出可唤醒 STOP
- (2) STOP 下关闭
- (3) 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。

12.2 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

12.3 WDTC 看门狗控制寄存器

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDTC	WDTC7	WDTC6	WDTC5	WDTC4	WDTC3	WDTC2	WDTC1	WDTC0
读/写	W	W	W	W	W	W	W	W
复位后	-	-	-	-	-	-	-	-

注：(1) WDTC 写入 0x5A 将清除 WDT 定时器，写入其他值无效。
 (2) CLRWDT 指令也可清除 WDT 定时器。

13 芯片配置字

烧录选项	内容	说明
CPU 运行速度选择	2T (LVR $\geq$ 3.8V)	系统时钟频率选择 (不推荐使用 2T)
	4T (LVR $\geq$ 2.4V)	
	8T (LVR $\geq$ 2.2V)	
	16T (LVR $\geq$ 1.8V)	
	32T(LVR $\geq$ 1.4V)	
	64T(LVR $\geq$ 1.4V)	
	128T(LVR $\geq$ 1.4V)	
	256T(LVR $\geq$ 1.4V)	
复位电压选择	LVR=1.4V	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=1.5V	
	LVR=1.6V	
	LVR=1.7V	
	LVR=1.8V	
	LVR=1.9V	
	LVR=2.0V	
	LVR=2.1V	
	LVR=2.2V	
	LVR=2.3V	
	LVR=2.4V	
	LVR=2.5V	
	LVR=3.5V	
	LVR=3.6V	
	LVR=3.7V	
	LVR=3.8V	
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	
输入端口施密特	使能	
	不使能	
输出端口读取	从端口读取	
	从输出寄存器读取	
芯片代码加密	使能	
	不使能	

烧录选项	内容	说明
仿真电压 选择	VDD 4.2V(<200ma)	
	VDD 3.0V(<300ma)	
	外供电源	

14 电性参数

14.1 极限参数

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

14.2 直流特性

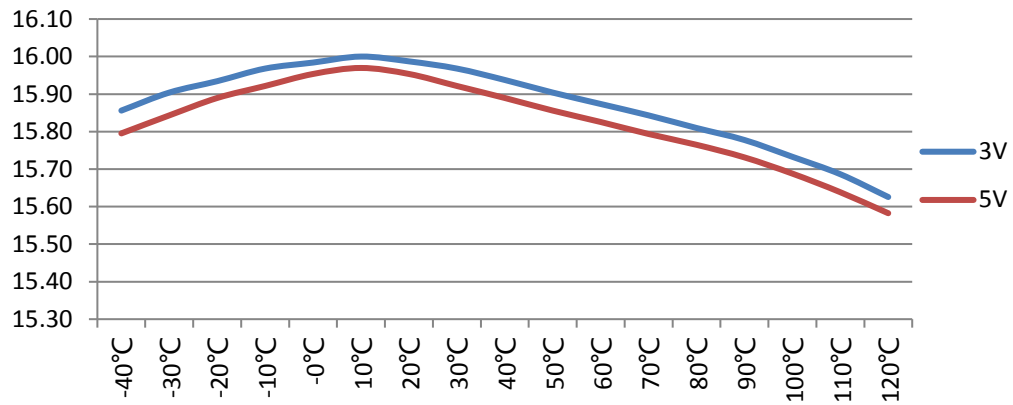
符号	参数	测试条件		最小值	典型值	最大值	单位	
		VDD	条件（常温 25℃）					
VDD	工作电压	—	Fosc = 16MHz， 16T		1.8		5.5	V
IDD1	工作电流 1	3V	高频工作（16M/16T）， 低频工作（64K）， 普通模式			1.60		mA
		5V				2.50		
ISP1	静态电流 1	3V	高频工作（16M/16T）， 低频工作（64K）， 绿色模式			100		uA
		5V				150		
ISP2	静态电流 2	3V	高频停止（16M/16T）， 低频工作（64K）， 绿色模式			2.0		
		5V				6.0		
ISP3	静态电流 3	3V	高频停止（16M/16T）， 低频停止（64K）， 休眠模式			0.6		
		5V				0.9		
ISP4	绿色模式下 WDT 唤醒	3V	高频停止（16M/16T）， 低频工作（64K）， 绿色模式	4.5ms		14.0		uA
				18ms		5.0		
				72ms		3.0		
				288ms		2.0		
		5V		4.5ms		25.0		
				18ms		10.0		
				72ms		7.0		
				288ms		6.0		
VIL1	输入低电平	3V	普通翻转		0		0.7VDD	V
VIH1	输入高电平	3V			0.2VDD		VDD	
VIL1	输入低电平	5V			0		0.7VDD	
VIH1	输入高电平	5V			0.2VDD		VDD	

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℃)				
V _{IL2}	输入低电平	3V	低翻转	0		0.2VDD	V
V _{IH2}	输入高电平	3V		0.3VDD		VDD	
V _{IL2}	输入低电平	5V		0		0.1VDD	
V _{IH2}	输入高电平	5V		0.2VDD		VDD	
R _{PH}	上拉电阻	5V	输入到 GND		25		kΩ
		3V	输入到 GND		45		
R _{PL}	下拉电阻	5V	输入到 VDD		25		
		3V	输入到 VDD		45		
I _{OL1}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		5		mA
		3V			3		
I _{OH1}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		4		
		3V			2		
I _{OL2}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		36		
		3V			24		
I _{OH2}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		20		
		3V			12		
I _{OL3}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		36		
		3V			23		
I _{OH3}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		60		
		3V			30		
I _{OL4}	输出灌电流	5V	输出口, V _{out} =GND+0.6V		210		
		3V			150		
I _{OH4}	输出拉电流	5V	输出口, V _{out} =VDD-0.6V		105		
		3V			60		

14.3 温度对系统时钟的影响

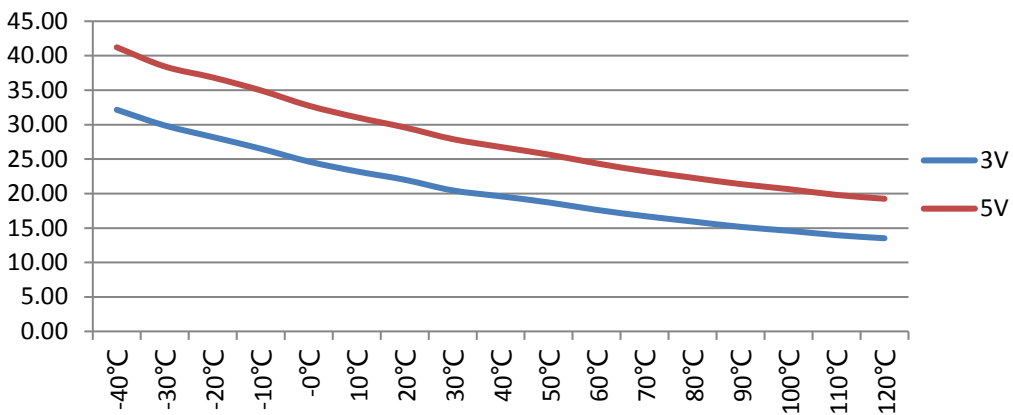
高频(16M)

单位: MHZ



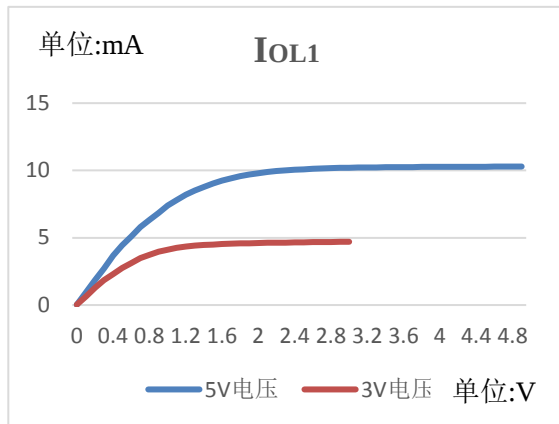
低频(64K)

单位: KHZ

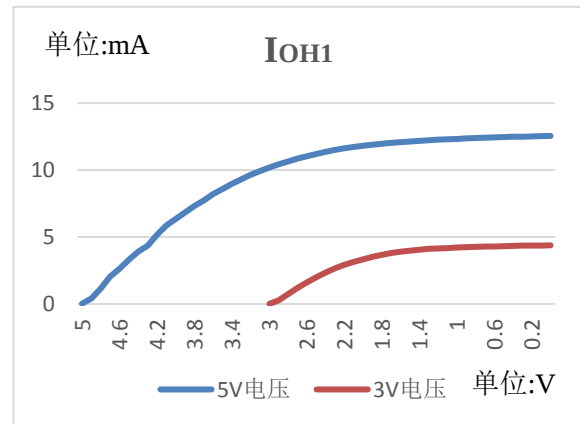


注: 具体值不做设计保证。

14.4 IO 口拉灌电流特性曲线

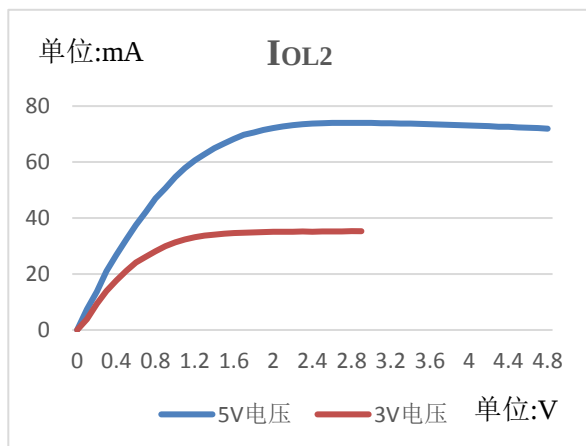


弱驱动模式下灌电流

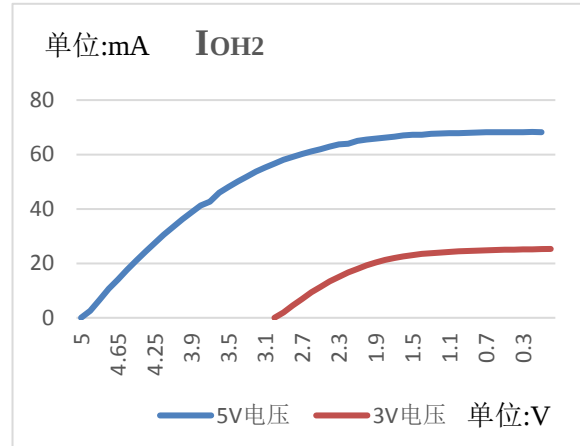


弱驱动模式下拉电流

注：具体值不做设计保证。

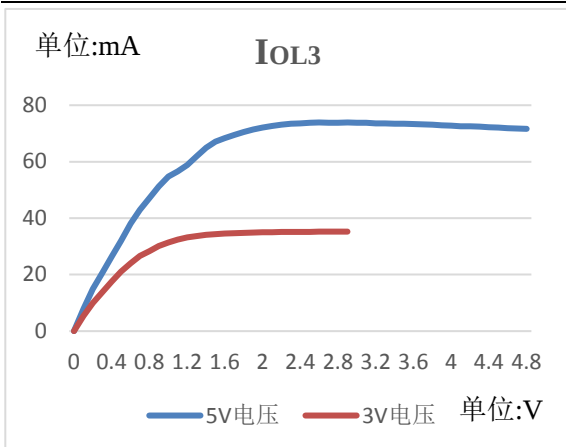


普通驱动模式下灌电流

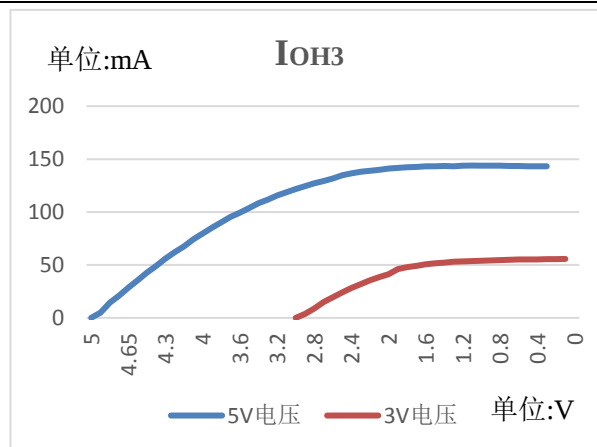


普通驱动模式下拉电流

注：具体值不做设计保证。

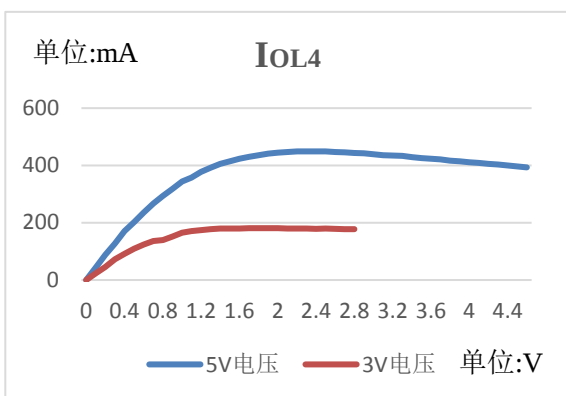


强驱动模式下 IOA4 灌电流

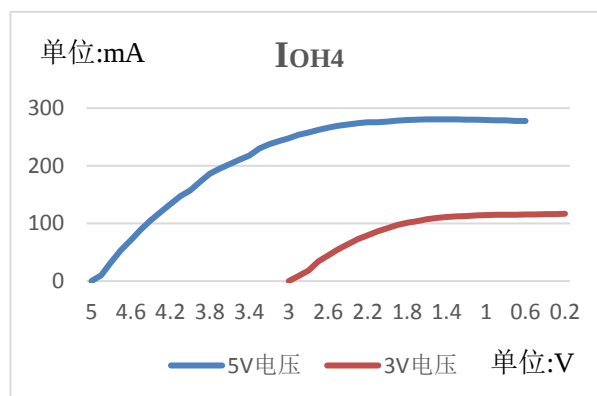


强驱动模式下 IOA4 拉电流

注：具体值不做设计保证。



强驱动模式下 IOB3, 4 灌电流

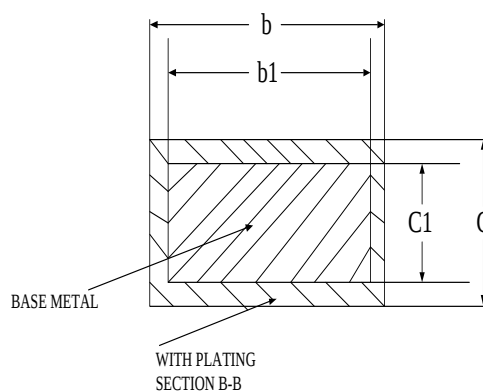
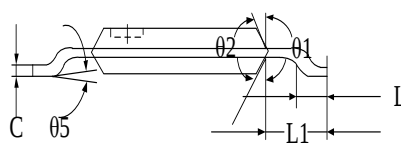


强驱动模式下 IOB3, 4 拉电流

注：具体值不做设计保证。

15.1 SOP16

SOP16 PACKAGE OUTLINE DIMENSIONS

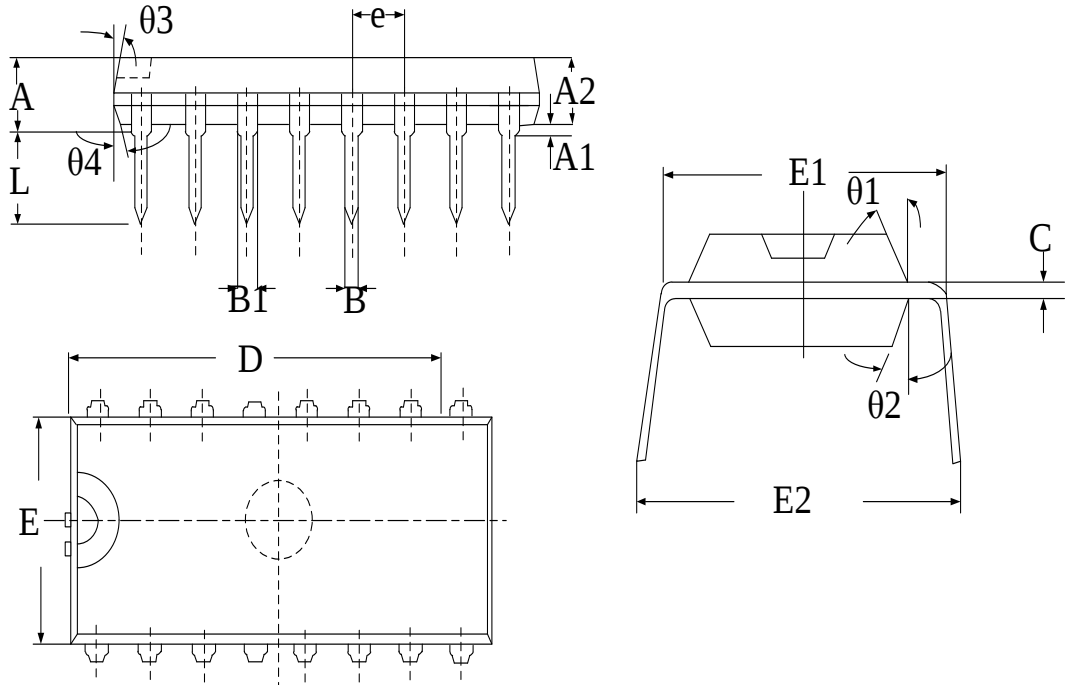


参考标准: JEDEC MS-013, MS-012。

符号	单位 (mm)		
	最小	正常	最大
A	---	---	1.70
A1	0.10	0.15	0.21
A2	1.40	1.45	1.50
A3	0.60	0.65	0.70
b	0.39	---	0.46
b1	0.38	0.41	0.44
c	0.20	---	0.24
c1	0.19	0.20	0.21
D	9.50	9.90	10.05
E	5.80	6.00	6.20
E1	3.75	3.85	4.00
e	1.27 (BSC)		
L	0.50	0.60	0.70
L1	1.10 (REF)		
θ1	6°	---	11°
θ2	6°	---	11°
θ3	5°	---	10°
θ4	6°	---	10°
θ5	0°	---	6°

15.2 DIP16

DIP16 PACKAGE OUTLINE DIMENSIONS



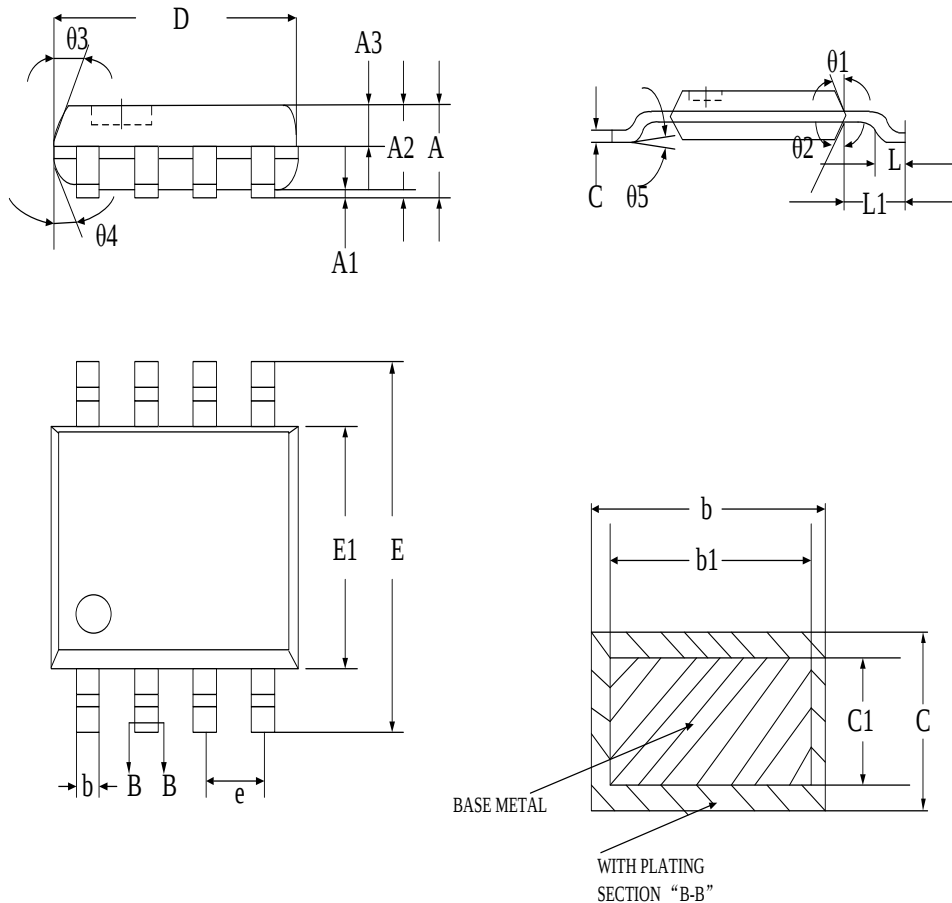
技术说明:

- 1) LEADFRAME MATERIAL: COPPER;
引线框架材料: 铜;
- 2) LEADFRAME THICKNESS: 0.254mm;
引线框架厚度: 0.254mm;
- 3) BOTH PACKAGE LENGTH AND WIDTH DO NOT INCLUDE MOLD FLASH;
塑封体长度及宽度尺寸不包括塑封溢胶;
- 4) REFERENCE: JEDEC MS-013, MS-012。
参考标准: JEDEC MS-013, MS-012。

符号	单位 (mm)		
	最小	正常	最大
A	3.75	3.90	4.05
A1	0.60	---	---
A2	3.20	3.30	3.45
B	0.38	0.48	0.56
B1	1.52 (BSC)		
C	0.20	0.25	0.34
D	18.80	19.05	19.30
E	6.20	6.35	6.50
E1	7.35	7.62	7.85
e	2.54 (BSC)		
L	3.00	3.30	3.60
E2	8.00	8.40	8.80
θ1	9°	---	15°
θ2	7°	---	13°
θ3	8°	---	14°
θ4	5°	---	12°

15.3 SOP8

SOP8(W) PACKAGE OUTLINE DIMENSIONS



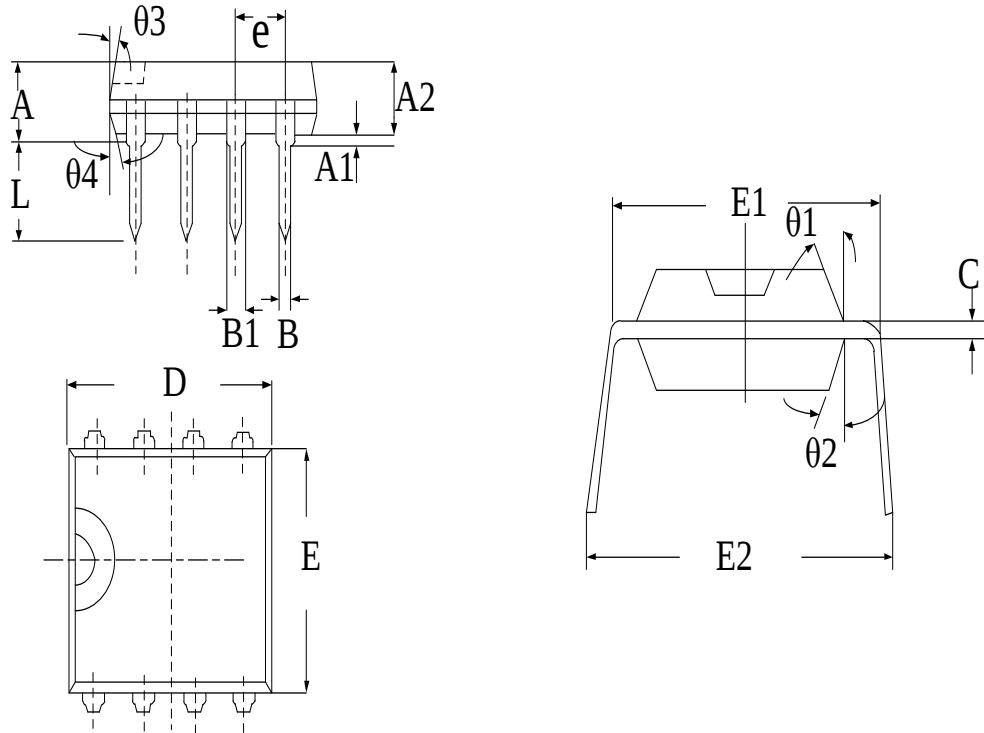
技术说明:

- 1) LEADFRAME MATERIAL: COPPER;
引线框架材料: 铜;
- 2) LEADFRAME THICKNESS: 0.203mm;
引线框架厚度: 0.203mm;
- 3) BOTH PACKAGE LENGTH AND WIDTH DO NOT INCLUDE MOLD FLASH;
塑封体长度及宽度尺寸不包括塑封溢胶;
- 4) REFERENCE: JEDEC MS-013, MS-012.
参考标准: JEDEC MS-013, MS-012。

符号	单位 (mm)		
	最小	正常	最大
A	---	---	1.95
A1	0.10	0.15	0.18
A2	0.75	1.80	1.90
A3	0.78	0.80	0.82
b	0.42	---	0.48
b1	0.40	0.44	0.48
c	0.20	---	0.24
c1	0.19	0.20	0.21
D	5.00	5.17	5.25
E	7.85	7.90	7.98
E1	5.16	5.22	5.26
e	1.27 (BSC)		
L	0.60	0.60	0.70
L1	1.31 (REF)		
θ1	8°	---	12°
θ2	8°	---	12°
θ3	8°	---	12°
θ4	8°	---	12°
θ5	0°	---	6°

15.4 DIP8

DIP8 PACKAGE OUTLINE DIMENSIONS



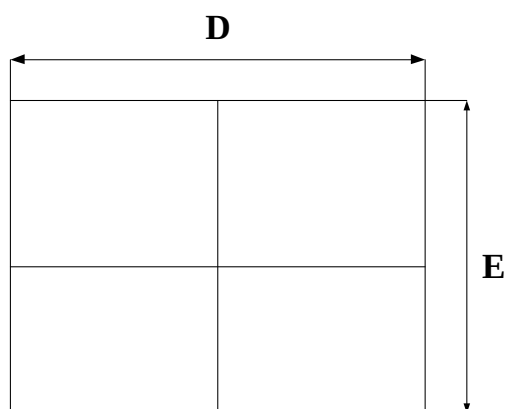
技术说明:

- 1) LEADFRAME MATERIAL: COPPER;
引线框架材料: 铜;
- 2) LEADFRAME THICKNESS: 0.254mm;
引线框架厚度: 0.254mm;
- 3) BOTH PACKAGE LENGTH AND WIDTH DO NOT INCLUDE MOLD FLASH;
塑封体长度及宽度尺寸不包括塑封溢胶;
- 4) REFERENCE: JEDEC MS-013, MS-012。
参考标准: JEDEC MS-013, MS-012。

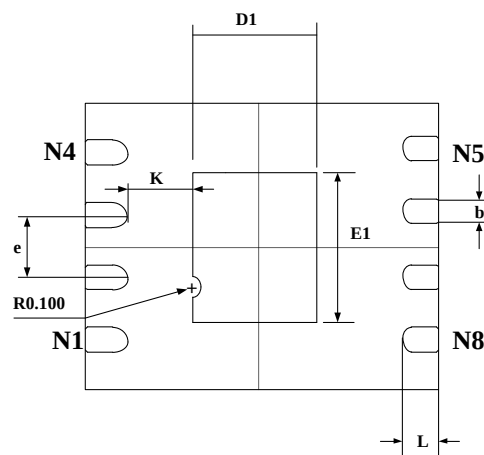
符号	单位 (mm)		
	最小	标准	最大
A	3.75	3.90	4.15
A1	0.60	--	--
A2	3.15	3.30	3.40
B	0.38	0.46	0.56
B1	1.52 (BSC)		
C	0.20	0.25	0.34
D	9.00	9.25	9.40
E	6.20	6.35	6.50
E1	7.35	7.62	7.85
e	2.54 (BSC)		
L	3.00	3.30	3.60
E2	8.00	8.40	8.80
θ1	8°	--	16°
θ2	8°	--	12°
θ3	8°	--	14°
θ4	6°	--	12°

15.5 DFN8

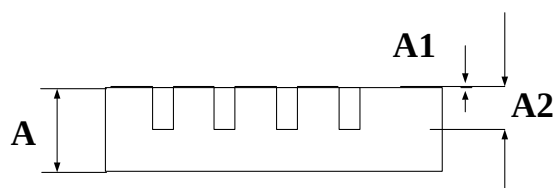
DFNWB2*2-8L-Y(P0.50T0.75) PACKAGE OUTLINE DIMENSIONS



TOP VIEW



BOTTOM VIEW



SIDE VIEW

符号	单位 (mm)			
	最小	最大	标准	最大
A	0.700	0.800	0.032	0.031
A1	0.000	0.050	0.001	0.002
A2	0.203REF		0.008REF	
D	1.900	2.100	0.208	0.083
E	1.900	2.100	0.000	0.083
D1	0.500	0.700	0.075	0.028
E1	1.100	1.300	0.043	0.051
k	0.350REF		0.014REF	
b	0.200	0.300	0.008	0.012
e	0.500REF		0.020REF	
L	0.274	0.426	0.011	0.017

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	$A-R-\overline{C} \rightarrow d$	1	Z,DC,C	
	SUBRA R,d	$R-A \rightarrow d$	1	Z,DC,C	
	SBCRA R,d	$R-A-\overline{C} \rightarrow d$	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	$\overline{R} \rightarrow d$	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	-	
	RRR R,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)	-	
	JCMPar 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	$I+A \rightarrow A$	1	Z,DC,C	
	ADCIA I	$I+A+C \rightarrow A$	1	Z,DC,C	
	SUBIA I	$I-A \rightarrow A$	1	Z,DC,C	
	SBCIA I	$I-A-C \rightarrow A$	1	Z,DC,C	
	SUBAI I	$A-I \rightarrow A$	1	Z,DC,C	
	SBCAI I	$A-I-C \rightarrow A$	1	Z,DC,C	
	ANDIA I	$A \& I \rightarrow A$	1	Z	
	ORIA I	$A I \rightarrow A$	1	Z	
	XORIA I	$A \wedge I \rightarrow A$	1	Z	
	MOVIA I	$I \rightarrow A$	1	-	
	RETIA I	Stack $\rightarrow$ PC, $I \rightarrow A$	2	-	
	JCMPAI I	SKIP if $A=I$	1(2)	Z,C	
	JNCPAI I	SKIP if $A \neq I$	1(2)	Z,C	
特殊操作指令	RLA	$A[7] \rightarrow C, \{A[6:0], C\} \rightarrow A$	1	C	
	RLANC	$\{A[6:0], 0\} \rightarrow A$	1	-	
	RRA	$A[0] \rightarrow C, \{C, A[7:1]\} \rightarrow A$	1	C	
	RRANC	$\{0, A[7:1]\} \rightarrow A$	1	-	
	DECA	$A-1 \rightarrow A$	1	Z	
	DJZA	$A-1 \rightarrow A$, SKIP if 0	1(2)	-	
	INCA	$A+1 \rightarrow A$	1	-	
	JZA	$A+1 \rightarrow A$, SKIP if 0	1(2)	-	
	RETIE	Stack $\rightarrow$ PC, $1 \rightarrow GIE$	2	-	
	RETURN	Stack $\rightarrow$ PC	2	-	
	NOP	None Operation	1	-	
	RDT	ROM[{fsr1,fsr0}] $\rightarrow$ {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	$I \rightarrow PC, PC \rightarrow$ Stack	2	-	
	GOTO I	$I \rightarrow PC$	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 修正记录

版本	日期	描述
V1.00	2017-06-15	初版
V1.01	2017-10-10	勘误
V1.02	2017-12-20	勘误
V1.03	2018-08-15	勘误
V1.04	2018-10-24	勘误
V1.05	2018-12-14	删除 TK7
V1.06	2018-12-27	数据存储器结构说明修改
V1.07	2019-04-23	增加指令集简述
V1.08	2019-09-10	增加模块功能范例
V1.09	2020-06-02	电性参数整理以及增加封装信息
V2.00	2020-11-18	更新电性参数，修改排版，勘误