



超声波雾化器 Flash 单片机

BS45F3833

版本: V1.20 日期: 2022-11-04

www.holtek.com

目录

特性	6
CPU 特性	6
周边特性	6
开发工具	7
概述	7
方框图	8
引脚图	8
引脚描述	9
极限参数	11
直流电气特性	11
交流电气特性	12
4/8/12MHz 内部高速振荡器频率精度	13
PTM 专用内部 RC 振荡器频率精度	14
转换速率控制特性	14
LVD & LVR 电气特性	15
A/D 转换器电气特性	16
过电流 / 过电压保护电气特性	16
上电复位电气特性	17
系统结构	18
时序和流水线结构	18
程序计数器	19
堆栈	19
算术逻辑单元 – ALU	20
Flash 程序存储器	20
结构	20
特殊向量	20
查表	21
查表范例	21
在线烧录	22
片上调试 – OCDS	23
RAM 数据存储器	24
结构	24
通用数据存储器	24
特殊功能数据存储器	24
特殊功能寄存器描述	26
间接寻址寄存器 – IAR0, IAR1	26
存储器指针 – MP0, MP1	26
存储区指针 – BP	26
累加器 – ACC	27

程序计数器低字节寄存器 – PCL	27
查表寄存器 – TBLP, TBHP, TBLH	27
状态寄存器 – STATUS	27
EEPROM 数据存储器	29
EEPROM 数据存储器结构	29
EEPROM 寄存器	29
从 EEPROM 中读取数据	30
写数据到 EEPROM	30
写保护	31
EEPROM 中断	31
编程注意事项	31
振荡器	32
振荡器概述	32
系统时钟配置	32
内部高速 RC 振荡器 – HIRC	33
内部 32kHz 振荡器 – LIRC	33
PTM 专用 RC 振荡器 – PTMOSC	33
工作模式和系统时钟	35
系统时钟	35
系统工作模式	35
控制寄存器	37
工作模式切换	38
待机电流注意事项	42
唤醒	42
看门狗定时器	43
看门狗定时器时钟源	43
看门狗定时器控制寄存器	43
看门狗定时器操作	44
复位和初始化	45
复位功能	45
复位初始条件	47
输入 / 输出端口	50
上拉电阻	51
PA 口唤醒	51
输入 / 输出端口控制寄存器	52
转换速率控制	53
源电流选择	53
引脚共用功能	54
输入 / 输出引脚结构	56
编程注意事项	57
定时器模块 – TM	58
简介	58
TM 操作	58
TM 时钟源	58

TM 中断	59
TM 外部引脚	59
编程注意事项	60
简易型 TM – CTM.....	61
简易型 TM 操作	61
简易型 TM 寄存器介绍	61
简易型 TM 工作模式	65
标准型 TM – STM	71
标准型 TM 操作	71
标准型 TM 寄存器介绍	71
标准型 TM 工作模式	75
周期型 TM – PTM	84
周期型 TM 操作	84
周期型 TM 寄存器介绍	84
周期型 TM 工作模式	88
A/D 转换器.....	97
A/D 转换器简介	97
A/D 转换器寄存器介绍	97
A/D 转换器操作	100
A/D 转换器输入信号	101
A/D 转换率和时序图	102
A/D 转换步骤	102
编程注意事项	103
A/D 转换功能	103
A/D 程序范例	104
雾化器谐振检测器	105
缺水保护	105
过电流 / 电压保护	105
OCVP 操作.....	105
OCVP 控制寄存器.....	106
输入电压范围	110
失调校准	110
触控按键功能	111
触控按键结构	111
触控按键寄存器定义	111
触控按键操作	116
触控按键中断	118
编程注意事项	119
低电压检测 – LVD	119
LVD 寄存器	119
LVD 操作	119
中断	120
中断寄存器	120

中断操作	126
外部中断	127
OCVP 中断.....	127
A/D 转换器中断	127
多功能中断	127
时基中断	127
EEPROM 中断	128
LVD 中断	129
触控按键模块中断	129
TM 中断	129
中断唤醒功能	129
编程注意事项	129
配置选项	130
应用电路	131
指令集	132
简介	132
指令周期	132
数据的传送	132
算术运算	132
逻辑和移位运算	132
分支和控制转换	133
位运算	133
查表运算	133
其它运算	133
指令集概要	134
惯例	134
指令定义	136
封装信息	147
16-pin NSOP (150mil) 外形尺寸	148
20-pin NSOP (150mil) 外形尺寸	149

特性

CPU 特性

- 工作电压：
 - ◆ $f_{SYS}=4\text{MHz}$: 2.2V~5.5V
 - ◆ $f_{SYS}=8\text{MHz}$: 2.2V~5.5V
 - ◆ $f_{SYS}=12\text{MHz}$: 2.7V~5.5V
- $V_{DD}=5\text{V}$, 系统时钟为 12MHz 时, 指令周期为 $0.33\mu\text{s}$
- 提供暂停和唤醒功能, 以降低功耗
- 振荡器类型:
 - ◆ 内部高速 RC – HIRC
 - ◆ 内部低速 32kHz RC – LIRC
 - ◆ PTM 专用 12MHz RC – PTMOSC
- 多种工作模式: 正常、低速、空闲和休眠
- 内部振荡器无需外部元件
- 所有指令都可在 1~2 个指令周期内完成
- 查表指令
- 61 条功能强大的指令
- 4 层堆栈
- 位操作指令

周边特性

- Flash 程序存储器: $2\text{K}\times 16$
- RAM 数据存储器: 128×8
- True EEPROM 存储器: 32×8
- 看门狗定时器功能
- 18 个双向 I/O 口
- 2 个引脚与外部中断口共用
- 转换速率控制用于 PB5 端口输出
- 可编程 I/O 口源电流用于 LED 应用
- 多个定时器模块用于时间测量、捕捉输入、比较匹配输出、PWM 输出及单脉冲输出
 - ◆ 10-bit CTM $\times 3$
 - ◆ 10-bit STM $\times 1$
 - ◆ 10-bit PTM $\times 1$
- 过电流 / 电压保护 (OCVP) 功能
- 双时基功能, 用于产生固定时间的中断信号
- 4 个外部通道的 12-bit A/D 转换器
- 4 个触控按键功能
- 低电压复位功能

- 低电压检测功能
- Flash 程序存储器烧录可达 100,000 次
- Flash 程序存储器数据可保存 10 年以上
- True EEPROM 数据存储器烧录可达 1,000,000 次
- True EEPROM 数据存储器数据可保存 10 年以上
- 封装类型：16/20-pin NSOP

开发工具

为加快产品开发并简化单片机参数设置，Holtek 提供相关开发工具，用户可通过以下链接下载：

https://www.holtek.com.cn/Atomiser_Workshop

概述

该单片机是超声波雾化应用专用的一款单片机。超声波雾化器应用原理为使用电子高频振荡，通过陶瓷雾化片的高频谐振，将液态水分子结构打散从而产生自然飘逸的水雾，不需加热或添加任何化学试剂。若与加热雾化方式比较，能源节省了 90%。此外，在雾化过程中将释放大量的负离子，其与空气中漂浮的烟雾、粉尘等产生静电反应，使其沉淀，同时还能有效去除甲醛，一氧化碳，细菌等有害物质，使空气得到净化，减少疾病的发生。

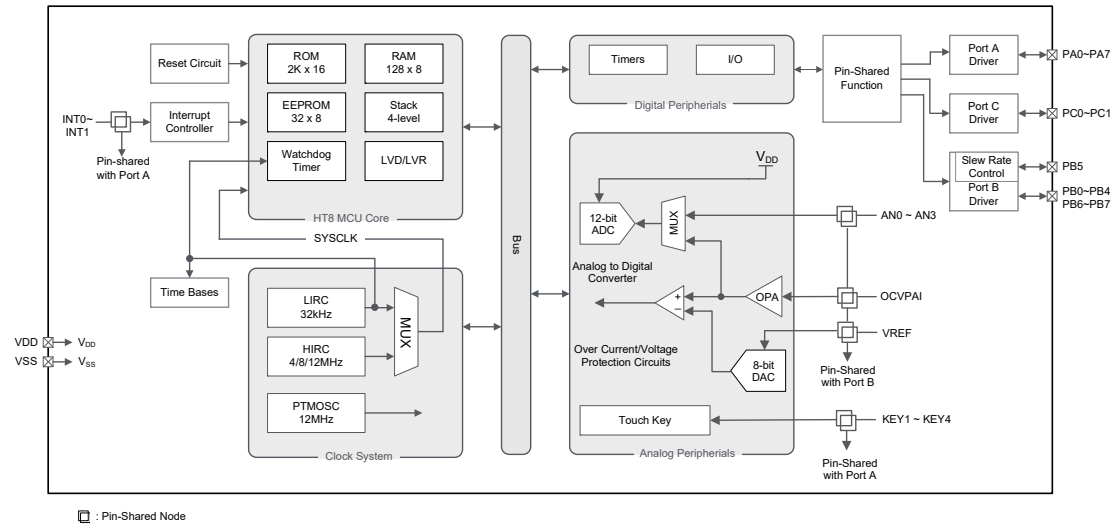
该单片机是一款具有 8-bit 高性能精简指令集的 Flash 单片机，包含专门用于超声波雾化器应用的内部电路。触摸按键功能完全集成于单片机内部，无需外部元件。该单片机具有一系列功能和特性，其 Flash 存储器可多次编程的特性给用户提供了较大的方便。存储器方面，还包含了一个 RAM 数据存储器和一个可用于存储序号、校准数据等非易失性数据的 True EEPROM 存储器。在模拟特性方面，该单片机包含了一个多通道的 12-bit A/D 转换器和一个过电流 / 电压保护功能。多个使用灵活的定时器模块，可提供定时功能、脉冲产生功能及 PWM 产生功能。内部看门狗定时器、低电压复位和低电压检测等内部保护特性，外加优秀的抗干扰和 ESD 保护性能，确保单片机在恶劣的电磁干扰环境下可靠地运行。

此单片机内建了高速和低速振荡器功能选项，可供不同的应用使用。其利用不同的时钟源在不同工作模式之间动态切换的能力，为用户提供了一个优化单片机操作和减少功耗的手段。

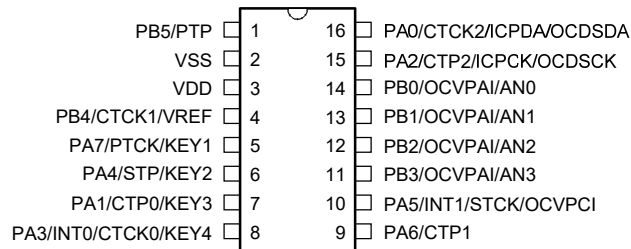
包含 I/O 使用灵活、时基功能和一个可调超声波雾化谐振频率发生器等其它特性，使该单片机可以广泛适用于不同的超声波雾化器应用。

该单片机可使用雾化谐振检测器来检测雾化器谐振频率，并使用雾化谐振频率选择来输出用于雾化器控制的 PFM 谐振频率，缺水保护功能和 OCVF 功能可用来实现缺水检测。

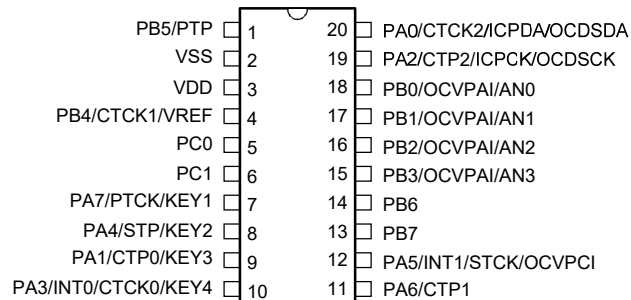
方框图



引脚图



BS45F3833/BS45V3833
16 NSOP-A



BS45F3833/BS45V3833
20 NSOP-A

- 注：1. 在较小封装中可能含有未引出的引脚，需合理设置其状态以避免输入浮空造成额外耗电，详见“待机电流注意事项”和“输入/输出端口”章节。
2. 若共用引脚同时有多种输出，所需引脚共用功能由相应的软件控制位决定。
3. 实际单片机与其对应的 OCDS EV 芯片共用同种封装类型，OCDS EV 芯片为 BS45V3833。OCDSCK 和 OCSDSA 引脚分别与 PA2 和 PA0 引脚共用，仅适用于 OCDS EV 芯片。

引脚描述

下述引脚功能表格是针对最大封装提供的引脚，对于小封装会有部分引脚未出现。

引脚名称	功能	OPT	I/T	O/T	说明
PA0/CTCK2/ ICPDA/OCSDSA	PA0	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTCK2	—	ST	—	CTM2 时钟输入
	ICPDA	—	ST	CMOS	ICP 数据 / 地址
	OCSDSA	—	ST	CMOS	OCDS 数据 / 地址，仅用于 EV 芯片
PA1/CTP0/KEY3	PA1	PAPU PAWU CTRL3	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP0	CTRL3	—	CMOS	CTM0 输出
	KEY3	CTRL3	AN	—	触控按键输入
PA2/CTP2/ ICPCK/OCDSCK	PA2	PAPU PAWU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP2	CTRL3	—	CMOS	CTM2 输出
	ICPCK	—	ST	—	ICP 时钟
	OCDSCK	—	ST	—	OCDS 时钟，仅用于 EV 芯片
PA3/INT0/ CTCK0/KEY4	PA3	PAPU PAWU CTRL5	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT0	CTRL5	ST	—	外部中断 0
	CTCK0	CTRL5	ST	—	CTM0 时钟输入
	KEY4	CTRL5	AN	—	触控按键输入
PA4/STP/KEY2	PA4	PAPU PAWU CTRL3	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	STP	CTRL3	ST	CMOS	STM 输出或捕捉输入
	KEY2	CTRL3	AN	—	触控按键输入
PA5/INT1/STCK/ OCVPCI	PA5	PAPU PAWU CTRL3	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	INT1	CTRL3 INTEG	ST	—	外部中断 1
	STCK	CTRL3	ST	—	STM 时钟输入
	OCVPCI	CTRL3	AN	—	OCVP 比较器同相信号输入
PA6/CTP1	PA6	PAPU PAWU CTRL3	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	CTP1	CTRL3	—	CMOS	CTM1 输出

引脚名称	功能	OPT	I/T	O/T	说明
PA7/PTCK/KEY1	PA7	PAPU PAWU CTRL3	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻和唤醒功能
	PTCK	CTRL3	ST	—	PTM 时钟输入或捕捉输入
	KEY1	CTRL3	AN	—	触控按键输入
PB0/OCVPAI0/ AN0	PB0	PBPU CTRL4	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OCVPAI0	CTRL2 CTRL4	AN	—	OCVPAI0 输入通道
	AN0	ACERL	AN	—	A/D 转换器模拟输入通道 0
PB1/OCVPAI0/ AN1	PB1	PBPU CTRL4	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OCVPAI0	CTRL2 CTRL4	AN	—	OCVPAI0 输入通道
	AN1	ACERL	AN	—	A/D 转换器模拟输入通道 1
PB2/OCVPAI0/ AN2	PB2	PBPU CTRL4	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OCVPAI0	CTRL2 CTRL4	AN	—	OCVPAI0 输入通道
	AN2	ACERL	AN	—	A/D 转换器模拟输入通道 2
PB3/OCVPAI0/ AN3	PB3	PBPU CTRL4	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	OCVPAI0	CTRL2 CTRL4	AN	—	OCVPAI0 输入通道
	AN3	ACERL	AN	—	A/D 转换器模拟输入通道 3
PB4/CTCK1/ VREF	PB4	PBPU CTRL5	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	CTCK1	CTRL5	ST	—	CTM1 时钟输入
	VREF	CTRL5	AN	—	OCVP D/A 转换器参考电压输入
PB5/PTP	PB5	PBPU CTRL5	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
	PTP	CTRL5	ST	CMOS	PTM 时钟输入或捕捉输入
PB6~PB7	PB6~PB7	PBPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
PC0~PC1	PC0~PC1	PCPU	ST	CMOS	通用 I/O 口，可通过寄存器设置上拉电阻
VDD	VDD	—	PWR	—	正电源
VSS	VSS	—	PWR	—	负电源，接地

注：I/T：输入类型； O/T：输出类型；
 OPT：通过寄存器选项来配置；
 PWR：电源； ST：施密特触发输入；
 CMOS：CMOS 输出； AN：模拟信号。

极限参数

电源供应电压	$V_{SS}-0.3V \sim V_{SS}+6.0V$
输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3V$
储存温度	$-60^{\circ}C \sim 150^{\circ}C$
工作温度	$-40^{\circ}C \sim 85^{\circ}C$
I_{OL} 总电流	80mA
I_{OH} 总电流	-80mA
总功耗	500mW

注：这里只强调额定功率，超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述标示范围外的工作状态，而且若长期在标示范围外的条件下工作，可能影响芯片的可靠性。

直流电气特性

$T_a = -40^{\circ}C \sim 85^{\circ}C$

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DD}	工作电压 (HIRC)	—	$f_{SYS}=f_{HIRC}=4MHz$	2.2	—	5.5	V
			$f_{SYS}=f_{HIRC}=8MHz$	2.2	—	5.5	
			$f_{SYS}=f_{HIRC}=12MHz$	2.7	—	5.5	
	工作电压 (LIRC)	—	$f_{SYS}=f_{LIRC}=32kHz$	2.2	—	5.5	V
I_{DD}	工作电流 (HIRC)	3V	无负载，所有外设除能，	—	2.2	3.3	mA
		5V	$f_{SYS}=f_{HIRC}=4MHz$	—	5	7.5	
		3V	无负载，所有外设除能，	—	2.2	3.3	
		5V	$f_{SYS}=f_{HIRC}=8MHz$	—	5	7.5	
		3V	无负载，所有外设除能，	—	2.2	3.3	
		5V	$f_{SYS}=f_{HIRC}=12MHz$	—	5	7.5	
	工作电流 (LIRC)	3V	无负载，所有外设除能，	—	10	20	μA
		5V	$f_{SYS}=f_{LIRC}=32kHz$	—	30	50	
I_{STB}	待机电流 (休眠模式 0)	3V	无负载，所有外设除能，	—	0.2	0.8	μA
		5V	WDT 除能	—	0.5	1	
	待机电流 (休眠模式 1)	3V	无负载，所有外设除能，	—	1.3	5	μA
		5V	WDT 使能	—	2.2	10	
	待机电流 (空闲模式 0)	3V	无负载，所有外设除能，	—	1.3	3	μA
		5V	f_{SUB} 使能	—	2.2	5	
	待机电流 (空闲模式 1, HIRC)	3V	无负载，所有外设除能，	—	0.6	1.2	mA
		5V	f_{SUB} 使能， $f_{SYS}=f_{HIRC}=12MHz$	—	1.2	2.4	
V_{IL}	I/O 口低电平输入电压	5V	—	0	—	1.5	V
		—	—	0	—	$0.2V_{DD}$	
V_{IH}	I/O 口高电平输入电压	5V	—	3.5	—	5	V
		—	—	$0.8V_{DD}$	—	V_{DD}	

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{OL}	I/O 口灌电流 (除 PB5 端口外)	3V	V _{OL} =0.1 V _{DD}	16	32	—	mA
		5V		32	65	—	
I _{OH}	I/O 口源电流 (除 PB5 端口外)	3V	V _{OH} =0.9V _{DD} SLEDCn[m+1,m]=00B n=0, 1; m=0, 2, 4, 6	-0.7	-1.5	—	mA
		5V		-1.5	-2.9	—	
		3V	V _{OH} =0.9V _{DD} SLEDCn[m+1,m]=01B n=0, 1; m=0, 2, 4, 6	-1.3	-2.5	—	mA
		5V		-2.5	-5.1	—	
		3V	V _{OH} =0.9V _{DD} SLEDCn[m+1,m]=10B n=0, 1; m=0, 2, 4, 6	-1.8	-3.6	—	mA
		5V		-3.6	-7.3	—	
		3V	V _{OH} =0.9V _{DD} SLEDCn[m+1,m]=11B n=0, 1; m=0, 2, 4, 6	-4	-8	—	mA
		5V		-8	-16	—	
R _{PH}	I/O 口上拉电阻	3V	—	20	60	100	kΩ
		5V		10	30	50	

交流电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
f _{SYS}	系统时钟 (HIRC)	2.2V~5.5V	f _{SYS} =f _{HIRC} =4MHz	—	4	—	MHz
		2.2V~5.5V	f _{SYS} =f _{HIRC} =8MHz	—	8	—	
		2.7V~5.5V	f _{SYS} =f _{HIRC} =12MHz	—	12	—	
	系统时钟 (LIRC)	2.2V~5.5V	f _{SYS} =f _{LIRC} =32kHz	—	32	—	kHz
f _{LIRC}	低速内部 RC 振荡器 (LIRC)	3V	Ta=25°C	-10%	32	+10%	kHz
		5V	Ta=25°C	-10%	32	+10%	
		3V±0.3V	Ta=-40°C~85°C	-40%	32	+40%	
		5V±0.5V	Ta=-40°C~85°C	-40%	32	+40%	
		2.2V~5.5V	Ta=-40°C~85°C	-50%	32	+60%	
t _{TIMER}	xTCKn 输入引脚最小脉宽	—	—	—	150	—	ns
t _{DEW}	数据 EEPROM 写入时间	—	—	—	2	4	ms
t _{INT}	外部中断最小脉宽	—	—	10	—	—	μs
t _{RSTD}	系统复位延迟时间 (上电复位, LVR 硬件复位, LVRC 软件复位, WDTC 软件复位)	—	—	25	50	100	ms
	系统复位延迟时间 (WDT 溢出硬件冷复位)	—	—	8.3	16.7	33.3	ms

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
t _{SST}	系统启动时间 (从 f _{SYS} off 的状态下唤醒)	—	f _{SYS} =f _H ~f _H /64, f _H =f _{HIRC}	16	—	—	t _{HIRC}
		—	f _{SYS} =f _{SUB} =f _{LIRC}	2	—	—	t _{LIRC}
	系统启动时间 (低速模式 ↔ 正常模式)	—	f _{HIRC} off → on (HTO=1)	16	—	—	t _{HIRC}
		—	f _{SYS} =f _{HIRC} ~f _{HIRC} /64	2	—	—	t _H
	系统启动时间 (从 f _{SYS} on 的状态下唤醒)	—	f _{SYS} =f _{LIRC}	2	—	—	t _{SUB}
		—	—	0	—	—	t _H
t _{SRESET}	最小软件复位时间	—	—	45	90	120	μs

注: t_{SYS}=1/f_{SYS}; t_{SUB}=1/f_{SUB}; t_{HIRC}=1/f_{HIRC}。

4/8/12MHz 内部高速振荡器频率精度

程序烧录时, 烧录器可调整 HIRC 振荡器使其工作在用户选择的 HIRC 频率和工作电压 (3V 或 5V) 条件下。

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	温度				
f _{HIRC}	通过烧录器调整后的 4MHz HIRC 频率	3V/5V	25°C	-1%	4	+1%	MHz
			-40°C~85°C	-2%	4	+2%	
		2.2V~5.5V	25°C	-2.5%	4	+2.5%	
			-40°C~85°C	-3%	4	+3%	
	通过烧录器调整后的 8MHz HIRC 频率	3V/5V	25°C	-1%	8	+1%	
			-40°C~85°C	-2%	8	+2%	
		2.2V~5.5V	25°C	-2.5%	8	+2.5%	
			-40°C~85°C	-3%	8	+3%	
	通过烧录器调整后的 12MHz HIRC 频率	5V	25°C	-1%	12	+1%	
			-40°C~85°C	-2%	12	+2%	
		2.7V~5.5V	25°C	-2.5%	12	+2.5%	
			-40°C~85°C	-3%	12	+3%	

注: 1. 烧录器可在 3V/5V 这两个可选的固定电压下对 HIRC 频率进行调整, 在此提供 V_{DD}=3V/5V 时的参数值。

2. 3V/5V 表格列下面提供的是全压条件下的参数值。对于电压范围在 2.2V~3.6V 的应用, 建议烧录器电压固定在 3V; 对于电压范围在 3.3V~5.5V 的应用, 建议烧录器电压固定在 5V。

PTM 专用内部 RC 振荡器频率精度

该 12MHz 内部 RC 振荡器专门用来提供时钟给 PTM。烧录器可在 V_{DD} 为 3V/5V 的条件下对 12MHz 振荡器频率进行调整，FADJH 值为 01H，FADJL 值为 00H。

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	温度				
f_{PTM}	通过烧录器调整后的 12MHz HIRC 频率	3V	Ta=25°C	-2%	12	+2%	MHz
			Ta=0°C~70°C	-7%	12	+7%	
			Ta=-40°C~85°C	-10%	12	+10%	
		5V	Ta=25°C	-2%	12	+2%	
			Ta=0°C~70°C	-7%	12	+7%	
			Ta=-40°C~85°C	-7%	12	+7%	
		2.7V~5.5V	Ta=25°C	-5%	12	+5%	
			Ta=0°C~70°C	-7%	12	+7%	
			Ta=-40°C~85°C	-10%	12	+10%	

转换速率控制特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
I_{OL}	带转换速率控制的 I/O 端口灌电流 (PB5 口)	3V	$V_{OL}=0.2V_{DD}$	24	60	—	mA
		5V		60	150	—	
I_{OH}	带转换速率控制的 I/O 端口源电流 (PB5 口)	3V	$V_{OH}=0.8V_{DD}$	-24	-60	—	mA
		5V		-60	-150	—	
SR_{RISE}	I/O 口输出上升沿转换速率 (PB5 口)	5V	SLEWC0[1:0]=00B 0.5V→4.5V, $C_{LOAD}=1000pF$	200	—	—	V/ μs
		5V	SLEWC0[1:0]=01B 0.5V→4.5V, $C_{LOAD}=1000pF$	—	120	—	
		5V	SLEWC0[1:0]=10B 0.5V→4.5V, $C_{LOAD}=1000pF$	—	60	—	
		5V	SLEWC0[1:0]=11B 0.5V→4.5V, $C_{LOAD}=1000pF$	—	45	—	

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
SR _{FALL}	I/O 口输出下降沿转换速率 (PB5 口)	5V	SLEWC0[1:0]=00B 4.5V→0.5V, C _{LOAD} =1000pF	200	—	—	V/μs
		5V	SLEWC0[1:0]=01B 4.5V→0.5V, C _{LOAD} =1000pF	—	120	—	
		5V	SLEWC0[1:0]=10B 4.5V→0.5V, C _{LOAD} =1000pF	—	60	—	
		5V	SLEWC0[1:0]=11B 4.5V→0.5V, C _{LOAD} =1000pF	—	45	—	

LVD & LVR 电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{LVR}	低电压复位电压	—	LVR 使能, 电压选择 2.1V	-5%	2.1	+5%	V
			LVR 使能, 电压选择 2.55V		2.55		
			LVR 使能, 电压选择 3.15V		3.15		
			LVR 使能, 电压选择 3.8V		3.8		
V _{LVD}	低电压检测电压	—	LVDEN=1, V _{LVD} =2.0V	-5%	2.0	+5%	V
		—	LVDEN=1, V _{LVD} =2.2V		2.2		
		—	LVDEN=1, V _{LVD} =2.4V		2.4		
		—	LVDEN=1, V _{LVD} =2.7V		2.7		
		—	LVDEN=1, V _{LVD} =3.0V		3.0		
		—	LVDEN=1, V _{LVD} =3.3V		3.3		
		—	LVDEN=1, V _{LVD} =3.6V		3.6		
		—	LVDEN=1, V _{LVD} =4.0V		4.0		
I _{LVR}	LVR 使能的额外电流	5V±3%	LVR 除能 → LVR 使能	—	60	90	μA
I _{LVD}	LVD 使能的额外电流	5V±3%	LVD 除能 → LVD 使能 (LVR 除能)	—	75	115	μA
			LVD 除能 → LVD 使能 (LVR 使能)	—	60	90	
t _{LVR}	产生 LVR 复位的低电压最短保持时间	—	—	120	240	480	μs
t _{LVD}	产生 LVD 中断的低电压最短保持时间	—	—	60	120	240	μs
t _{LVDS}	LVDO 稳定时间	—	LVR 使能, LVD off → on	—	—	15	μs
		—	LVR 除能, LVD off → on	—	—	150	

注：V_{LVR} 或 V_{LVD} 是指发生 LVR 复位或 LVD 中断的电压等级。

A/D 转换器电气特性

Ta=-40°C~85°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{DD}	工作电压	—	—	2.7	5	5.5	V
V _{ADI}	输入电压	—	—	0	—	V _{REF}	V
V _{REF}	参考电压	—	—	2.0	—	V _{DD}	V
DNL	非线性微分误差	3V	V _{REF} =V _{DD} , t _{ADCK} =0.5μs 或 10μs	-3	—	+5	LSB
		5V					
INL	非线性积分误差	3V	V _{REF} =V _{DD} , t _{ADCK} =0.5μs 或 10μs	-5	—	+6	LSB
		5V					
I _{ADC}	A/D 转换器使能的额外电流	3V	无负载, t _{ADCK} =0.5μs	—	0.9	1.35	mA
		5V		—	1.2	1.8	
t _{ADCK}	A/D 转换器时钟周期	—	—	0.5	—	10	μs
t _{ON2ST}	A/D 转换器 On-to-Start 时间	—	—	4	—	—	μs
t _{ADC}	A/D 转换时间 (包括 A/D 采样和保持时间)	—	12-bit ADC	—	16	—	t _{ADCK}

过电流 / 过电压保护电气特性

Ta=-40°C~85°C

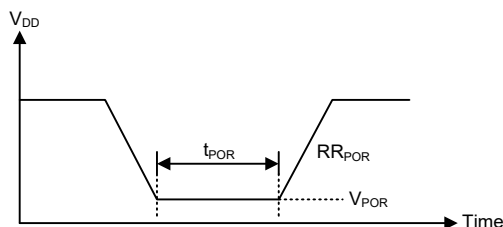
符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
I _{OCVP}	工作电流	3V	DAC V _{REF} =2.5V	—	—	1250	μA
		5V		—	730	1250	
V _{OS_CMP}	比较器输入失调电压	3V	无校准 (OCVPCOF[4:0]=10000B)	-15	—	15	mV
		5V		-15	—	15	
		3V	校准后	-4	—	4	mV
		5V		-4	—	4	
V _{HYS}	迟滞	3V	—	20	40	60	mV
		5V	—	20	40	60	
V _{CM_CMP}	比较器共模电压范围	3V	—	V _{SS}	—	V _{DD} - 1.4	V
		5V	—	V _{SS}	—	V _{DD} - 1.4	
V _{OS_OPA}	OPAMP 输入失调电压	3V	无校准 (OCVPOOF[5:0]=100000B)	-15	—	15	mV
		5V		-15	—	15	
		3V	校准后	-4	—	4	mV
		5V		-4	—	4	

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{CM_OPA}	OPAMP 共模电压范围	3V	—	V _{SS}	—	V _{DD} - 1.4	V
		5V	—	V _{SS}	—	V _{DD} - 1.4	
V _{OR}	OPAMP 最大输出电压范围	3V	—	V _{SS} + 0.1	—	V _{DD} - 0.1	V
		5V	—	V _{SS} + 0.1	—	V _{DD} - 0.1	
Ga	PGA 增益精度	3V/5V	同相模式中, R2/R1 ≤ 50, 使用内部电阻, 且输入电压 > 80mV	-5	—	5	%
			同相模式中, R2/R1=65 或 80, 使用内部电阻, 且输入电压 > 50mV	-8	—	8	%
			同相模式中, R2/R1=130, 使用内部电阻, 且输入电压 > 35mV	-10	—	10	%
			反相模式中, R2/R1 ≤ 50, 使用内部电阻, 且 -300mV < 输入电压 < -80mV	-5	—	5	%
			反相模式中, R2/R1=65 或 80, 使用内部电阻, 且 -300mV < 输入电压 < -50mV	-8	—	8	%
			反相模式中, R2/R1=130, 使用内部电阻, 且 -300mV < 输入电压 < -35mV	-10	—	10	%
DNL	非线性微分误差	3V	DAC V _{REF} =V _{DD}	—	—	±2	LSB
		5V		—	—	±1	LSB
INL	非线性积分误差	3V	DAC V _{REF} =V _{DD}	—	—	±2	LSB
		5V		—	—	±1.5	LSB

上电复位电气特性

Ta=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{POR}	上电复位电压	—	—	—	—	100	mV
RR _{POR}	上电复位电压速率	—	—	0.035	—	—	V/ms
t _{POR}	V _{DD} 保持为 V _{POR} 的最小时间	—	—	1	—	—	ms



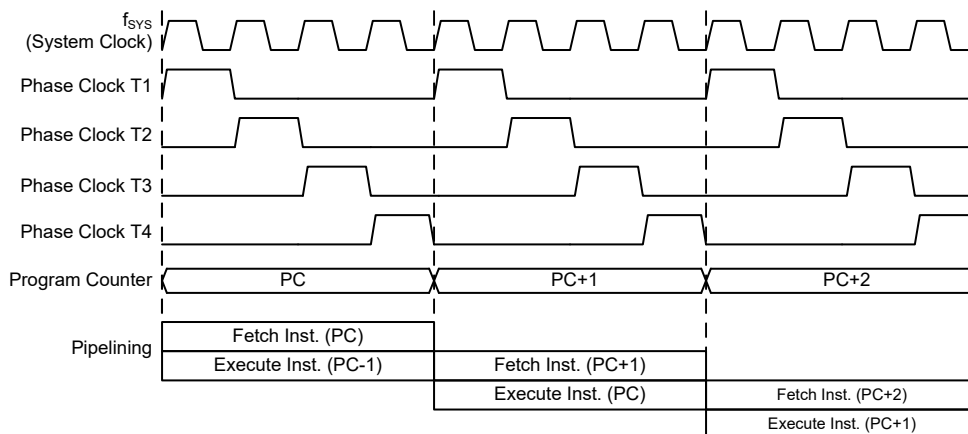
系统结构

内部系统结构是 Holtek 单片机具有良好性能的主要因素。由于采用 RISC 结构，此单片机具有高运算速度和高性能的特点。通过流水线的方式，指令的取得和执行同时进行，此举使得除了跳转和调用指令外，其它指令都能在一个指令周期内完成。8 位 ALU 参与指令集中所有的运算，它可完成算术运算、逻辑运算、移位、递增、递减和分支等功能，而内部的数据路径则是以通过累加器和 ALU 的方式加以简化。有些寄存器在数据存储器中被实现，且可以直接或间接寻址。简单的寄存器寻址方式和结构特性，确保了在提供具有较大可靠度和灵活性的 I/O 和 A/D 控制系统时，仅需要少数的外部器件。使得这些单片机适用于低成本和批量生产的控制应用。

时序和流水线结构

主系统时钟由 HIRC 或 LIRC 振荡器提供，它被细分为 T1~T4 四个内部产生的非重叠时序。在 T1 时间，程序计数器自动加一并抓取一条新的指令。剩下的时间 T2~T4 完成译码和执行功能，因此，一个 T1~T4 时钟周期构成一个指令周期。虽然指令的抓取和执行发生在连续的指令周期，但单片机流水线结构会保证指令在一个指令周期内被有效执行。除非程序计数器的内容被改变，如子程序的调用或跳转，在这种情况下指令将需要多一个指令周期的时间去执行。

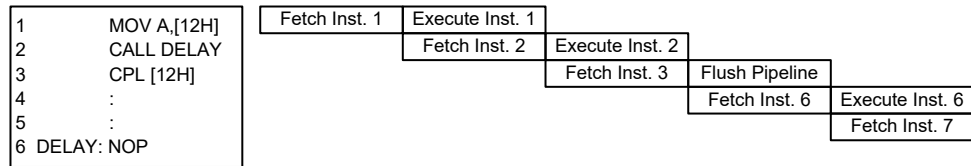
如果指令牵涉到分支，例如跳转或调用等指令，则需要两个指令周期才能完成指令执行。需要一个额外周期的原因是程序先用一个周期取出实际要跳转或调用的地址，再用另一个周期去实际执行分支动作，因此用户需要特别考虑额外周期的问题，尤其是在执行时间要求较严格的时候。



系统时序和流水线

如果指令牵涉到分支，例如跳转或调用等指令，则需要两个指令周期才能完成指令执行。需要一个额外周期的原因是程序先用一个周期取出实际要跳转或调

用的地址，再用另一个周期去实际执行分支动作，因此用户需要特别考虑额外周期的问题，尤其是在执行时间要求较严格的时候。



指令捕捉

程序计数器

在程序执行期间，程序计数器用来指向下一个要执行的指令地址。除了“JMP”和“CALL”指令需要跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完成以后自动加一。只有较低的 8 位，即所谓的程序计数器低字节寄存器 PCL，可以被用户直接读写。

当执行的指令要求跳转到不连续的地址时，如跳转指令、子程序调用、中断或复位等，单片机通过加载所需要的位址到程序寄存器来控制程序，对于条件跳转指令，一旦条件符合，在当前指令执行时取得的下一条指令将会被舍弃，而由一个空指令周期来取代。

程序计数器	
高字节	低字节 (PCL 寄存器)
PC10~PC8	PCL7~PCL0

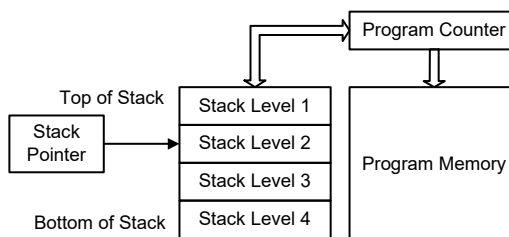
程序计数器的低字节，即程序计数器的低字节寄存器 PCL，可以通过程序控制，且它是可以读取和写入的寄存器。通过直接写入数据到这个寄存器，一个程序短跳转可直接执行，然而只有低字节的操作是有效的，跳转被限制在存储器的当前页中，即 256 个存储器地址范围内，当这样一个程序跳转要执行时，会插入一个空指令周期。PCL 的使用可能引起程序跳转，因此需要额外的指令周期。

堆栈

堆栈是一个特殊的存储空间，用来存储程序计数器中的内容。该单片机有 4 层堆栈，堆栈既不是数据部分也不是程序空间部分，而且它既不是可读取也不是可写入的。当前层由堆栈指针 (SP) 加以指示，同样也是不可读写的。在子程序调用或中断响应服务时，程序计数器的内容被压入到堆栈中。当子程序或中断响应结束时，返回指令 (RET 或 RETI) 使程序计数器从堆栈中重新得到它以前的值。当一个芯片复位后，堆栈指针将指向堆栈顶部。

如果堆栈已满，且有非屏蔽的中断发生，中断请求标志会被置位，但中断响应将被禁止。当堆栈指针减少 (执行 RET 或 RETI)，中断将被响应。这个特性提供程序设计者简单的方法来预防堆栈溢出。然而即使堆栈已满，CALL 指令仍然可以被执行，而造成堆栈溢出。使用时应避免堆栈溢出的情况发生，因为这可能导致不可预期的程序分支指令执行错误。

若堆栈溢出，则首个存入堆栈的程序计数器数据将会丢失。



算术逻辑单元 – ALU

算术逻辑单元是单片机中很重要的部分，执行指令集中的算术和逻辑运算。ALU 连接到单片机的数据总线，在接收相关的指令码后执行需要的算术与逻辑操作，并将结果存储在指定的寄存器，当 ALU 计算或操作时，可能导致进位、借位或其它状态的改变，而相关的状态寄存器会因此更新内容以显示这些改变，ALU 所提供的功能如下：

- 算术运算：
 - ADD, ADDM, ADC, ADCM, SUB, SUBM, SBC, SBCM, DAA
- 逻辑运算：
 - AND, OR, XOR, ANDM, ORM, XORM, CPL, CPLA
- 移位运算：
 - RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
- 递增和递减：
 - INCA, INC, DECA, DEC
- 分支判断：
 - JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA, CALL, RET, RETI

Flash 程序存储器

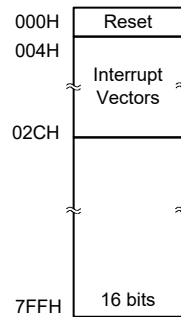
程序存储器用来存放用户代码即储存程序。程序存储器为 Flash 类型意味着可以多次重复编程，方便用户使用同一芯片进行程序的修改。使用适当的单片机编程工具，此单片机提供用户灵活便利的调试方法和项目开发规划及更新。

结构

程序存储器的容量为 2K×16 位，程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口。数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。

特殊向量

程序存储器内部某些地址保留用作诸如复位和中断入口等特殊用途。地址 000H 是芯片复位后的程序起始地址。在芯片复位之后，程序将跳到这个地址并开始执行。



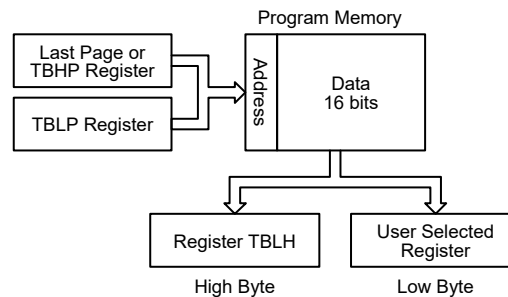
程序存储器结构

查表

程序存储器中的任何地址都可以定义成一个表格，以便储存固定的数据。使用表格时，表格指针必须先行设定，其方式是将表格的地址放在表格指针寄存器 TBLP 和 TBHP 中。这些寄存器定义表格总的地址。

在设定完表格指针后，表格数据可以使用“TABRD [m]”或“TABRDL [m]”指令分别从程序存储器查表读取。当这些指令执行时，程序存储器中表格数据低字节，将被传送到使用者所指定的数据存储器 [m]，程序存储器中表格数据的高字节，则被传送到 TBLH 特殊寄存器，而高字节中未使用的位将被读取为“0”。

下图是查表中寻址 / 数据流程：



查表范例

以下范例说明表格指针和表格数据如何被定义和执行。这个例子使用的表格数据用 ORG 伪指令储存在存储器中。ORG 指令的值“700H”指向的地址是单片机 2K 程序存储器中最后一页的起始地址。表格指针低字节寄存器的初始值设为 06H，这可保证从数据表格读取的第一笔数据位于程序存储器地址 706H，即最后一页起始地址后的第六个地址。值得注意的是，假如“TABRD [m]”指令被使用，则表格指针指向 TBHP 和 TBLP 指定的地址。在这个例子中，表格数据的高字节等于零，而当“TABRD [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

TBLH 寄存器为只读寄存器，不能重新储存，若主程序和中断服务程序都使用表格读取指令，应该注意它的保护。使用表格读取指令，中断服务程序可能会改变 TBLH 的值，若随后在主程序中再次使用这个值，则会发生错误，因此建议避免同时使用表格读取指令。然而在某些情况下，如果同时使用表格读取指令是不可避免的，则在执行任何主程序的表格读取指令前，中断应该先除能，另外要注意的是所有与表格相关的指令，都需要两个指令周期去完成操作。

表格读取程序范例

```

tempreg1 db ?      ; temporary register #1
tempreg2 db ?      ; temporary register #2
:
:
mov a,06h           ; initialise low table pointer - note that this address
                    ; is referenced
mov tblp,a          ; to the last page or specific page
mov a,07h           ; initialise high table pointer
mov tbhp,a
:
:
tabrd tempreg1      ; transfers value in table referenced by table pointer
                    ; data at program memory address "706H" transferred to
                    ; tempreg1 and TBLH
dec tblp            ; reduce value of table pointer by one
tabrd tempreg2      ; transfers value in table referenced by table pointer
                    ; data at program memory address "705H" transferred to
                    ; tempreg2 and TBLH.
                    ; In this example the data "1AH" is transferred to
                    ; tempreg1 and data "0FH" to register tempreg2 while the
                    ; value 00H will be transferred to the high byte
                    ; register TBLH
:
:
org 700h            ;sets initial address of program memory
dc 00Ah,00Bh,00Ch,00Dh,00Eh,00Fh,01Ah,01Bh
:

```

在线烧录

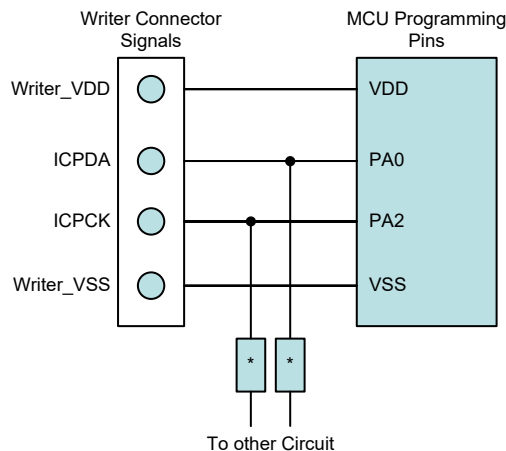
Flash 型程序存储器提供用户便利地对同一芯片进行程序的更新和修改。另外，Holtek 单片机提供 4 线接口的在线烧录方式。用户可将进行过烧录或未经过烧录的单片机芯片连同电路板一起制成，最后阶段进行程序的更新和程序的烧写，在无需去除或重新插入芯片的情况下方便地保持程序为最新版。

Holtek Flash 型单片机对应的烧录器引脚如下表所示：

Holtek 烧录器引脚	MCU 在线烧录引脚	功能
ICPDA	PA0	串行数据 / 地址烧录
ICPCK	PA2	串行时钟
VDD	VDD	电源
VSS	VSS	地

芯片内部程序存储器和 EEPROM 数据存储器都可以通过 4 线的接口在线进行烧录。其中一条用于数据串行下载或上传，一条用于串行时钟，两条用于提供电源。芯片在线烧写的详细使用说明超出此文档的描述范围，将由专门的参考文献提供。

在烧录过程中，烧录器会控制 ICPDA 和 ICPCK 脚进行数据和时钟烧录，用户必须确保这两个引脚没有连接至其它输出脚。



注：* 可能为电阻或电容。若为电阻则其值必须大于 1kΩ。若为电容则其值必须小于 1nF。

片上调试 – OCDS

EV 芯片 BS45V3833 用于 BS45F3833 单片机的仿真。此 EV 芯片提供片上调试功能 (OCDS) 用于开发过程中实际单片机的调试。除了片上调试功能，EV 芯片和实际 MCU 在功能上几乎是兼容的。用户可将 OCDSDA 和 OCDSCK 引脚连接至 Holtek HT-IDE 开发工具，用 EV 芯片来仿真实际单片机芯片的性能。OCDSDA 引脚为 OCDS 数据 / 地址输入 / 输出脚，OCDSCK 引脚为 OCDS 时钟输入脚。当用户用 EV 芯片进行调试时，OCDSDA 和 OCDSCK 引脚上的其它共用功能对 EV 芯片无效。这两个 OCDS 引脚与 ICP 引脚共用，因此在线烧录时仍用作 Flash 存储器烧录引脚。关于 OCDS 功能的详细描述，参考“Holtek e-Link for 8-bit MCU OCDS 使用手册”文档。

Holtek e-Link 引脚	EV 芯片引脚	引脚描述
OCDSDA	OCDSDA	片上调试数据 / 地址输入 / 输出
OCDSCK	OCDSCK	片上调试时钟输入
VDD	VDD	电源
GND	VSS	地

RAM 数据存储

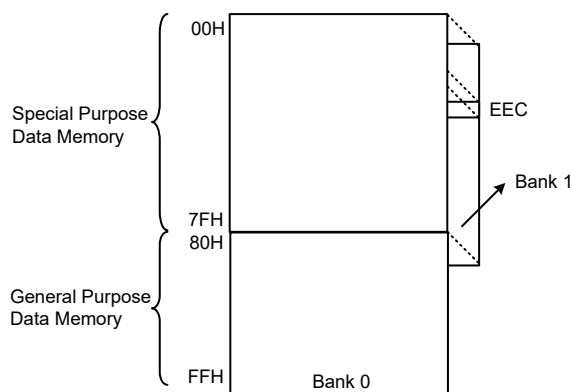
数据存储是内容可更改的 8 位 RAM 内部存储器，用来储存临时数据。

结构

数据存储分为两部分，第一部分是特殊功能数据存储，这些寄存器有固定的地址且与单片机的正确操作密切相关。大多特殊功能寄存器都可在程序控制下直接读取和写入，但有些被加以保护而不对用户开放。第二部分是通用数据存储，该存储器内所有地址都可在程序的控制下进行读取和写入。

整体数据存储被分为 2 个 Bank。除了在 40H 的 EEC 寄存器只能在 Bank1 中被寻址外，特殊功能数据寄存器在所有 Bank 中可被访问。可通过设置存储区指针 (BP) 的值实现不同数据存储 Bank 之间的切换。此单片机数据存储的起始地址为 00H。

特殊功能数据存储	通用数据存储	
地址	容量	地址
Bank 0: 00H~7FH Bank 1: 00H~7FH	128×8	Bank 0: 80H~FFH



数据存储结构

通用数据存储

所有的单片机程序需要一个读 / 写的存储区，让临时数据可以被储存和再使用，该 RAM 区域就是通用数据存储。这个数据存储区可让使用者进行读取和写入的操作。在程序控制下使用位操作指令可对个别的位做置位或复位的操作，较大地方便了用户在数据存储内进行位操作。

特殊功能数据存储

这个区域的数据存储器是存放特殊寄存器的，这些寄存器与单片机的正确操作密切相关。大多数的寄存器可进行读取和写入，只有一些是被写保护而只能读取的，相关细节的介绍请参看有关特殊功能寄存器的部分。要注意的是，任何读取指令对存储器中未定义地址进行读取将返回“00H”。

Bank 0 : Bank 1		Bank 0 : Bank 1	
00H	IAR0	40H	EEC
01H	MP0	41H	
02H	IAR1	42H	
03H	MP1	43H	SLEWC0
04H	BP	44H	SLEDC0
05H	ACC	45H	SLEDC1
06H	PCL	46H	PTMC0
07H	TBLP	47H	PTMC1
08H	TBLH	48H	PTMDL
09H	TBHP	49H	PTMDH
0AH	STATUS	4AH	PTMAL
0BH	SMOD	4BH	PTMAH
0CH	LVDC	4CH	PTMRPL
0DH	INTEG	4DH	PTMRPH
0EH	INTC0	4EH	CTM2C0
0FH	INTC1	4FH	CTM2C1
10H	INTC2	50H	CTM2DL
11H	MFI0	51H	CTM2DH
12H	MFI1	52H	CTM2AL
13H	MFI2	53H	CTM2AH
14H	PA	54H	CTM1C0
15H	PAC	55H	CTM1C1
16H	PAPU	56H	CTM1DL
17H	PAWU	57H	CTM1DH
18H	MFI3	58H	CTM1AL
19H		59H	CTM1AH
1AH	WDTC	5AH	
1BH	TBC	5BH	OCVPC0
1CH		5CH	OCVPC1
1DH		5DH	OCVPC2
1EH	EEA	5EH	OCVPDA
1FH	EED	5FH	OCVPOCAL
20H		60H	OCVPCCAL
21H		61H	
22H		62H	TKTMR
23H		63H	TKC0
24H	FADJL	64H	TK16DL
25H	FADJH	65H	TK16DH
26H	CTRL	66H	TKC1
27H	LVRC	67H	TKM016DL
28H	CTM0C0	68H	TKM016DH
29H	CTM0C1	69H	TKM0ROL
2AH	CTM0DL	6AH	TKM0ROH
2BH	CTM0DH	6BH	TKM0C0
2CH	CTM0AL	6CH	TKM0C1
2DH	CTM0AH	6DH	SADOL
2EH		6EH	SADOH
2FH	STMC0	6FH	SADC0
30H	STMC1	70H	SADC1
31H	STMDL	71H	ACERL
32H	STMDH	72H	
33H	STMAL		
34H	STMAH		
35H			
36H	PC		
37H	PCC		
38H	PCPU		
39H	CTRL2		
3AH	CTRL3		
3BH	CTRL4		
3CH	CTRL5		
3DH	PB		
3EH	PBC		
3FH	PBPU		

: Unused, read as 00H

特殊功能数据存储结构

特殊功能寄存器描述

大部分特殊功能寄存器的细节将在相关功能章节描述，但有几个寄存器需在此章节单独描述。

间接寻址寄存器 – IAR0, IAR1

间接寻址寄存器 IAR0 和 IAR1 的地址虽位于数据存储区，但其并没有实际的物理地址。间接寻址的方法准许使用间接寻址寄存器和存储器指针做数据操作，以取代定义实际存储器地址的直接存储器寻址方法。在间接寻址寄存器 IAR0 和 IAR1 上的任何动作，将对间接寻址指针 MP0 或 MP1 所指定的存储器地址产生对应的读 / 写操作。它们总是成对出现，IAR0 和 MP0 可以访问 Bank 0，而 IAR1 和 MP1 可以访问任何 Bank。因为这些间接寻址寄存器不是实际存在的，直接读取将返回 “00H” 的结果，而直接写入此寄存器则不做任何操作。

存储器指针 – MP0, MP1

该单片机提供两个存储器指针，即 MP0 和 MP1。由于这些指针在数据存储区中能像普通的寄存器一般被操作，因此提供了一个寻址和数据追踪的有效方法。当对间接寻址寄存器进行任何操作时，单片机指向的实际地址是由相关的存储器指针所指定的地址。MP0 和间接寻址寄存器 IAR0 用于访问 Bank 0，而 MP1 和 IAR1 根据 BP 寄存器可以访问所有的 Bank。直接寻址仅可以用在 Bank 0 中，所有 Bank 都可使用 MP1 和 IAR1 进行间接寻址。

以下例子说明如何清除一个具有 4 RAM 地址的区块，它们已事先定义成地址 adres1 到 adres4。

间接寻址程序范例

```
data .section 'data'
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block db ?
code .section at 0 'code'
org 00h
start:
    mov a, 04h                ; setup size of block
    mov block, a
    mov a, offset adres1      ; Accumulator loaded with first RAM address
    mov mp0, a                ; setup memory pointer with first RAM address
loop:
    clr IAR0                  ; clear the data at address defined by mp0
    inc mp0                    ; increment memory pointer
    sdz block                  ; check if last memory location has been cleared
    jmp loop
continue:
```

在上面的例子中有一点值得注意，即并没有确定 RAM 地址。

存储区指针 – BP

该单片机中，数据存储区被分为 2 个 bank，即 Bank 0~Bank 1。可以通过设置存储区指针 (Bank Pointer) 值来访问不同的数据存储区。BP 指针的第 0 位用于选择数据存储器的 Bank 0 或 Bank 1。

复位后，数据存储器会初始化到 Bank 0，但是在休眠模式或空闲模式下的 WDT 溢出复位，不会改变数据存储器的存储区号。应该注意的是特殊功能数据存储器不受存储区的影响，也就是说，不论是在哪一个存储区，都能对特殊功能寄存器进行读写操作。数据存储器的直接寻址总是访问 Bank 0，不影响存储区指针的值。要访问 Bank1，则必须要使用间接寻址方式。

● BP 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DMBP0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 未定义，读为“0”

Bit 0 **DMBP0**: 数据存储器 Bank 选择位
0: Bank 0
1: Bank 1

累加器 – ACC

对任何单片机来说，累加器是相当重要的，且与 ALU 所完成的运算有密切关系，所有 ALU 得到的运算结果都会暂时存在 ACC 累加器里。若没有累加器，ALU 必须在每次进行如加法、减法和移位的运算时，将结果写入到数据存储器，这样会造成程序编写和时间的负担。另外数据传送也常常牵涉到累加器的临时储存功能，例如在使用者定义的一个寄存器和另一个寄存器之间传送数据时，由于两寄存器之间不能直接传送数据，因此必须通过累加器来传送数据。

程序计数器低字节寄存器 – PCL

为了提供额外的程序控制功能，程序计数器低字节设置在数据存储器的特殊功能区域内，程序员可对此寄存器进行操作，很容易的直接跳转到其它程序地址。直接给 PCL 寄存器赋值将导致程序直接跳转到程序存储器的某一地址，然而由于寄存器只有 8 位长度，因此只允许在本页的程序存储器范围内进行跳转，而当使用这种运算时，要注意会插入一个空指令周期。

查表寄存器 – TBLP, TBHP, TBLH

这三个特殊功能寄存器对存储在程序存储器中的表格进行操作。TBLP 和 TBHP 为表格指针，指向表格数据存储的地址。它们的值必须在任何表格读取指令执行前加以设定，由于它们的值可以被如“INC”或“DEC”的指令所改变，这就提供了一种简单的方法对表格数据进行读取。表格读取数据指令执行之后，表格数据高字节存储在 TBLH 中。其中要注意的是，表格数据低字节会被传送到使用者指定的地址。

状态寄存器 – STATUS

这 8 位的状态寄存器由零标志位 (Z)、进位标志位 (C)、辅助进位标志位 (AC)、溢出标志位 (OV)、暂停标志位 (PDF) 和看门狗定时器溢出标志位 (TO) 组成。这些算术 / 逻辑操作和系统运行标志位是用来记录单片机的运行状态。

除了 PDF 和 TO 标志外，状态寄存器中的位像其它大部分寄存器一样可以被改变。任何数据写入到状态寄存器将不会改变 TO 或 PDF 标志位。另外，执行不同的指令后，与状态寄存器有关的运算可能会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出或执行“CLR WDT”或“HALT”指令影响。PDF 标志位只会受执行“HALT”或“CLR WDT”指令或系统上电影响。

Z、OV、AC 和 C 标志位通常反映最新运算的状态。

- C: 当加法运算的结果产生进位，或减法运算的结果没有产生借位时，C 被置位，否则 C 被清零，同时 C 也会被带进位的移位指令所影响。
- AC: 当低半字节加法运算的结果产生进位，或低半字节减法运算的结果没有产生借位时，AC 被置位，否则 AC 被清零。
- Z: 当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零。
- OV: 当运算结果高两位的进位状态异或结果为 1 时，OV 被置位，否则 OV 被清零。
- PDF: 系统上电或执行“CLR WDT”指令会清零 PDF，而执行“HALT”指令则会置位 PDF。
- TO: 系统上电或执行“CLR WDT”或“HALT”指令会清零 TO，而当 WDT 溢出则会置位 TO。

另外，当进入一个中断程序或执行子程序调用时，状态寄存器不会自动压入到堆栈保存。假如状态寄存器的内容是重要的且子程序可能改变状态寄存器的话，则需谨慎的去做正确的储存。

● STATUS 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W
POR	—	—	0	0	x	x	x	x

“x”：未知

Bit 7~6 未定义，读为“0”

Bit 5 **TO**: 看门狗溢出标志位
0: 系统上电或执行“CLR WDT”或“HALT”指令后
1: 看门狗溢出发生

Bit 4 **PDF**: 暂停标志位
0: 系统上电或执行“CLR WDT”指令后
1: 执行“HALT”指令

Bit 3 **OV**: 溢出标志位
0: 无溢出
1: 运算结果高两位的进位状态异或结果为 1

Bit 2 **Z**: 零标志位
0: 算术或逻辑运算结果不为 0
1: 算术或逻辑运算结果为 0

Bit 1 **AC**: 辅助进位标志位
0: 无辅助进位
1: 在加法运算中低四位产生了向高四位进位，或减法运算中低四位不发生从高四位借位

Bit 0 **C**: 进位标志位
0: 无进位
1: 如果在加法运算中结果产生了进位，或在减法运算中结果不发生借位
“C”位也受带进位的移位指令的影响。

EEPROM 数据存储

该单片机的一个特性是内建 EEPROM 数据存储。EEPROM 为电可擦可编程存储器，由于其非易失的存储结构，即使在电源掉电的情况下存储器内的数据仍然保存完好。这种存储区扩展了数据存储空间，对设计者来说增加了许多新的应用机会。EEPROM 可以用来存储产品编号、校准值、用户特定数据、系统配置参数或其它产品信息等。EEPROM 的数据读取和写入过程也会变的更简单。

EEPROM 数据存储结构

该单片机的 EEPROM 数据存储容量为 32×8。由于映射方式与程序存储器和数据存储不同，因此不能像其它类型的存储器一样寻址。使用 Bank0 中的地址和数据寄存器以及 Bank1 中的一个控制寄存器，可以实现对 EEPROM 的单字节读写操作。

EEPROM 寄存器

有三个寄存器控制内部 EEPROM 数据存储总的操作，即地址寄存器 EEA、数据寄存器 EED 及控制寄存器 EEC。EEA 和 EED 寄存器位于 Bank 0 中，它们能像其它特殊功能寄存器一样直接被访问。而 EEC 位于 Bank1 中，只能通过存储器指针 MP1 和间接寻址寄存器 IAR1 进行间接读取或写入。由于 EEC 控制寄存器位于 Bank 1 中的“40H”，在 EEC 寄存器上的任何操作被执行前，存储器指针 MP1 必须先设为“40H”，存储区指针寄存器 BP 则设置为“01H”。

寄存器名称	位							
	7	6	5	4	3	2	1	0
EEA	—	—	—	EEA4	EEA3	EEA2	EEA1	EEA0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

EEPROM 控制寄存器列表

• EEA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 未定义，读为“0”

Bit 4~0 数据 EEPROM 地址

数据 EEPROM 地址 bit 4~bit 0

• EED 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 数据 EEPROM 数据

数据 EEPROM 数据 Bit 7~Bit 0

• EEC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义，读为“0”

Bit 3 **WREN**: 数据 EEPROM 写使能位

0: 除能

1: 使能

此位为数据 EEPROM 写使能位，向数据 EEPROM 写操作之前需将此位置高。将此位清零时，则禁止向数据 EEPROM 写操作。

Bit 2 **WR**: EEPROM 写控制位

0: 写周期结束

1: 激活写周期

此位为数据 EEPROM 写控制位，由应用程序将此位置高将激活写周期。写周期结束后，硬件自动将此位清零。当 WREN 未先置高时，此位置高无效。

Bit 1 **RDEN**: 数据 EEPROM 读使能位

0: 除能

1: 使能

此位为数据 EEPROM 读使能位，向数据 EEPROM 读操作之前需将此位置高。将此位清零时，则禁止向数据 EEPROM 读操作。

Bit 0 **RD**: EEPROM 读控制位

0: 读周期结束

1: 激活读周期

此位为数据 EEPROM 读控制位，由应用程序将此位置高将激活读周期。读周期结束后，硬件自动将此位清零。当 RDEN 未首先置高时，此位置高无效。

注：在同一条指令中 WREN、WR、RDEN 和 RD 不能同时置为“1”。WR 和 RD 不能同时置为“1”。

从 EEPROM 中读取数据

从 EEPROM 中读取数据，EEPROM 中读取数据的地址要先放入 EEA 寄存器中。EEC 寄存器中的读使能位 RDEN 先置为高以使能读功能。若 EEC 寄存器中的 RD 位被置高，一个读周期将开始。若 RD 位已置为高而 RDEN 位还未被设置则不能开始读操作。若读周期结束，RD 位将自动清除为“0”，数据可以从 EED 寄存器中读取。数据在其它读或写操作执行前将一直保留在 EED 寄存器中。应用程序将轮询 RD 位以确定数据可以有效地被读取。

写数据到 EEPROM

写数据至 EEPROM，EEPROM 中写入数据的地址要先放入 EEA 寄存器中，写入的数据需存入 EED 寄存器中。EEC 寄存器中的写使能位 WREN 先置为高以使能写功能，然后 EEC 寄存器中的 WR 位需立即置高以开始写操作，这两条指令必须连续执行。总中断位 EMI 在写周期开始前应当被清零，写周期开始后再将其使能。若 WR 位已置为高而 WREN 位还未被设置则不能开始写操作。由于控制 EEPROM 写周期是一个内部时钟，与单片机的系统时钟异步，所以数据写入 EEPROM 的时间将有所延迟。可通过轮询 EEC 寄存器中的 WR 位或判断 EEPROM 写中断以侦测写周期是否完成。若写周期完成，WR 位将自动清除为“0”，通知用户数据已写入 EEPROM。因此，应用程序将轮询 WR 位以确定写周期是否结束。

写保护

防止误写入的写保护有以下几种。单片机上电后控制寄存器中的写使能位将被清除以杜绝任何写入操作。上电后存储区指针 BP 将重置为“0”，这意味着数据存储器 Bank 0 被选中。由于 EEPROM 控制寄存器位于 Bank 1 中，这增加了对写操作的保护措施。在正常程序操作中确保控制寄存器中的写使能位被清除将能防止不正确的写操作。

EEPROM 中断

EEPROM 写周期结束后将产生 EEPROM 写中断，需先通过设置相关中断寄存器的 DEE 位使能 EEPROM 中断。由于 EEPROM 中断包含在多功能中断中，相应的多功能中断使能位需被设置。当 EEPROM 写周期结束，DEF 请求标志位及其相关多功能中断请求标志位将被置位。若总中断，EEPROM 和多功能中断使能且堆栈未满的情况下将跳转到相应的多功能中断向量中执行。当中断被响应，只有多功能中断标志位将自动复位，而 EEPROM 中断标志将通过应用程序手动复位。更多详情请参见中断章节。

编程注意事项

必须注意的是数据不会无意写入 EEPROM。在没有写动作时写使能位被正常清零可以达到定期保护功能。存储区指针 BP 也可以正常清零以阻止进入 EEPROM 控制寄存器存在的 Bank 1。尽管没有必要，写一个简单的读回程序以检查新写入的数据是否正确还是应该考虑的。WREN 位置位后，EEC 寄存器中的 WR 位需立即置位，以确保写周期正确地执行。写周期执行前总中断位 EMI 应先清零，写周期开始执行后再将此位重新使能。注意，单片机不应在 EEPROM 读或写操作完全完成之前进入空闲或休眠模式，否则 EEPROM 读或写操作将失败。

程序范例

从 EEPROM 中读取数据 – 轮询法

```
MOV A, EEPROM_ADRES      ; user defined address
MOV EEA, A
MOV A, 040H               ; setup memory pointer MP1
MOV MP1, A                ; MP1 points to EEC register
MOV A, 01H                ; setup Bank Pointer
MOV BP, A
SET IAR1.1                ; set RDEN bit, enable read operations
SET IAR1.0                ; start Read Cycle - set RD bit
BACK:
SZ IAR1.0                 ; check for read cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM write
CLR BP
MOV A, EED                ; move read data to register
MOV READ_DATA, A
```

写数据到 EEPROM – 轮询法

```
MOV A, EEPROM_ADRES      ; user defined address
MOV EEA, A
MOV A, EEPROM_DATA        ; user defined data
MOV EED, A
MOV A, 040H               ; setup memory pointer MP1
MOV MP1, A                ; MP1 points to EEC register
```

```

MOV A, 01H                ; setup Bank Pointer
MOV BP, A                  ; BP points to data memory bank 1
CLR EMI
SET IAR1.3                 ; set WREN bit, enable write operations
SET IAR1.2                 ; start Write Cycle - set WR bit - executed
                           ; immediately after set WREN bit

SET EMI
BACK:
SZ IAR1.2                  ; check for write cycle end
JMP BACK
CLR IAR1                   ; disable EEPROM write
CLR BP

```

振荡器

不同的振荡器选择可以让使用者在不同的应用需求中实现更大范围的功能。振荡器的灵活性使得在速度和功耗方面可以达到较佳的优化。振荡器选择和操作是通过配置选项和寄存器共同完成的。

振荡器概述

振荡器除了作为系统时钟源，还作为看门狗定时器和时基中断的时钟源。高度集成的内部振荡器不需要任何外围器件。它们提供的高速和低速系统振荡器具有较宽的频率范围。较高频率的振荡器提供更高的性能，但要求有更高的功率，反之亦然。动态切换快慢系统时钟的能力使单片机具有灵活而优化的性能 / 功耗比，此特性对功耗敏感的应用领域尤为重要。

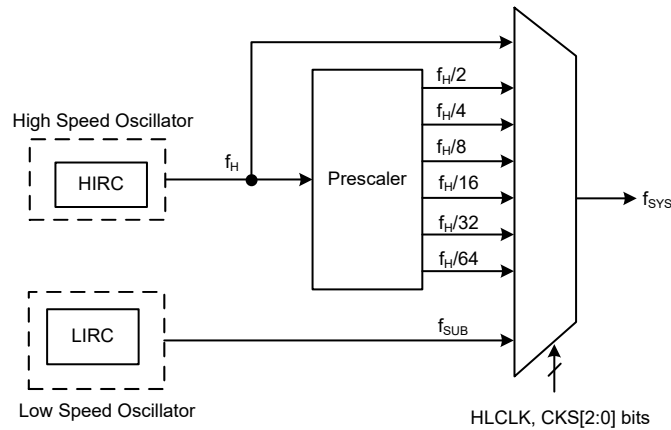
类型	名称	频率
内部高速 RC 振荡器	HIRC	4MHz/8MHz/12MHz
内部低速 RC 振荡器	LIRC	32kHz
PTM 专用 RC 振荡器	PTMOSC	12MHz (可调)

振荡器类型

系统时钟配置

该单片机有两个振荡器可用来产生系统时钟，即一个高速内部 RC 振荡器和一个低速内部 RC 振荡器。使用高速或低速振荡器作为系统时钟的选择是通过设置 SMOD 寄存器中的 HLCLK 位和 CKS2~CKS0 位决定的，系统时钟可动态选择。

通过寄存器选择采用高速或低速振荡器作为实际时钟源。利用 SMOD 寄存器中的 HLCLK 位和 CKS2~CKS0 位段来确定低速或高速系统时钟的频率。需注意两个振荡器必须做出选择，即选择高速或选择低速系统振荡器。



系统时钟配置

内部高速 RC 振荡器 – HIRC

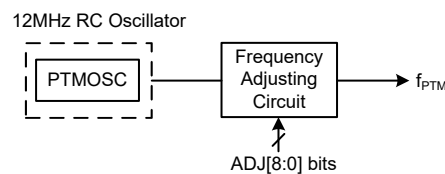
内部 RC 振荡器是一个集成的系统振荡器，不需其它外部器件。内部 RC 振荡器具有三种固定的频率：4MHz、8MHz 和 12MHz。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因电源电压、温度以及芯片制成工艺不同的影响较大程度地降低。

内部 32kHz 振荡器 – LIRC

内部 32kHz 系统振荡器是一个低频振荡器。这是一个完全集成 RC 振荡器，它在 5V 电压下运行的典型频率值为 32kHz 且无需外部元件。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡器因电源电压、温度及芯片制成工艺不同的影响较大程度地降低。

PTM 专用 RC 振荡器 – PTMOSC

PTM 专用振荡器也是一个完全集成的 RC 振荡器，无需其它外部器件。内部 RC 振荡器有一个 12MHz 的频率，可通过改变 ADJ[8:0] 位段的值进行调整。芯片在制造时进行调整且内部含有频率补偿电路，使得振荡频率因电源电压、温度以及芯片制成工艺不同的影响较大程度地降低。



位于 FADJH 和 FADJL 寄存器中的 ADJ[8:0] 位段用于调整 PTMOSC 振荡频率。ADJ[8:0] 位段的默认值为 100000000B，此时 PTMOSC 振荡器的频率应为 12MHz。写入 ADJ[8:0] 位段的值越大，所得 PTMOSC 的频率就越低。当 ADJ[8:0] 位段设置为 00000000B 时 PTMOSC 得最大调整频率。需注意当配置 ADJ[8:0] 位段改变 PTMOSC 频率时，PTMOSC 振荡器达到稳定需要一定的延时。直到所更新的频率稳定后才可使用新的 PTMOSC 频率。

● FADJL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ADJ7	ADJ6	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **ADJ7~ADJ0**: PTMOSC 频率调整控制位 bit 7~bit 0

● FADJH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	ADJ8
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	1

Bit 7~1 未定义，读为“0”

Bit 0 **ADJ8**: PTMOSC 频率调整控制位 bit 8

● CTRL2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FPTMEN	FPTMF	—	—	—	—	OCVPS1	OCVPS0
R/W	R/W	R	—	—	—	—	R/W	R/W
POR	1	0	—	—	—	—	0	0

Bit 7 **FPTMEN**: PTMOSC 振荡器使能控制位

0: 除能

1: 使能

Bit 6 **FPTMF**: PTMOSC 振荡器稳定标志位

0: 未稳定

1: 稳定

此位用于表明 PTMOSC 振荡器是否稳定。当 FPTMEN 位置高使能 PTMOSC 振荡器，PTMOSC 振荡器需要 256 个时钟才能稳定。

Bit 5~2 未定义，读为“0”

Bit 1~0 **OCVPS1~OCVPS**: OCVPAI0 信号输入源引脚选择位

详见过电流 / 电压保护章节

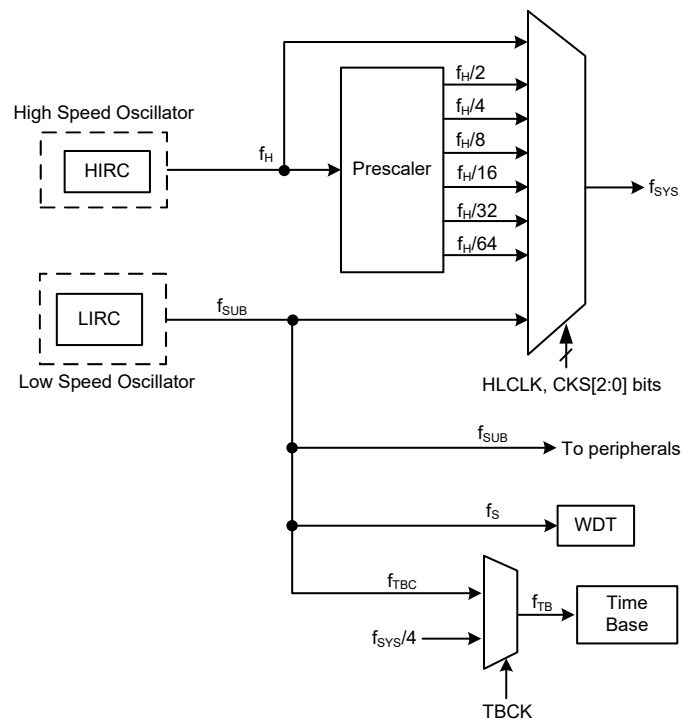
工作模式和系统时钟

现今的应用要求单片机具有较高的性能及尽可能低的功耗，这种矛盾的要求在便携式电池供电的应用领域尤为明显。高性能所需要的高速时钟将增加功耗，反之亦然。Holtek 单片机了提供高、低速两种时钟源，它们之间可以动态切换，用户可通过优化单片机操作来获得较佳性能 / 功耗比。

系统时钟

该单片机为 CPU 和外围功能操作提供了多种不同的时钟源。用户使用配置选项和寄存器编程可获取多种时钟，进而使系统时钟获取较大的应用性能。

主系统时钟可来自高频时钟源 f_H 或低频时钟源 f_{SUB} ，通过 SMOD 寄存器中的 HLCLK 位和 CKS2~CKS0 位进行选择。高频系统时钟来自 HIRC 振荡器，低频系统时钟源来自 LIRC 振荡器。其它系统时钟还有高速系统振荡器的分频 $f_H/2 \sim f_H/64$ 。



单片机时钟配置

注：当系统时钟源 f_{SYS} 由 f_H 切换到 f_{SUB} ，高速振荡器将停止以节省耗电。因此没有 $f_H \sim f_H/64$ 的时钟可供外部电路使用。

系统工作模式

单片机有 6 种不同的工作模式，每种有它自身的特性，根据应用中不同的性能和功耗要求可选择不同的工作模式。单片机正常工作有两种模式：正常模式和低速模式。剩余的 4 种工作模式：休眠模式 0、休眠模式 1、空闲模式 0 和空闲模式 1 用于单片机 CPU 关闭时以节省耗电。

工作模式	说明			
	CPU	f _{SYS}	f _H	f _{SUB}
正常模式	On	f _H ~f _H /64	On	On
低速模式	On	f _{SUB}	Off	On
空闲模式 0	Off	Off	Off	On
空闲模式 1	Off	On	On/Off	On
休眠模式 0	Off	Off	Off	Off
休眠模式 1	Off	Off	Off	On

正常模式

顾名思义，这是主要的工作模式之一，单片机的所有功能均可在此模式中实现且系统时钟由一个高速振荡器提供。该模式下单片机正常工作的时钟源来自 HIRC 振荡器。高速振荡器频率可被分为 1~64 的不等比率，实际的比率由 SMOD 寄存器中的 HLCLK 位和 CKS2~CKS0 位进行选择。单片机使用高速振荡器分频作为系统时钟可减少工作电流。

低速模式

此模式的系统时钟虽为较低速时钟源，但单片机仍能正常工作。该低速时钟源可来自 f_{SUB}。单片机在此模式中运行所耗工作电流较低。在低速模式中，f_H 关闭。

休眠模式 0

在 HALT 指令执行后且 SMOD 寄存器中的 IDLEN 位为低时，系统进入休眠模式。在休眠模式 0 中，CPU 停止运行，f_{SUB} 和 f_S 时钟也将停止，看门狗定时器除能。在此模式下，LVDEN 必须清零，若 LVDEN 置高则无法进入休眠模式 0。

休眠模式 1

在 HALT 指令执行后且 SMOD 寄存器中的 IDLEN 位为低时，系统进入休眠模式。在休眠模式 1 中，CPU 停止运行，若 LVDEN 为“1”或看门狗定时器使能，则 f_{SUB} 和 f_S 时钟继续运行。

空闲模式 0

执行 HALT 指令后且 SMOD 寄存器中的 IDLEN 位为高，CTRL 寄存器中的 FSYSON 位为低时，系统进入空闲模式 0。在空闲模式 0 中，CPU 停止，系统振荡器停止，但 f_{SUB} 时钟将开启以驱动一些外围功能。

空闲模式 1

执行 HALT 指令后且 SMOD 寄存器中的 IDLEN 位为高，CTRL 寄存器中的 FSYSON 位为高时，系统进入空闲模式 1。在空闲模式 1 中，CPU 停止，但高速或低速振荡器都将继续运行。在空闲模式 1 中低频时钟 f_{SUB} 将开启以驱动一些外围功能。

注：若 LVDEN=1 且进入了休眠或空闲模式，则 LVD 功能将不被除能，f_{SUB} 时钟将强制使能。

控制寄存器

SMOD 和 CTRL 寄存器用于控制单片机的内部时钟。

• SMOD 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	LTO	HTO	IDLEN	HLCLK
R/W	R/W	R/W	R/W	—	R	R	R/W	R/W
POR	0	0	0	—	0	0	1	1

Bit 7~5 **CKS2~CKS0**: HLCLK 为 “0” 时系统时钟选择位

000: f_{SUB}
001: f_{SUB}
010: $f_H/64$
011: $f_H/32$
100: $f_H/16$
101: $f_H/8$
110: $f_H/4$
111: $f_H/2$

这三位用于选择系统时钟源。除了 LIRC 可作为系统时钟源外，高速系统振荡器的分频比也可作为系统时钟源。

Bit 4 未定义，读为 “0”

Bit 3 **LTO**: 低速振荡器就绪标志位

0: 未就绪
1: 就绪

此位为低速系统振荡器就绪标志位，用于表明低速系统振荡器在系统上电复位或唤醒后何时稳定下来，唤醒后该位转换为高需 1~2 个周期。

Bit 2 **HTO**: 高速振荡器就绪标志位

0: 未就绪
1: 就绪

此位为高速系统振荡器就绪标志位，用于表明唤醒后高速系统振荡器何时稳定下来。此标志在系统上电后经硬件清零，高速系统振荡器稳定后变为高电平。因此，此位在单片机上电后由应用程序读取的值为 “1”。在休眠模式或空闲模式 0 中，该标志上电复位或唤醒后会处于低电平状态，若使用 HIRC 振荡器则该标志位将在 15~16 个时钟周期后变为高电平状态。

Bit 1 **IDLEN**: 空闲模式控制位

0: 除能
1: 使能

此位为空闲模式控制位，用于决定 HALT 指令执行后发生的动作。若此位为高，当指令 HALT 执行后，单片机进入空闲模式。若 FSYSON 位为高，在空闲模式 1 中 CPU 停止运行，系统时钟将继续工作以保持外围功能继续工作；若 FSYSON 为低，在空闲模式 0 中 CPU 和系统时钟都将停止运行。若此位为低，单片机将在 HALT 指令执行后进入休眠模式。

Bit 0 **HLCLK**: 系统时钟选择位

0: $f_H/2 \sim f_H/64$ 或 f_{SUB}
1: f_H

此位用于选择 f_H 或 $f_H/2 \sim f_H/64$ 或 f_{SUB} 作为系统时钟。该位为高时选择 f_H 作为系统时钟，为低时则选择 $f_H/2 \sim f_H/64$ 或 f_{SUB} 作为系统时钟。当系统时钟由 f_H 时钟向 f_{SUB} 时钟转换时， f_H 将自动关闭以降低功耗。

• CTRL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

“x”：未知

Bit 7 **FSYSON**: 空闲模式下 f_{sys} 控制位
0: 除能
1: 使能

Bit 6~3 未定义，读为“0”

Bit 2 **LVRF**: LVR 复位标志位
详见其它章节

Bit 1 **LRF**: LVRC 控制寄存器软件复位标志位
详见其它章节

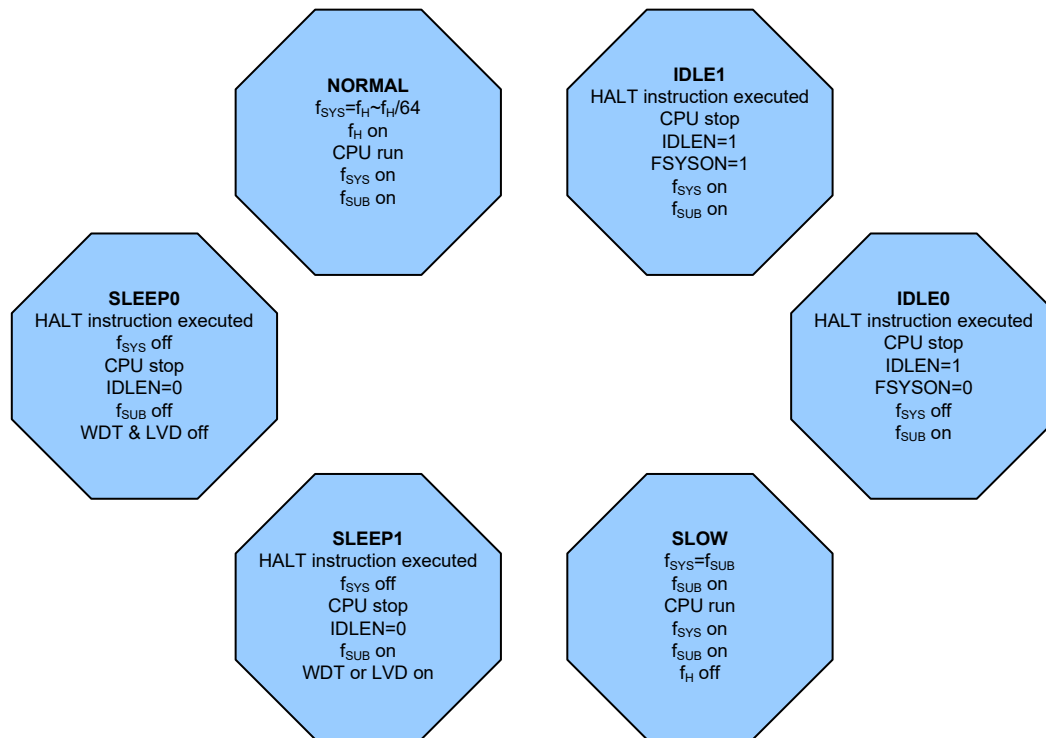
Bit 0 **WRF**: WDTC 控制寄存器软件复位标志位
详见其它章节

工作模式切换

单片机可在各个工作模式间自由切换，使得用户可根据所需选择较佳的性能 / 功耗比。用此方式，对单片机工作的性能要求不高的情况下，可使用较低频时钟以减少工作电流，在便携式应用上延长电池的使用寿命。

简单来说，正常模式和低速模式间的切换仅需设置 SMOD 中的 HLCLK 位及 CKS2~CKS0 位即可实现，而正常模式 / 低速模式与休眠模式 / 空闲模式间的切换经由 HALT 指令实现。当 HALT 指令执行后，单片机是否进入空闲模式或休眠模式由 SMOD 寄存器中的 IDLEN 位和 CTRL 寄存器中的 FSYSON 位决定的。

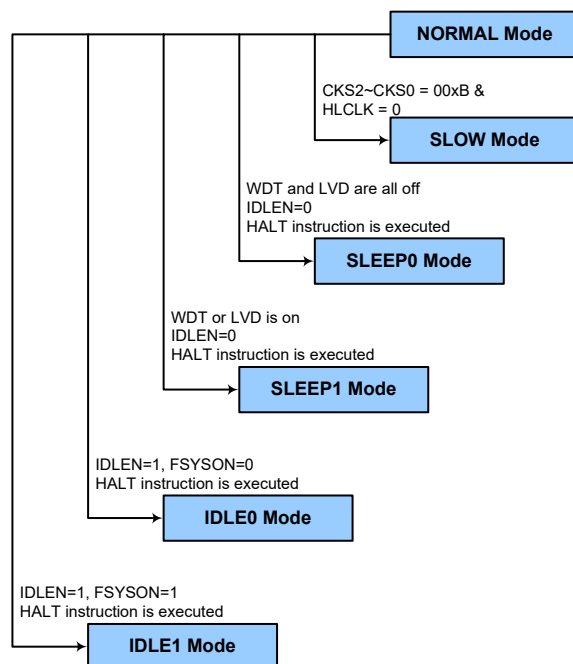
当 HLCLK 位变为低电平时，时钟源将由高速时钟源 f_H 转换成时钟源 $f_H/2 \sim f_H/64$ 或 f_{SUB} 。若时钟源来自 f_{SUB} ，高速时钟源将停止运行以节省耗电。此时必须注意内部时钟源 $f_H/16 \sim f_H/64$ 也将停止运行，这可能会影响到其它内部功能如 TM 的操作。所附流程图显示了单片机在不同工作模式间切换时的变化。



正常模式切换到低速模式

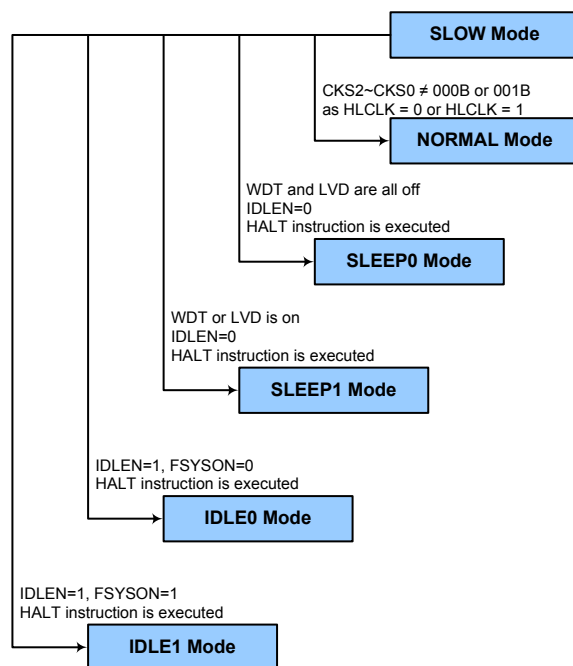
系统运行在正常模式时使用高速系统振荡器，因此较为耗电。可通过设置 SMOD 寄存器中的 HLCLK 位为“0”及 CKS2~CKS0 位为“000”或“001”使系统时钟切换至运行在低速模式下。此时将使用低速系统振荡器以节省耗电。用户可在对性能要求不高的操作中使用此方法以减少耗电。

低速模式的时钟源来自 LIRC 振荡器，因此要求该振荡器在所有模式切换动作发生前稳定下来。该动作由 SMOD 寄存器中 LTO 位监测。



低速模式切换到正常模式

在低速模式系统使用 LIRC 低速振荡器。切换回使用高速系统时钟振荡器的正常模式需设置 HLCLK 位为“1”，也可设置 HLCLK 位为“0”但 CKS2~CKS0 需设为“010”、“011”、“100”、“101”、“110”或“111”。高频时钟需要一定的稳定时间，通过检测 HTO 位的状态可进行判断。高速振荡器稳定需要多少的延时取决于所选的高速振荡器类型。



进入休眠模式 0

进入休眠模式 0 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 和 LVD 功能除能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟与 f_{SUB} 时钟停止运行，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- WDT 将被清零并停止运行。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入休眠模式 1

进入休眠模式 1 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“0”且 WDT 或 LVD 功能使能。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟停止运行，应用程序停止在“HALT”指令处，而 WDT 或 LVD 继续运行，其时钟源来自 f_{SUB} 。
- 数据存储器中的内容和寄存器将保持当前值。
- 若 WDT 功能使能，WDT 将被清零并重新开始计数。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入空闲模式 0

进入空闲模式 0 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“1”且 CTRL 寄存器中的 FSYSON 位为“0”。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟停止运行，应用程序停止在“HALT”指令处， f_{SUB} 时钟将继续运行。
- 数据存储器中的内容和寄存器将保持当前值。
- 若 WDT 功能使能，WDT 将被清零并重新开始计数。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

进入空闲模式 1

进入空闲模式 1 的方法仅有一种——应用程序中执行“HALT”指令前需设置寄存器 SMOD 中 IDLEN 位为“1”且 CTRL 寄存器中的 FSYSON 位为“1”。在上述条件下执行该指令后，将发生的情况如下：

- 系统时钟和 f_{SUB} 时钟开启，应用程序停止在“HALT”指令处。
- 数据存储器中的内容和寄存器将保持当前值。
- 若 WDT 功能使能，WDT 将被清零并重新开始计数。
- 输入 / 输出口将保持当前值。
- 状态寄存器中暂停标志 PDF 将被置起，看门狗溢出标志 TO 将被清除。

待机电流注意事项

由于单片机进入休眠或空闲模式的主要原因是将单片机的电流降低到尽可能低，可能到只有几个微安的级别（空闲模式 1 除外），所以如果要将电路的电流进一步降低，电路设计者还应有其它的考虑。应该特别注意的是单片机的输入/输出引脚。所有高阻抗输入脚都必须连接到固定的高或低电平，因为引脚浮空会造成内部振荡并导致耗电增加。这也应用于有不同封装的单片机，因为它们可能含有未引出的引脚，这些引脚也必须设为输出或带有上拉电阻的输入。

另外还需注意单片机设为输出的 I/O 引脚上的负载。应将它们设置在有最小拉电流的状态或将它们和其它的 CMOS 输入一样接到没有拉电流的外部电路上。在空闲模式 1 中，系统时钟开启。若系统时钟来自高速系统振荡器，额外的待机电流也可能会有几百微安。

唤醒

单片机进入休眠模式或空闲模式后，系统时钟将停止以降低功耗。然而单片机再次唤醒，原来的系统时钟重新起振、稳定且恢复正常工作需要一定的时间。

系统进入休眠或空闲模式之后，可以通过以下几种方式唤醒：

- PA 口下降沿
- 系统中断
- WDT 溢出

单片机执行 HALT 指令，系统进入休眠或空闲模式，PDF 将被置位。系统上电或执行清除看门狗的指令，PDF 将被清零。

若系统由看门狗定时器溢出唤醒，会发生看门狗定时器复位，TO 将被置位。看门狗定时器溢出将会置位 TO 标志并唤醒系统，这种复位会重置程序计数器和堆栈指针，其它标志保持原有状态。

PA 口中的每个引脚都可以通过 PAWU 寄存器使能下降沿唤醒功能。PA 端口唤醒后，程序将在“HALT”指令后继续执行。如果系统是通过中断唤醒，则有两种可能发生。第一种情况是：相关中断除能或是中断使能且堆栈已满，则程序会在“HALT”指令之后继续执行。这种情况下，唤醒系统的中断会等到相关中断使能或有堆栈层可以使用之后才执行。第二种情况是：相关中断使能且堆栈未滿，则中断可以马上执行。如果在进入休眠或空闲模式之前中断标志位已经被设置为“1”，则相关中断的唤醒功能将无效。

看门狗定时器

看门狗定时器的功能在于防止如电磁的干扰等外部不可控制事件，所造成的程序不正常动作或跳转到未知的地址。

看门狗定时器时钟源

WDT 定时器时钟源来自于内部时钟 f_s 。 f_s 的时钟源由 LIRC 振荡器提供。看门狗定时器的时钟源可分频为 $2^8 \sim 2^{18}$ 以提供更大的溢出周期，分频比由 WDTC 寄存器中的 WS2~WS0 位来决定。LIRC 内部振荡器在 5V 供电下有一个约为 32kHz 的频率，应注意此指定的内部时钟可随 V_{DD} ，温度以及制成工艺的不同而变化。

看门狗定时器控制寄存器

WDTC 寄存器用于控制 WDT 功能的使能 / 除能及选择溢出周期。

• WDTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0**: WDT 功能使能 / 除能控制位

10101: 除能
01010: 使能
其它值: MCU 复位

如果由于不利的环境因素或通过软件设置使这些位发生改变，则单片机将在一段延迟时间 t_{SRESET} 后复位，且 CTRL 寄存器中的 WRF 位将置高。

Bit 2~0 **WS2~WS0**: WDT 溢出周期选择位

000: $2^8/f_s$
001: $2^{10}/f_s$
010: $2^{12}/f_s$
011: $2^{14}/f_s$
100: $2^{15}/f_s$
101: $2^{16}/f_s$
110: $2^{17}/f_s$
111: $2^{18}/f_s$

这三位控制 WDT 时钟源的分频比，从而实现对 WDT 溢出周期的控制。

• CTRL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

“x”: 未知

Bit 7 **FSYSON**: 空闲模式下 f_{sys} 控制位

详见其它章节

Bit 6~3 未定义，读为 “0”

Bit 2 **LVRF**: LVR 复位标志位

详见其它章节

- Bit 1 **LRF**: LVRC 控制寄存器软件复位标志位
详见其它章节
- Bit 0 **WRF**: WDTC 控制寄存器软件复位标志位
0: 未发生
1: 发生
当 WDTC 控制寄存器发生复位时, 此位置为“1”, 通过应用程序清零。应注意的是该位只能通过应用程序清零。

看门狗定时器操作

当 WDT 溢出时, 它产生一个芯片复位的动作。这也就意味着正常工作期间, 用户需在应用程序中看门狗溢出前有策略地清除看门狗定时器以防止其产生复位, 可使用清除看门狗指令实现。无论什么原因, 程序失常跳转到一个未知的地址或进入一个死循环, 这些清除指令都不能被正确执行, 此种情况下, 看门狗将溢出以使单片机复位。看门狗定时器控制寄存器 WDTC 中有五位 WE4~WE0 可提供使能/除能控制以及看门狗定时器复位操作。当 WE4~WE0 位段被设置为“10101B”时 WDT 将除能, 而 WE4~WE0 位段的值等于“01010B”则 WDT 使能。若由于环境噪声或软件设置使得 WE4~WE0 位段得到除“01010B”和“10101B”以外的值, 单片机将在一段延迟时间 t_{SRESET} 后复位。上电后这些位初始值为“01010B”。

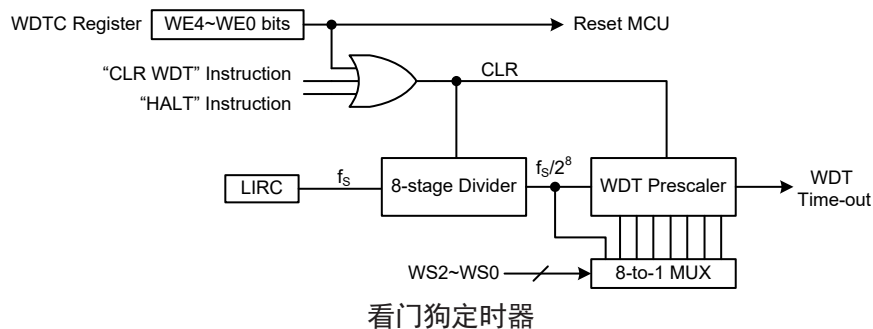
WE4~WE0 位	WDT 功能
10101B	除能
01010B	使能
其它值	复位单片机

看门狗定时器使能/除能控制

程序正常运行时, WDT 溢出将导致芯片复位, 并置位状态标志位 TO。若系统处于休眠或空闲模式, 当 WDT 发生溢出时, 状态寄存器中的 TO 被置位, 程序计数器 PC 和堆栈指针 SP 将被复位。有三种方法可以用来清除 WDT 的内容。第一种是 WDT 复位, 即写入除 01010B 和 10101B 外的任何值到 WE4~WE0 位, 而第二种是通过 WDT 软件清除指令, 第三种是通过“HALT”指令。

只有一种软件指令用于清除看门狗寄存器。因此只要执行“CLR WDT”便清除 WDT。

当设置分频比为 2^{18} 时, 溢出周期最大。例如, 时钟源为 32kHz LIRC 振荡器, 分频比为 2^{18} 时最大溢出周期约 8s, 分频比为 2^8 时最小溢出周期约 8ms。



复位和初始化

复位功能是任何单片机中基本的部分，使得单片机可以设定一些与外部参数无关的先置条件。最重要的复位条件是在单片机首次上电以后，经过短暂的延迟，内部硬件电路使得单片机处于预期的稳定状态并开始执行第一条程序指令。上电复位以后，在程序执行之前，部分重要的内部寄存器将会被设定为预先设定的状态。程序计数器就是其中之一，它会被清除为零，使得单片机从最低的程序存储器地址开始执行程序。

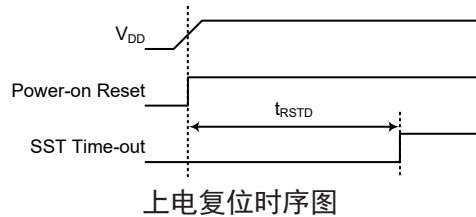
另一种复位为看门狗溢出单片机复位。不同方式的复位操作会对寄存器产生不同的影响。另一种复位为低电压复位即 LVR 复位，在电源供应电压低于一定的阈值时，系统会产生完全复位。

复位功能

通过内部事件触发，此单片机有以下几种复位方式：

上电复位

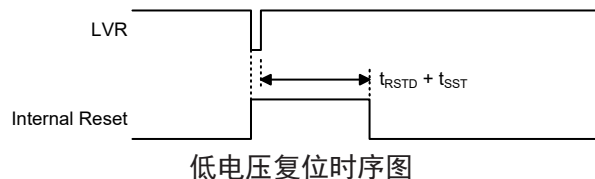
这是最基本且不可避免的复位，发生在单片机初次上电后。除了保证程序存储器从开始地址执行，上电复位也使得其它寄存器被设定在预设条件。所有的输入/输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设定为输入状态。



低电压复位 – LVR

单片机具有低电压复位电路，用来监测它的电源电压。在正常模式和低速模式下，低电压复位功能始终使能于特定电压值，V_{LVR}。例如在更换电池的情况下，单片机供应的电压可能会落在 0.9V~V_{LVR} 的范围内，这时 LVR 将会自动复位单片机且 CTRL 寄存器的 LVRF 标志位会被置位。一个有效的 LVR 信号，即在 0.9V~V_{LVR} 的低电压状态的时间，必须超过 LVD&LVR 电气特性中 t_{LVR} 参数的值。如果低电压存在不超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。

V_{LVR} 参数值可通过 LVRC 寄存器中的 LVS7~LVS0 位设置。若由于受到干扰 LVS7~LVS0 变为除了 LVRC 寄存器定义的某些特定值外的其它值时，需经过一段延迟时间 t_{SRESET} 响应复位。此时 CTRL 寄存器的 LRF 位被置位。上电后 LVRC 寄存器的值为 01010101B。应注意当单片机进入休眠模式或空闲模式时 LVR 功能将自动除能。



• LVRC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS7~LVS0**: LVR 电压选择控制位

01010101: 2.1V

00110011: 2.55V

10011001: 3.15V

10101010: 3.8V

其它值: MCU 复位 – 寄存器复位至 POR 值

若低电压情况发生且满足以上定义的低电压复位值, 则单片机复位。保持低电压状态的时间需超过 t_{LVR} 才能响应复位。此时复位后的寄存器内容保持不变。

除了以上定义的低电压复位值外, 其它值也能导致单片机复位。需要经过一段延迟时间 t_{SRESET} 响应复位。但此时寄存器内容将复位为 POR 值。

• CTRL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

“x”: 未知

Bit 7 **FSYSON**: 空闲模式下 f_{SYS} 控制位

详见其它章节

Bit 6~3 未定义, 读为 “0”

Bit 2 **LVRF**: LVR 复位标志位

0: 未发生

1: 发生

当 LVR 复位情况发生时, 该位被置为 “1”, 此位只能通过应用程序清零。

Bit 1 **LRF**: LVRC 控制寄存器复位标志位

0: 未发生

1: 发生

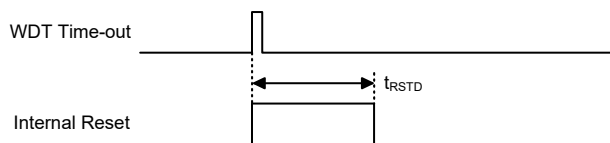
当 LVRC 寄存器包含未定义的 LVR 电压值时, 该位被置为 “1”, 此复位为软件复位功能。该位只能通过应用程序清零。

Bit 0 **WRF**: WDTC 控制寄存器复位标志位

详见其它章节

正常运行时看门狗溢出复位

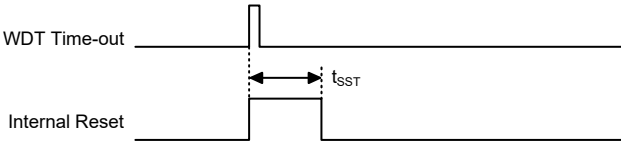
除了看门狗溢出标志位 TO 将被置高外, 正常运行时的看门狗溢出复位与 LVR 复位相同。



正常运行时 WDT 溢出复位时序图

休眠或空闲模式时看门狗溢出复位

休眠或空闲时看门狗溢出复位和其它类型的复位有些不同。除了程序计数器与堆栈指针将被清“0”及 TO 位被设为“1”外，绝大部分的条件保持不变。图中 t_{SST} 的详细说明请参考交流电气特性。



休眠或空闲模式时 WDT 溢出复位时序图

复位初始条件

不同的复位形式以不同的方式影响复位标志位。这些标志位，即 PDF 和 TO 位存放在状态寄存器中，由休眠或空闲模式功能或看门狗计数器等多种控制器操作控制。复位标志位如下所示：

TO	PDF	复位条件
0	0	上电复位
u	u	正常模式或低速模式时的 LVR 复位
1	u	正常模式或低速模式时的 WDT 溢出复位
1	1	空闲或休眠模式时的 WDT 溢出复位

注：“u”表示不改变

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

项目	复位后情况
程序计数器	清除为零
中断	所有中断被除能
看门狗定时器	复位后清零，WDT 重新开始计数
定时器模块	所有定时器模块停止
输入 / 输出口	I/O 口设为输入模式
堆栈指针	堆栈指针指向堆栈顶端

不同的复位形式对单片机内部寄存器的影响是不同的。为保证复位后程序能正常执行，了解寄存器在特定条件复位后的设置是非常重要的。下表即为不同方式复位后内部寄存器的状况。

寄存器	上电复位	WDT 溢出 (正常操作)	WDT 溢出 (休眠 / 空闲)
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu
BP	---- --0	---- --0	---- --u
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBHP	---- -xxx	---- -uuu	---- -uuu
STATUS	--00 xxxx	--1u uuuu	--11 uuuu
SMOD	000- 0011	000- 0011	uuu- uuuu

寄存器	上电复位	WDT 溢出 (正常操作)	WDT 溢出 (休眠 / 空闲)
LVDC	--00 -000	--00 -000	--uu -uuu
INTEG	---- 0000	---- 0000	---- uuuu
INTC0	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	uuuu uuuu
INTC2	0000 0000	0000 0000	uuuu uuuu
MFIO	0000 0000	0000 0000	uuuu uuuu
MF11	0000 0000	0000 0000	uuuu uuuu
MF12	--00 --00	--00 --00	--uu --uu
PA	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	uuuu uuuu
PAPU	0000 0000	0000 0000	uuuu uuuu
PAWU	0000 0000	0000 0000	uuuu uuuu
MF13	--00 --00	--00 --00	--uu --uu
WDTC	0101 0011	0101 0011	uuuu uuuu
TBC	0011 -111	0011 -111	uuuu -uuu
EEA	---0 0000	---0 0000	---u uuuu
EED	0000 0000	0000 0000	uuuu uuuu
EEC	---- 0000	---- 0000	---- uuuu
FADJL	0000 0000	0000 0000	uuuu uuuu
FADJH	---- ---1	---- ---1	---- ---u
CTRL	0--- -x00	0--- -000	u--- -uuu
LVRC	0101 0101	0101 0101	uuuu uuuu
CTM0C0	0000 0000	0000 0000	uuuu uuuu
CTM0C1	0000 0000	0000 0000	uuuu uuuu
CTM0DL	0000 0000	0000 0000	uuuu uuuu
CTM0DH	---- --00	---- --00	---- --uu
CTM0AL	0000 0000	0000 0000	uuuu uuuu
CTM0AH	---- --00	---- --00	---- --uu
STMC0	0000 0000	0000 0000	uuuu uuuu
STMC1	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	uuuu uuuu
STMDH	---- --00	---- --00	---- --uu
STMAL	0000 0000	0000 0000	uuuu uuuu
STMAH	---- --00	---- --00	---- --uu
CTM1C0	0000 0000	0000 0000	uuuu uuuu
CTM1C1	0000 0000	0000 0000	uuuu uuuu
CTM1DL	0000 0000	0000 0000	uuuu uuuu
CTM1DH	---- --00	---- --00	---- --uu
CTM1AL	0000 0000	0000 0000	uuuu uuuu
CTM1AH	---- --00	---- --00	---- --uu
CTM2C0	0000 0000	0000 0000	uuuu uuuu

寄存器	上电复位	WDT 溢出 (正常操作)	WDT 溢出 (休眠 / 空闲)
CTM2C1	0000 0000	0000 0000	uuuu uuuu
CTM2DL	0000 0000	0000 0000	uuuu uuuu
CTM2DH	---- --00	---- --00	---- --uu
CTM2AL	0000 0000	0000 0000	uuuu uuuu
CTM2AH	---- --00	---- --00	---- --uu
PC	---- --11	---- --11	---- --uu
PCC	---- --11	---- --11	---- --uu
PCPU	---- --00	---- --00	---- --uu
CTRL2	10-- --00	10-- --00	uu-- --uu
CTRL3	0000 0000	0000 0000	uuuu uuuu
CTRL4	0000 0000	0000 0000	uuuu uuuu
CTRL5	---- -000	---- -000	---- -uuu
PB	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	uuuu uuuu
PBPU	0000 0000	0000 0000	uuuu uuuu
SLEWC0	---- --00	-----00	---- --uu
SLEDC0	0000 0000	0000 0000	uuuu uuuu
SLEDC1	---- --00	---- --00	---- --uu
PTMC0	0000 0---	0000 0---	uuuu u---
PTMC1	0000 0000	0000 0000	uuuu uuuu
PTMDL	0000 0000	0000 0000	uuuu uuuu
PTMDH	---- --00	---- --00	---- --uu
PTMAL	0000 0000	0000 0000	uuuu uuuu
PTMAH	---- --00	---- --00	---- --uu
PTMRPL	0000 0000	0000 0000	uuuu uuuu
PTMRPH	---- --00	---- --00	---- --uu
OCVPC0	1100 1000	1100 1000	uuuu uuuu
OCVPC1	0000 --00	0000 --00	uuuu --uu
OCVPC2	-000 0000	-000 0000	-uuu uuuu
OCVPDA	0000 0000	0000 0000	uuuu uuuu
OCVPOCAL	0010 0000	0010 0000	uuuu uuuu
OCVPCCAL	0001 0000	0001 0000	uuuu uuuu
TKTMR	0000 0000	0000 0000	uuuu uuuu
TKC0	-000 0-00	-000 0-00	-uuu u-uu
TK16DL	0000 0000	0000 0000	uuuu uuuu
TK16DH	0000 0000	0000 0000	uuuu uuuu
TKC1	---- --11	---- --11	---- --uu
TKM016DL	0000 0000	0000 0000	uuuu uuuu
TKM016DH	0000 0000	0000 0000	uuuu uuuu
TKM0ROL	0000 0000	0000 0000	uuuu uuuu
TKM0ROH	---- --00	---- --00	---- --uu

寄存器	上电复位	WDT 溢出 (正常操作)	WDT 溢出 (休眠 / 空闲)
TKM0C0	0000 0000	0000 0000	uuuu uuuu
TKM0C1	0-00 0000	0-00 0000	u-uu uuuu
SADOL	xxxx ----	xxxx ----	uuuu ---- (ADRF5=0)
			uuuu uuuu (ADRF5=1)
SADOH	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRF5=0)
			---- uuuu (ADRF5=1)
SADC0	0000 0000	0000 0000	uuuu uuuu
SADC1	000- -000	000- -000	uuu- -uuu
ACERL	---- 0000	---- 0000	---- uuuu

注：“-”表示未定义
“u”表示未改变
“x”表示未知

输入 / 输出端口

Holtek 单片机的输入 / 输出控制具有很大的灵活性。大部分引脚可在用户程序控制下被设定为输入或输出。所有引脚的上拉电阻设置以及指定引脚的唤醒设置也都由软件控制，这些特性也使得此类单片机在广泛应用上都能符合开发的需求。

该单片机提供 PA~PC 双向输入 / 输出。这些 I/O 端口映射到 RAM 数据存储器中的特定地址上，如特殊功能数据存储器表所示。所有 I/O 口可用于输入输出操作。作为输入操作，输入引脚无锁存功能，也就是说输入数据必须在执行“MOV A, [m]”时 T2 的上升沿准备好，m 为端口地址。对于输出操作，所有数据都是被锁存的，且保持不变直到输出锁存被重写。

寄存器 名称	位							
	7	6	5	4	3	2	1	0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PC	—	—	—	—	—	—	PC1	PC0
PCC	—	—	—	—	—	—	PCC1	PCC0
PCPU	—	—	—	—	—	—	PCPU1	PCPU0

I/O 逻辑功能寄存器列表

上拉电阻

许多产品应用在端口处于输入状态时，通常需要外加一个上拉电阻来实现上拉的功能。为了免去外部上拉电阻，当引脚配置为输入时，可由内部连接到一个上拉电阻。这些上拉电阻可通过寄存器 PAPU~PCPU 来设置，它用一个 PMOS 晶体管来实现上拉电阻功能。

应注意只有在 I/O 端口被设置为数字输入或 NMOS 输出时，可通过寄存器 PAPU~PCPU 使能内部上拉功能。其它情况下，内部上拉功能除能。

• PAPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **PAPU7~PAPU0**: PA 引脚上拉控制位
 0: 除能
 1: 使能

• PBPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **PBPU7~PBPU0**: PB 引脚上拉控制位
 0: 除能
 1: 使能

• PCPU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PCPU1	PCPU0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **PCPU1~PCPU0**: PC 引脚上拉控制位
 0: 除能
 1: 使能

PA 口唤醒

当使用暂停指令“HALT”迫使单片机进入休眠或空闲模式，单片机的系统时钟将会停止以降低功耗，此功能对于电池及低功耗应用很重要。唤醒单片机有很多种方法，其中之一就是使 PA 口的其中一个引脚从高电平转为低电平。这个功能特别适合于通过外部开关来唤醒的应用。PA 口的每个引脚可以通过设置 PAWU 寄存器来单独选择是否具有唤醒功能。

应注意只有在 PA 引脚被配置为普通 I/O 口且单片机处于 HALT 状态下，PA 端口唤醒功能可通过 PAWU 寄存器中的相关位来使能。在其它情况下，唤醒功能除能。

• PAWU 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **PAWU7~PAWU0**: PA 引脚唤醒控制位

0: 除能

1: 使能

输入 / 输出端口控制寄存器

每一个输入 / 输出都具有各自的控制寄存器，即 PAC~PCC，用来控制输入 / 输出配置。从而每个 I/O 引脚都可以在软件控制下，动态地设置为 CMOS 输出或输入。I/O 端口的每个引脚都映射到其相关端口控制寄存器中的一个位。若 I/O 引脚要实现输入功能，则对应的控制寄存器的位需要设置为“1”。这时程序指令可以直接读取输入脚的逻辑状态。若控制寄存器相应的位被设定为“0”，则此引脚被设置为 CMOS 输出。当引脚设置为输出状态时，程序指令读取的是输出端口寄存器的内容。但应注意，如果对输出口做读取动作时，程序读取到的是内部输出数据锁存器中的状态，而不是输出引脚上实际的逻辑状态。

• PAC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 **PAC7~PAC0**: PA 引脚输入 / 输出类型选择位

0: 输出

1: 输入

• PBC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 **PBC7~PBC0**: PB 引脚输入 / 输出类型选择位

0: 输出

1: 输入

• PCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PCC1	PCC0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

Bit 7~2 未定义，读为“0”

Bit 1~0 **PCC1~PCC0**: PC 引脚输入 / 输出类型选择位

0: 输出

1: 输入

转换速率控制

通过 SLEWC0 寄存器，可设置 PB5 端口的转换速率。具体可参考前面的转换速率控制特性表格。

• SLEWC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	SLEWC01	SLEWC00
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **SLEWC01~SLEWC00**: PB5 输出转换速率选择

00: 转换速率 = Level 0

01: 转换速率 = Level 1

10: 转换速率 = Level 2

11: 转换速率 = Level 3

注：用户可参考转换速率控制特性章节针对不同应用设置所需的端口的输出转换速率。

源电流选择

该单片机的每个 I/O 口都支持不同的源电流驱动能力，通过相应的源电流选择位控制。仅当对应的引脚被设为 CMOS 输出时，其源电流选择位才有效。否则，这些选择位无效。用户可参考交流电气特性章节为不同应用选择所需的源电流。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SLEDC0	SLEDC07	SLEDC06	SLEDC05	SLEDC04	SLEDC03	SLEDC02	SLEDC01	SLEDC00
SLEDC1	—	—	—	—	—	—	SLEDC11	SLEDC10

I/O 口源电流选择寄存器列表

• SLEDC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SLEDC07	SLEDC06	SLEDC05	SLEDC04	SLEDC03	SLEDC02	SLEDC01	SLEDC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **SLEDC07~SLEDC06**: PB7、PB6、PB4 源电流选择位

00: Level 0 (最小)

01: Level 1

10: Level 2

11: Level 3 (最大)

Bit 5~4 **SLEDC05~SLEDC04**: PB3~PB0 源电流选择位

00: Level 0 (最小)

01: Level 1

10: Level 2

11: Level 3 (最大)

Bit 3~2 **SLEDC03~SLEDC02**: PA7~PA4 源电流选择位

00: Level 0 (最小)

01: Level 1

10: Level 2

11: Level 3 (最大)

Bit 1~0 **SLEDC01~SLEDC00:** PA3~PA0 源电流选择位
 00: Level 0 (最小)
 01: Level 1
 10: Level 2
 11: Level 3 (最大)

● SLEDC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	SLEDC11	SLEDC10
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **SLEDC11~SLEDC10:** PC1、PC0 源电流选择位
 00: Level 0 (最小)
 01: Level 1
 10: Level 2
 11: Level 3 (最大)

引脚共用功能

引脚的多功能可以增加单片机应用的灵活性。有限的引脚个数将会限制设计者，而引脚的多功能将会解决很多此类问题。此外，这些引脚功能可以通过应用程序控制一系列寄存器进行设定。

引脚共用功能选择寄存器

封装中有限的引脚个数会对单片机某些功能造成影响。然而，引脚功能共用功能通过引脚功能选择，使得小封装单片机具有更多不同的功能。该单片机包含 CTRL3、CTRL4 和 CTRL5 寄存器，用于选择多功能共用引脚上的所需功能。

要注意的最重要一点是，确保所需的引脚共用功能被正确地选择和取消。对于大部分引脚共用功能，要选择所需的引脚共用功能，首先应通过相应的引脚共用控制寄存器正确地选择该功能，然后再配置相应的外围功能设置以使能外围功能。但是，在设置相关引脚控制字段时，一些数字输入引脚如 INT、xTCKn 等，与对应的通用 I/O 口共用同一个引脚共用设置选项。要选择这些引脚功能，除了上述的必要的引脚共用控制和外围功能设置外，还必须将其对应的端口控制寄存器位设置为输入。要正确地取消引脚共用功能，首先应除能外围功能，然后再修改相应的引脚共用控制寄存器以选择其它的共用功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
CTRL3	IOCN7	IOCN6	IOCN5	IOCN4	IOCN3	IOCN2	IOCN1	IOCN0
CTRL4	IOCN15	IOCN14	IOCN13	IOCN12	IOCN11	IOCN10	IOCN9	IOCN8
CTRL5	—	—	—	—	—	IOCN18	IOCN17	IOCN16

引脚共用功能选择寄存器列表

• CTRL3 寄存器

Bit	7	6	5	4	3	2	1	0
Name	IOCN7	IOCN6	IOCN5	IOCN4	IOCN3	IOCN2	IOCN1	IOCN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **IOCN7:** PA7 引脚共用功能选择
0: PA7/PTCK
1: KEY1
- Bit 6 **IOCN6:** PA6 引脚共用功能选择
0: PA6
1: CTP1
- Bit 5 **IOCN5:** PA5 引脚共用功能选择
0: PA5/INT1/STCK
1: OCVPCI
- Bit 4~3 **IOCN4~IOCN3:** PA4 引脚共用功能选择
00: PA4
01: STP
10: KEY2
11: KEY2
- Bit 2 **IOCN2:** PA2 引脚共用功能选择
0: PA2
1: CTP2
- Bit 1~0 **IOCN1~IOCN0:** PA1 引脚共用功能选择
00: PA1
01: CTP0
10: KEY3
11: KEY3

• CTRL4 寄存器

Bit	7	6	5	4	3	2	1	0
Name	IOCN15	IOCN14	IOCN13	IOCN12	IOCN11	IOCN10	IOCN9	IOCN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~6 **IOCN15~IOCN14:** PB3 引脚共用功能选择
00: PB3
01: PB3
10: OCVPAI0
11: OCVPAI0

仅当 ACERL 寄存器中的 ACE3 位为 0 时，这两个位选择才有效。如果 ACE3 位为 1，则 PB3 引脚则总是作为 A/D 转换器 AN3 输入功能。

- Bit 5~4 **IOCN13~IOCN12**: PB2 引脚共用功能选择
 00: PB2
 01: PB2
 10: OCVPAI0
 11: OCVPAI0
 仅当 ACERL 寄存器中的 ACE2 位为 0 时，这两个位选择才有效。如果 ACE2 位为 1，则 PB2 引脚则总是作为 A/D 转换器 AN2 输入功能。
- Bit 3~2 **IOCN11~IOCN10**: PB1 引脚共用功能选择
 00: PB1
 01: PB1
 10: OCVPAI0
 11: OCVPAI0
 仅当 ACERL 寄存器中的 ACE1 位为 0 时，这两个位选择才有效。如果 ACE1 位为 1，则 PB1 引脚则总是作为 A/D 转换器 AN1 输入功能。
- Bit 1~0 **IOCN9~IOCN8**: PB0 引脚共用功能选择
 00: PB0
 01: PB0
 10: OCVPAI0
 11: OCVPAI0
 仅当 ACERL 寄存器中的 ACE0 位为 0 时，这两个位选择才有效。如果 ACE0 位为 1，则 PB0 引脚则总是作为 A/D 转换器 AN0 输入功能。

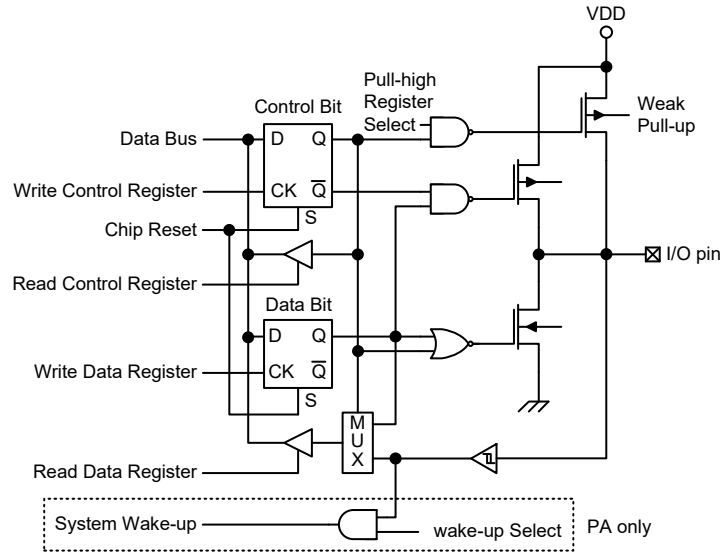
• CTRL5 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	IOCN18	IOCN17	IOCN16
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

- Bit 7~3 未定义，读为“0”
- Bit 2 **IOCN18**: PB5 引脚共用功能选择
 0: PB5
 1: PTP
- Bit 1 **IOCN17**: PA3 引脚共用功能选择
 0: PA3/INT0/CTCK0
 1: KEY4
- Bit 0 **IOCN16**: PB4 引脚共用功能选择
 0: PB4/CTCK1
 1: VREF

输入 / 输出引脚结构

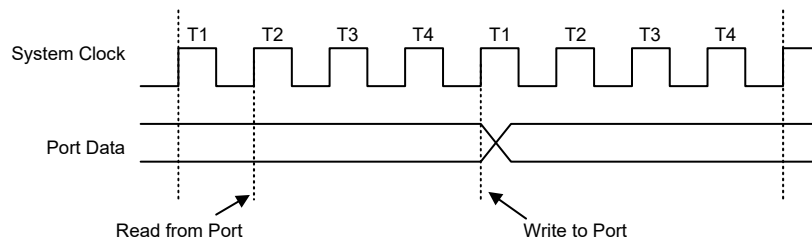
下图为输入 / 输出引脚逻辑功能的内部结构图。输入 / 输出引脚的准确逻辑结构图可能与此图不同，这里只是为了方便对 I/O 引脚逻辑功能的理解提供一个参考。由于存在诸多的引脚共用结构，在此不方便提供所有类型引脚功能结构图。



逻辑功能输入 / 输出端口结构

编程注意事项

在编程中，最先要考虑的是端口的初始化。复位之后，所有的输入 / 输出数据及端口控制寄存器都将被设为逻辑高。所有输入 / 输出引脚默认为输入状态，而其电平则取决于其它相连接电路以及是否选择了上拉电阻。如果端口控制寄存器 PAC~PCC，某些引脚位被设定输出状态，这些输出引脚会有初始高电平输出，除非数据寄存器端口 PA~PC 在程序中被预先设定。设置哪些引脚是输入及哪些引脚是输出，可通过设置正确的值到适当的端口控制寄存器，或使用指令“SET [m].i”及“CLR [m].i”来设定端口控制寄存器中个别的位。注意，当使用这些位控制指令时，系统即将产生一个读 - 修改 - 写的操作。单片机需要先读入整个端口上的数据，修改个别的位，然后重新把这些数据写入到输出端口。



读 / 写时序

PA 口的每个引脚都带唤醒功能。单片机处于休眠或空闲模式时，有很多方法可以唤醒单片机，其中之一就是通过 PA 任一引脚电平从高到低转换的方式，可以设置 PA 口一个或多个引脚具有唤醒功能。

定时器模块 – TM

控制和测量时间在任何单片机中都是一个很重要的部分。该单片机提供几个定时器模块 (简称 TM)，来实现和时间有关的功能。定时器模块是包括多种操作的定时单元，提供的操作有：定时 / 计数器、捕捉输入、比较匹配输出、单脉冲输出以及 PWM 输出等功能。每个定时器模块有两个独立中断。每个 TM 外加的输入输出引脚，扩大了定时器的灵活性，便于用户使用。

这里只介绍各种 TM 的共性，更多详细资料请参考简易型、标准型和周期型定时器章节。

简介

该单片机含有三个 10-bit 简易型 TM，CTM0~CTM2，一个 10-bit 标准型 TM – STM 以及一个 10-bit 周期型 TM – PTM，虽然性质相似，但不同 TM 特性复杂度不同。本章介绍简易型，标准型和周期型 TM 的共性，更多详细资料见后面各相应章节。三种类型 TM 的特性和区别见下表。

TM 功能	CTM	STM	PTM
定时 / 计数器	√	√	√
捕捉输入	—	√	√
比较匹配输出	√	√	√
PWM 通道数	1	1	1
单脉冲输出	—	1	1
PWM 对齐方式	边沿对齐	边沿对齐	边沿对齐
PWM 调节周期 & 占空比	占空比或周期	占空比或周期	占空比或周期

TM 功能概要

CTM	STM	PTM
10-bit CTM0 10-bit CTM1 10-bit CTM2	10-bit STM	10-bit PTM

TM 名称 / 类型参考

TM 操作

这三种不同类型的 TM 提供从简单的定时操作到 PWM 信号产生等多种功能。理解 TM 操作的关键是比较 TM 内独立运行的计数器的值与内部比较器的预置值。当计数器的值与比较器的预置值相同时，则比较匹配，TM 中断信号产生，清零计数器并改变 TM 输出引脚的状态。用户选择内部时钟或外部时钟来驱动内部 TM 计数器。

TM 时钟源

驱动 TM 计数器的时钟源很多。通过设置 xTMn 控制寄存器的 xTnCK2~xTnCK0 位可选择所需的时钟源，其中“x”代表 C、S 或 P 型 TM，“n”为指定 TM 编号。对于 CTM0~CTM2 和 STM，时钟源可来自系统时钟 f_{SYS} 或内部高速时钟 f_{H} 或 f_{SUB} 时钟源或外部 xTCKn 引脚。对于 PTM，支持微调和使能 / 除能控制的专用振荡器可作为 PTM 时钟输入。 f_{SUB} 时钟源或外部 PTCK 引脚输入信号也可作为 PTM 时钟源。外部引脚时钟源用于允许外部信号作为 TM 时钟源或用于事件计数。

TM 中断

简易型、标准型和周期型 TM 各有两个内部中断，即内部比较器 A 和比较器 P，当发生比较匹配时比较器将产生一个 TM 中断。中断产生时将清除计数器，并改变 TM 输出引脚的状态。

TM 外部引脚

无论哪种类型的 TM，都有一个 TM 输入引脚 xTCKn。xTCKn 作为 TM 的时钟源输入脚，通过设置 xTMnC0 寄存器中的 xTnCK2~xTnCK0 位进行选择。外部时钟源可通过 xTCKn 输入引脚来驱动内部 TM。TM 输入引脚可以选择上升沿或下降沿有效。PTM 的单脉冲输出模式中，PTCK 引脚也可用作外部触发输入引脚。

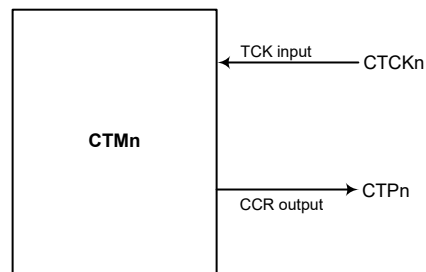
STM 和 PTM 有另一个输入引脚 xTP，可用作 STM 或 PTM 输出引脚。STM 或 PTM 引脚为捕捉输入引脚，其有效边沿可为上升沿，下降沿或双沿。通过配置 xTMC1 寄存器中的 xTIO1~xTIO0 位段选择有效边沿转换类型。对于 PTM，工作于捕捉输入功能时，除 PTP 引脚外 PTCK 也可被选作外部触发源输入引脚。

每个 TM 有一个输出引脚 xTPn。可利用引脚共用功能章节所述的相关引脚共用功能选择位来选择 TM 输出引脚。当 TM 工作在比较匹配输出模式且比较匹配发生时，这些引脚会由 TM 控制切换到高电平或低电平或翻转。外部 xTPn 输出引脚也被 TM 用来产生 PWM 输出波形。

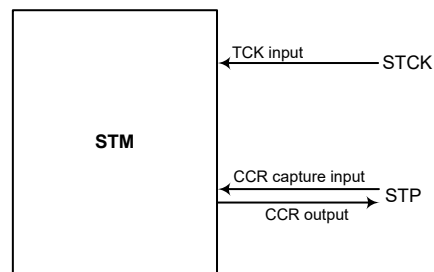
由于 TM 输入或输出引脚与其它功能共用，TM 输出功能需要通过寄存器先被设置。寄存器中的单个位决定了其相关引脚是用作外部 TM 输出引脚还是其它功能。更多引脚共用功能选择详见引脚共用功能章节。

CTMn (n=0~2)		STM		PTM	
输入	输出	输入	输出	输入	输出
CTCKn	CTPn	STCK/STP	STP	PTCK/PTP	PTP

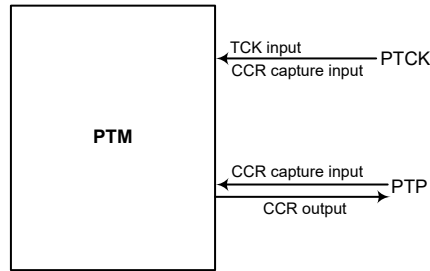
TM 外部引脚



CTMn 功能引脚框图 (n=0~2)



STM 功能引脚框图

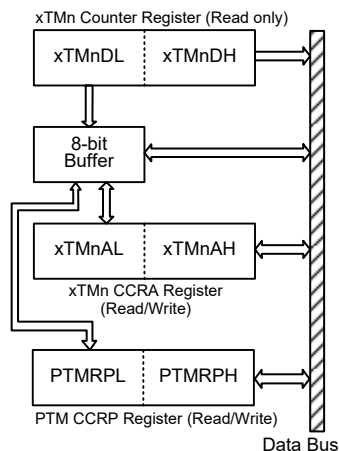


PTM 功能引脚框图

编程注意事项

10-bit 的 TM 计数器寄存器和捕捉 / 比较 CCRA 和 CCRP 寄存器，含有低字节和高字节结构。高字节可直接访问，低字节则仅能通过一个内部 8-bit 的缓存器进行访问。读写这些成对的寄存器需通过特殊的方式。值得注意的是 8-bit 缓存器的存取数据及相关低字节的读写操作仅在其相应的高字节读取操作执行时发生。

由于 CCRA 和 CCRP 寄存器访问方式如下图所示，读写这些成对的寄存器需通过特殊的方式。建议使用“MOV”指令，通过以下步骤访问 CCRA 和 CCRP 低字节寄存器，xTMnAL 或 PTMRPL。否则将导致不可预期的结果。

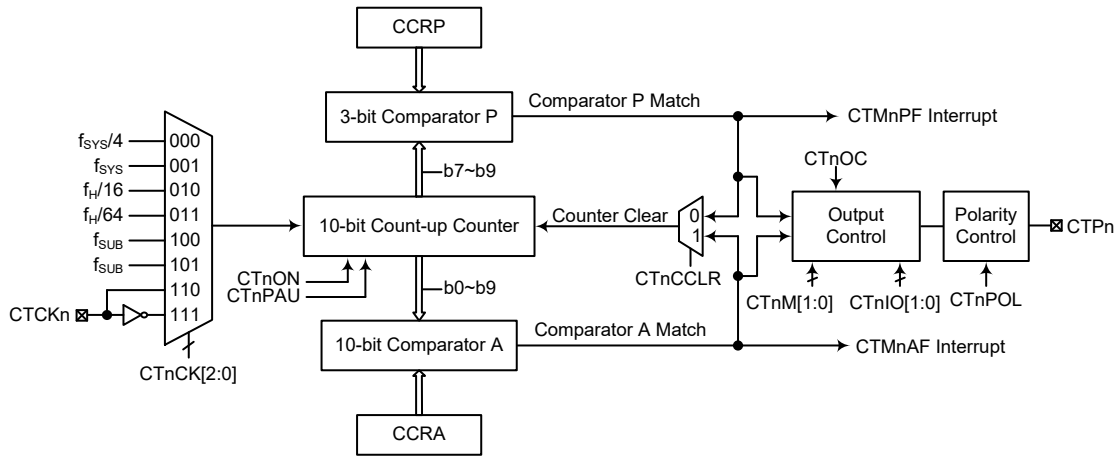


读写流程如下步骤所示：

- 写数据至 CCRA 或 CCRP
 - ◆ 步骤 1. 写数据至低字节寄存器 xTMnAL 或 PTMRPL
 - 注意，此时数据仅写入 8-bit 缓存器。
 - ◆ 步骤 2. 写数据至高字节寄存器 xTMnAH 或 PTMRPH
 - 注意，此时数据直接写入高字节寄存器，同时锁存在 8-bit 缓存器中的数据写入低字节寄存器。
- 由计数器寄存器和 CCRA 或 CCRP 中读取数据
 - ◆ 步骤 1. 从高字节寄存器 xTMnDH、xTMnAH 或 PTMRPH 中读取数据
 - 注意，此时高字节寄存器中的数据直接读取，同时由低字节寄存器读取的数据锁存至 8-bit 缓存器中。
 - ◆ 步骤 2. 从低字节寄存器 xTMnDL、xTMnAL 或 PTMRPL 中读取数据
 - 注意，此时读取 8-bit 缓存器中的数据。

简易型 TM – CTM

尽管简易型 TM 是三种 TM 类型中最简单的形式，它仍包括三种工作模式、即比较匹配输出、定时 / 事件计数器和 PWM 输出模式。简易型 TM 也可由一个外部输入脚控制并驱动一个外部输出脚。



注：CTMn 外部引脚为多种功能共用引脚，因此在使用 CTMn 功能之前应该合理配置相应引脚共用功能选择寄存器以选择所需的 CTMn 引脚功能。

简易型 TM 方框图 (n=0~2)

简易型 TM 操作

简易型 TM 核心是一个由用户选择的内部或外部时钟源驱动的 10-bit 向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 是 3 位的，与计数器的高 3 位比较；而 CCRA 是 10 位的，与计数器的所有位比较。

通过应用程序改变 10-bit 计数器值的唯一方法是使 CTnON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况下会产生 CTM 中断信号。简易型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制两个输出脚。所有工作模式的设定都是通过设置相关内部寄存器来实现的。

简易型 TM 寄存器介绍

简易型 TM 的所有操作由一系列寄存器控制。包含一对只读寄存器用来存放 10-bit 计数器的值，一对读 / 写寄存器存放 10-bit CCRA 的值，剩下两个控制寄存器设置不同的操作和控制模式以及 CCRP 的 3 个位。

寄存器名称	位							
	7	6	5	4	3	2	1	0
CTMnC0	CTnPAU	CTnCK2	CTnCK1	CTnCK0	CTnON	CTnRP2	CTnRP1	CTnRP0
CTMnC1	CTnM1	CTnM0	CTnIO1	CTnIO0	CTnOC	CTnPOL	CTnDPX	CTnCCLR
CTMnDL	D7	D6	D5	D4	D3	D2	D1	D0
CTMnDH	—	—	—	—	—	—	D9	D8
CTMnAL	D7	D6	D5	D4	D3	D2	D1	D0
CTMnAH	—	—	—	—	—	—	D9	D8

10-bit 简易型 TM 寄存器列表 (n=0~2)

• CTMnCO 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTnPAU	CTnCK2	CTnCK1	CTnCK0	CTnON	CTnRP2	CTnRP1	CTnRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CTnPAU**: CTMn 计数器暂停控制位

0: 运行
1: 暂停

通过设置此位为高可暂停计数器。此位清零可恢复正常计数器操作。在暂停情况下 CTM 将保持上电并继续产生功耗。当此位由低到高转换，计数器将保留其剩余值，当其电平再变为低时从此位开始继续计数。

Bit 6~4 **CTnCK2~CTnCK0**: CTMn 计数器时钟选择位

000: $f_{SYS}/4$

001: f_{SYS}

010: $f_H/16$

011: $f_H/64$

100: f_{SUB}

101: f_{SUB}

110: CTCKn 上升沿时钟

111: CTCKn 下降沿时钟

此三位用于选择 CTM 的时钟源。外部引脚时钟源可被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_H 和 f_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **CTnON**: CTMn 计数器 On/Off 控制位

0: Off

1: On

此位控制 CTM 的总开关功能。设置此位为高可使能计数器使其运行，清零此位则除能 CTM。清零此位将停止计数器并关闭 CTM 以减少功耗。当此位经由低到高转换时，内部计数器将复位清零，当此位由高到低转换时，内部计数器将保持其剩余值直到此位再次转变为高。

若 CTM 处于比较匹配输出模式或 PWM 输出模式，当 CTnON 位经由低到高转换时，CTM 输出脚将复位至 CTnOC 位指定的初始值。

Bit 2~0 **CTnRP2~CTnRP0**: CTMn CCRP 3-bit 寄存器，与 CTMn 计数器 bit 9~bit 7 比较器 P 匹配周期

000: 1024 个 CTM 时钟

001: 128 个 CTM 时钟

010: 256 个 CTM 时钟

011: 384 个 CTM 时钟

100: 512 个 CTM 时钟

101: 640 个 CTM 时钟

110: 768 个 CTM 时钟

111: 896 个 CTM 时钟

此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 CTnCCLR 位设定为 0 时，选中该比较结果清除内部计数器。CTnCCLR 位设为 0，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，会使得计数器在最大值溢出。

• CTMnC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTnM1	CTnM0	CTnIO1	CTnIO0	CTnOC	CTnPOL	CTnDPX	CTnCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **CTnM1~CTnM0**: CTMn 工作模式选择位

- 00: 比较匹配输出模式
- 01: 未定义
- 10: PWM 输出模式
- 11: 定时 / 计数器模式

这两个位为 CTM 设置所需的操作模式。为保证操作的可靠性，在此位段发生任何改变之前需先关闭 CTM。在定时 / 计数器模式下，CTM 输出引脚状态未定义。

Bit 5~4 **CTnIO1~CTnIO0**: CTPn 输出功能选择位

比较匹配输出模式

- 00: 未改变
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 输出模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 未定义

定时 / 计数器模式
未使用

这两位用于确定在达到某些条件时 CTM 输出引脚状态应如何改变。这些位的功能选择取决于当下 CTM 的运行模式。

在比较匹配输出模式下，CTnIO1 和 CTnIO0 位决定当比较器 A 比较匹配输出发生时 CTM 输出脚如何改变状态。当比较器 A 比较匹配输出发生时 CTM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时，输出将不会改变。CTM 输出脚的初始值通过 CTMnC1 寄存器中的 CTnOC 位设置取得。注意，由 CTnIO1 和 CTnIO0 位得到的输出电平必须与通过 CTnOC 位设置的初始值不同，否则当比较匹配发生时，CTM 输出脚将不会发生变化。在 CTM 输出脚改变状态后，通过 CTnON 位由低到高电平的转换可复位至初始值。

在 PWM 输出模式，CTnIO1 和 CTnIO0 用于决定比较匹配条件发生时怎样改变 CTM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 CTM 关闭时改变 CTnIO1 和 CTnIO0 位的值是很有必要的。若在 CTM 运行时改变 CTnIO1 和 CTnIO0 的值，PWM 输出的值是无法预料的。

Bit 3 **CTnOC**: CTPn 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 输出模式

- 0: 低有效
- 1: 高有效

此位为 CTM 输出脚的输出控制位。它取决于 CTM 此时正运行于比较匹配输出模式还是 PWM 输出模式。若 CTM 处于定时 / 计数器模式，则其不受影响。在比较匹配输出模式时，比较匹配发生前其决定 CTM 输出脚的逻辑电平值。在 PWM 输出模式时，其决定 PWM 信号是高有效还是低有效。

- Bit2 **CTnPOL**: CTPn 输出极性控制位
0: 同相
1: 反相
此位控制 CTPn 输出脚的极性。此位为高时 CTM 输出脚反相，为低时 CTM 输出脚同相。若 CTM 处于定时 / 计数器模式时其不受影响。
- Bit 1 **CTnDPX**: CTMn PWM 周期 / 占空比控制位
0: CCRP - 周期; CCRA - 占空比
1: CCRP - 占空比; CCRA - 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。
- Bit 0 **CTnCCLR**: 选择 CTMn 计数器清零条件位
0: CTMn 比较器 P 匹配
1: CTMn 比较器 A 匹配
此位用于选择清除计数器的方法。简易型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用于清除内部计数器。CTnCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 的位都被清除为 0 时才能生效。CTnCCLR 位在 PWM 输出模式时未使用。

• CTMnDL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **D7~D0**: CTMn 计数器低字节寄存器 bit 7~bit 0
CTMn 10-bit 计数器 bit 7~bit 0

• CTMnDH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

- Bit 7~2 未定义，读为“0”
- Bit 1~0 **D9~D8**: CTMn 计数器高字节寄存器 bit 1~bit 0
CTMn 10-bit 计数器 bit 9~bit 8

• CTMnAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **D7~D0**: CTMn CCRA 低字节寄存器 bit 7~bit 0
CTMn 10-bit CCRA bit 7~bit 0

● CTMnAH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8:** CTMn CCRA 高字节寄存器 bit 1~bit 0
CTMn 10-bit CCRA bit 9~bit 8

简易型 TM 工作模式

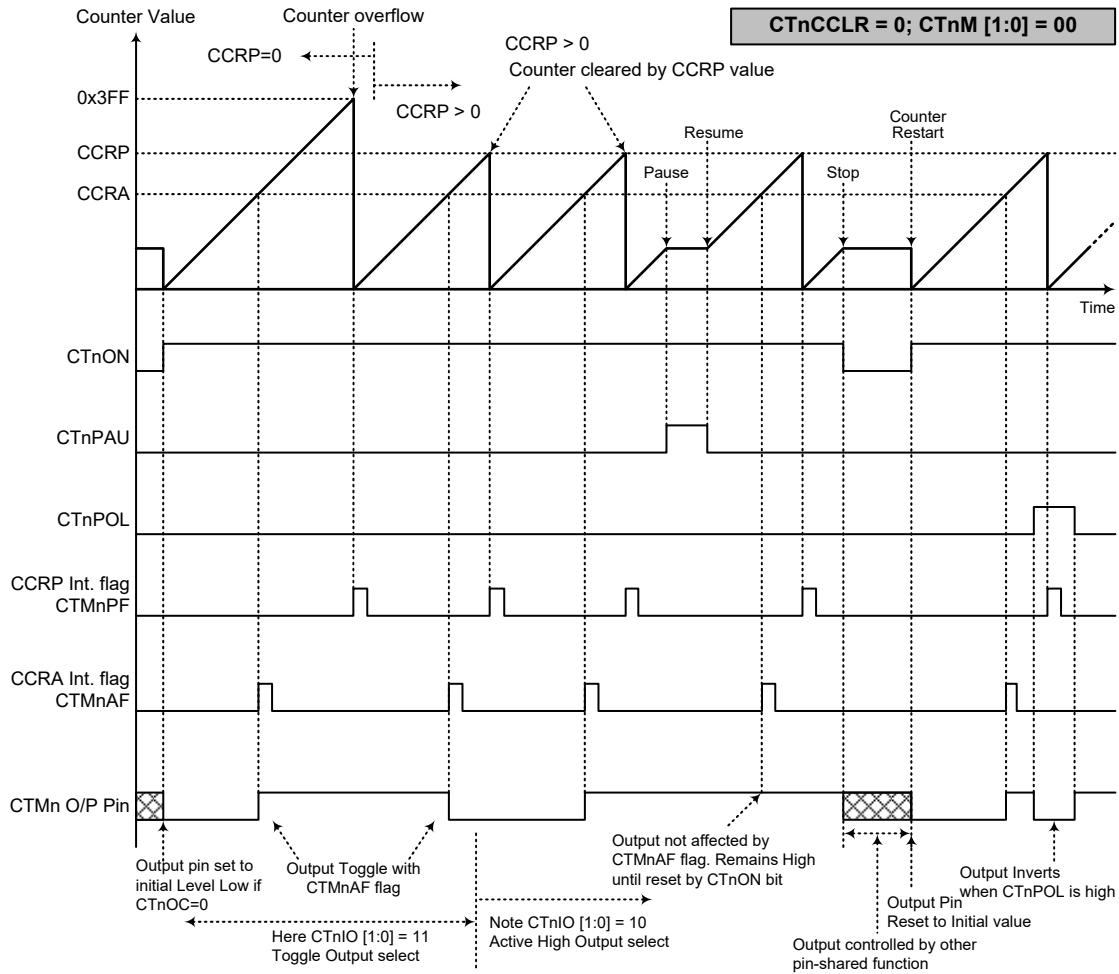
简易型 TM 有三种工作模式，即比较匹配输出模式、PWM 输出模式或定时 / 计数器模式。通过设置 CTMnC1 寄存器的 CTnM1 和 CTnM0 位选择任意工作模式。

比较匹配输出模式

要工作在此模式，CTMnC1 寄存器中的 CTnM1 和 CTnM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 CTnCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 CTMnAF 和 CTMnPF 将分别置起。

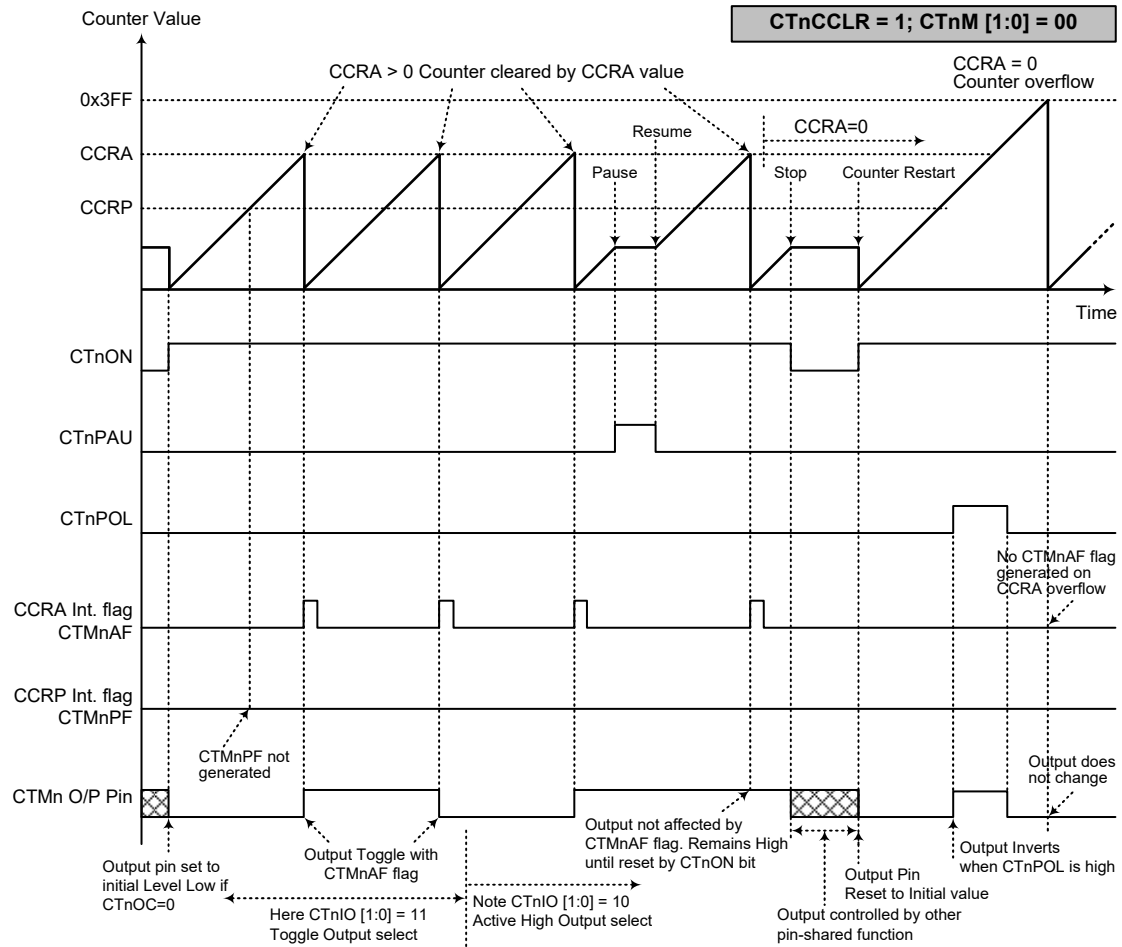
如果 CTMnC1 寄存器的 CTnCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 CTMnAF 中断请求标志产生。所以当 CTnCCLR 为高时，不产生 CTMnPF 中断请求标志。如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 CTMnAF 请求标志。

正如该模式名所言，当比较匹配发生后，CTM 输出脚状态改变。当比较器 A 比较匹配发生后 CTMnAF 标志产生时，CTM 输出脚状态改变。比较器 P 比较匹配发生时产生的 CTMnPF 标志不影响 CTM 输出脚。CTM 输出脚状态改变方式由 CTMnC1 寄存器中 CTnIO1 和 CTnIO0 位决定。当比较器 A 比较匹配发生时，CTnIO1 和 CTnIO0 位决定 CTM 输出脚输出高，低或翻转当前状态。CTM 输出脚初始值，在 CTnON 位由低到高电平的变化后通过 CTnOC 位设置。注意，若 CTnIO1 和 CTnIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – CTnCCLR=0 (n=0~2)

- 注：1. 当 CTnCCLR=0，比较器 P 匹配将清除计数器
2. CTM 输出引脚仅由 CTMnAF 标志位控制
3. 输出引脚通过 CTnON 位上升沿复位至初始值



比较匹配输出模式 – CTnCCLR=1 (n=0~2)

- 注：1. 当 CTnCCLR=1，比较器 A 匹配将清除计数器
2. CTM 输出引脚仅由 CTMnAF 标志位控制
3. 输出引脚通过 CTnON 上升沿复位至初始值
4. 当 CTnCCLR=1，不产生 CTMnPF

定时 / 计数器模式

要工作在此模式，CTMnC1 寄存器中的 CTnM1 和 CTnM0 位需要设置为“11”。定时 / 计数器模式与比较匹配输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 CTMn 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以帮助理解此功能。

PWM 输出模式

要工作在此模式，CTMnC1 寄存器中的 CTnM1 和 CTnM0 位需要设置为“10”。CTMn 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 CTM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可控，其波形的选择就较为灵活。在 PWM 输出模式中，CTnCCLR 位不影响 PWM 操作。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 CTMnC1 寄存器的 CTnDPX 位。所以 PWM 波形频率和占空比由 CCRA 和 CCRP 寄存器共同决定。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。CTMnC1 寄存器中的 CTnOC 位决定 PWM 波形的极性，CTnIO1 和 CTnIO0 位使能 PWM 输出或将 CTM 输出脚置为逻辑高或逻辑低。CTnPOL 位对 PWM 输出波形的极性取反。

● 10-bit CTM，PWM 输出模式，边沿对齐模式，CTnDPX=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

若 $f_{SYS}=8\text{MHz}$ ，CTM 时钟源选择 $f_{SYS}/4$ ，CCRP=100b，CCRA=128，

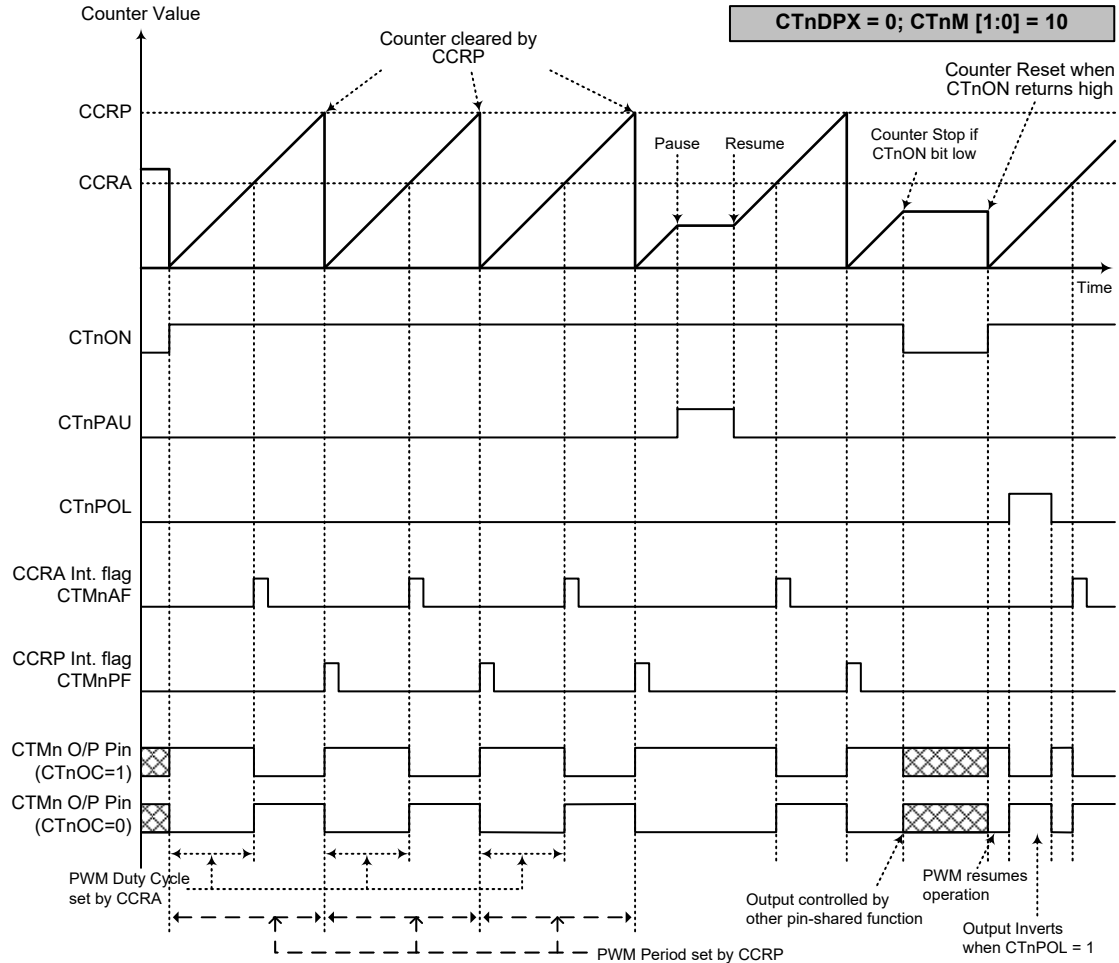
CTM PWM 输出频率 = $(f_{SYS}/4)/512=f_{SYS}/2048=4\text{kHz}$ ，duty=128/512=25%

若由 CCRA 寄存器定义的 Duty 值等于或大于 Period 值，PWM 输出占空比为 100%。

● 10-bit CTM，PWM 输出模式，边沿对齐模式，CTnDPX=1

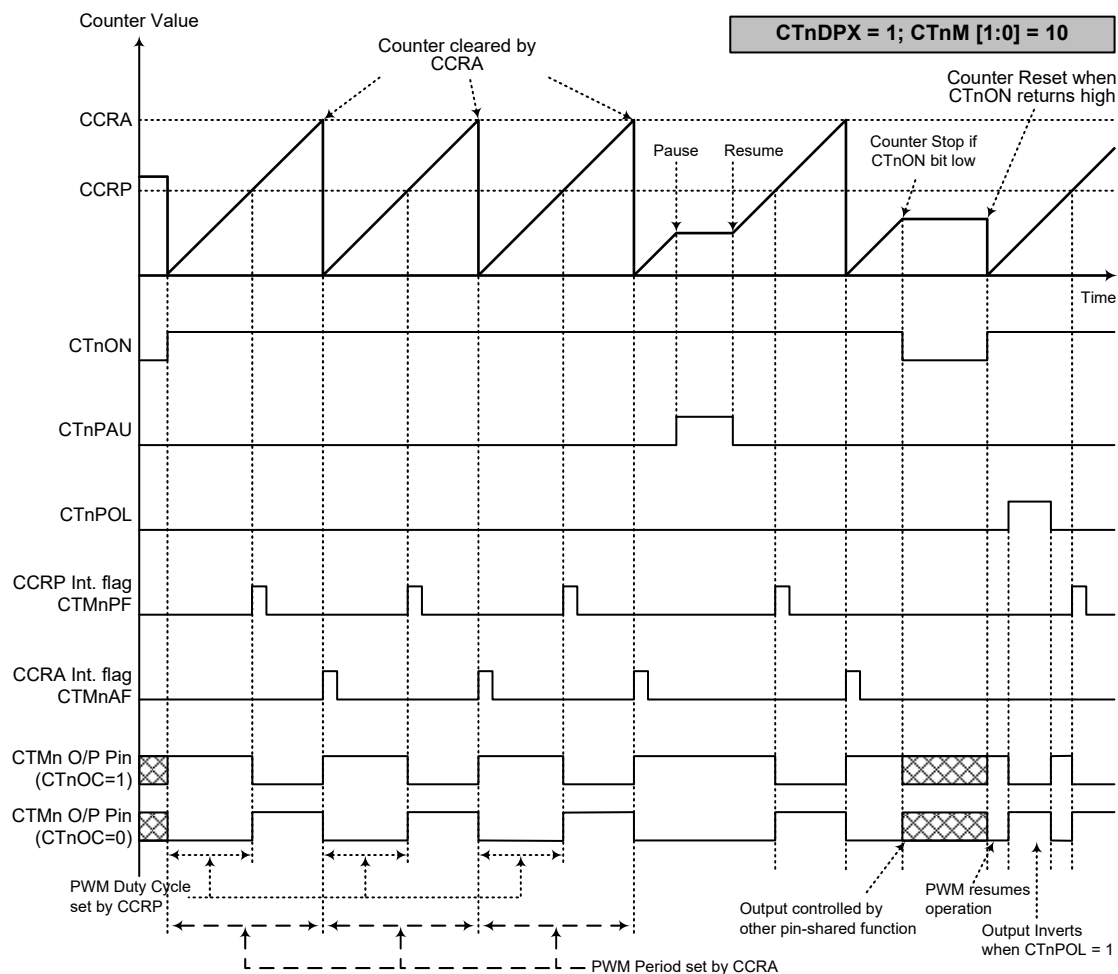
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

PWM 的输出周期由 CCRA 寄存器的值与 CTM 的时钟共同决定，PWM 的占空比由 CCRP 寄存器的值决定。



PWM 输出模式 – CTnDPX=0 (n=0~2)

- 注：1. CTnDPX=0 – 计数器由 CCRP 清除
2. 计数器清除设置 PWM 周期
3. 即使在 CTnIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. CTnCCLR 位对 PWM 操作无影响

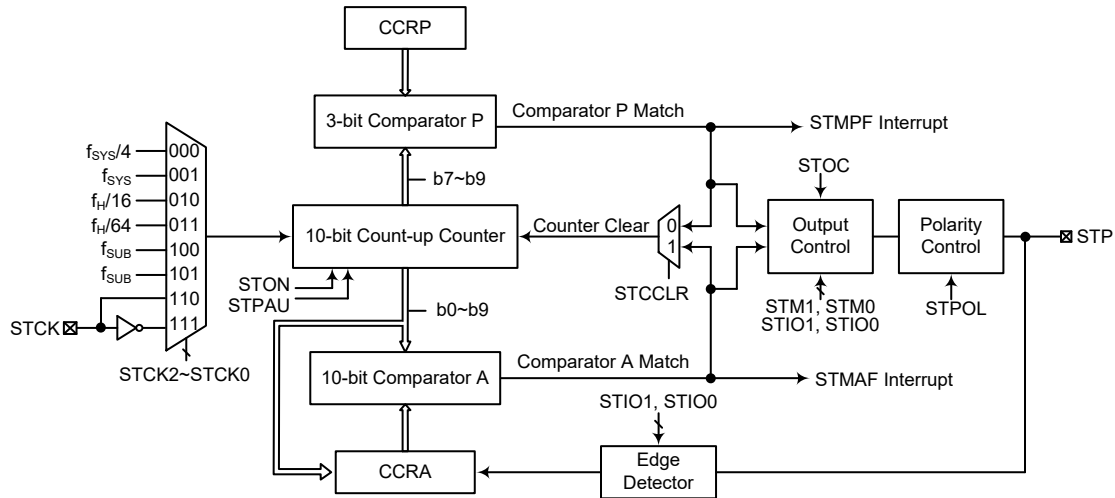


PWM 输出模式 – CTnDPX=1 (n=0~2)

- 注：1. CTnDPX=1 – 计数器由 CCRA 清除
2. 计数器清零设置 PWM 周期
3. 即使在 CTnIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. CTnCCLR 位对 PWM 操作无影响

标准型 TM – STM

标准型 TM 包括 5 种工作模式，即比较匹配输出、定时 / 事件计数器、捕捉输入、单脉冲输出和 PWM 输出模式。标准型 TM 可由一个外部输入脚控制并驱动一个外部输出脚。



注：STM 外部引脚为多种功能共用引脚，因此在使用 STM 功能之前应该合理配置相应引脚共用功能选择寄存器以选择所需的 STM 引脚功能。

标准型 TM 方框图

标准型 TM 操作

标准型 TM 的核心是一个由用户选择的内部或外部时钟源驱动的 10-bit 向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRP 和 CCRA 寄存器中的值进行比较。CCRP 是 3 位宽度，与计数器的高 3 位比较；而 CCRA 是 10 位的，与计数器的所有位比较。

通过应用程序改变 10-bit 计数器值的唯一方法是使 STON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 STM 中断信号。标准型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关寄存器来实现的。

标准型 TM 寄存器介绍

标准型 TM 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 10-bit 计数器的值，一对读 / 写寄存器存放 10-bit CCRA 的值，剩下两个控制寄存器设置不同的操作和控制模式以及 3 位 CCRP 的值。

寄存器名称	位							
	7	6	5	4	3	2	1	0
STMC0	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
STMC1	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
STMDL	D7	D6	D5	D4	D3	D2	D1	D0
STMDH	—	—	—	—	—	—	D9	D8
STMAL	D7	D6	D5	D4	D3	D2	D1	D0
STMAH	—	—	—	—	—	—	D9	D8

10-bit 标准型 TM 寄存器列表

● STMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **STPAU**: STM 计数器暂停控制位

0: 运行

1: 暂停

通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，STM 保持上电状态并继续耗电。当此位由低到高转换时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **STCK2~STCK0**: STM 计数器时钟选择位

000: $f_{SYS}/4$

001: f_{SYS}

010: $f_{H}/16$

011: $f_{H}/64$

100: f_{SUB}

101: f_{SUB}

110: STCK 上升沿时钟

111: STCK 下降沿时钟

此三位用于选择 STM 的时钟源。外部引脚时钟源能被选择在上升沿或下降沿有效。 f_{SYS} 是系统时钟， f_{H} 和 f_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **STON**: STM 计数器 On/Off 控制位

0: Off

1: On

此位控制 STM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 STM。清零此位将停止计数器并关闭 STM 以减少耗电。当此位经由低到高转换时，内部计数器将复位清零，当此位由高到低转换时，内部计数器将保持其剩余值直到此位再次变高。若 STM 处于比较匹配输出模式或 PWM 输出模式或单脉冲输出模式时，当 STON 位经由低到高转换时，STM 输出脚将复位至 STOC 位指定的初始值。

Bit 2~0 **STRP2~STRP0**: STM CCRP 3-bit 寄存器，与 STM 计数器 bit 9~bit 7 比较

比较器 P 匹配周期

000: 1024 个 STM 时钟

001: 128 个 STM 时钟

010: 256 个 STM 时钟

011: 384 个 STM 时钟

100: 512 个 STM 时钟

101: 640 个 STM 时钟

110: 768 个 STM 时钟

111: 896 个 STM 时钟

此三位设定内部 CCRP 3-bit 寄存器的值，然后与内部计数器的高三位进行比较。如果 STCCLR 位设定为 0 时，选中该比较结果清除内部计数器。STCCLR 位设为 0，内部计数器在比较器 P 比较匹配发生时被重置；由于 CCRP 只与计数器高三位比较，比较结果是 128 时钟周期的倍数。CCRP 被清零时，会使得计数器在最大值溢出。

• STMC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **STM1~STM0**: 选择 STM 工作模式位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 输出模式或单脉冲输出模式
- 11: 定时 / 计数器模式

这两位设置 STM 需要的工作模式。为了确保操作可靠, STM 应在 STM1 和 STM0 位有任何改变前先关掉。在定时 / 计数器模式, STM 输出脚状态未定义。

Bit 5~4 **STIO1~STIO0**: 选择 STM 功能位

比较匹配输出模式

- 00: 无变化
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 输出模式 / 单脉冲输出模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 STP 上升沿输入捕捉
- 01: 在 STP 下降沿输入捕捉
- 10: 在 STP 双沿输入捕捉
- 11: 输入捕捉除能

定时 / 计数器模式

未使用

此两位用于决定在一定条件达到时 STM 输出脚如何改变状态。这两位值的选择取决于 STM 运行在何种模式下。

在比较匹配输出模式下, STIO1 和 STIO0 位决定了当比较器 A 比较匹配发生时 STM 输出脚如何改变状态。当比较器 A 比较匹配发生时 STM 输出脚能设为切换高、切换低或翻转当前状态。若此两位同时为 0 时, 这个输出将不会改变。STM 输出脚的初始值通过 STOC 位设置取得。注意, 由 STIO1 和 STIO0 位得到的输出电平必须与通过 STOC 位设置的初始值不同, 否则当比较匹配发生时, STM 输出脚将不会发生变化。在 STM 输出脚改变状态后, 通过 STON 位由低到高电平的转换复位至初始值。

在 PWM 输出模式, STIO1 和 STIO0 用于决定比较匹配条件发生时怎样改变 STM 输出脚的状态。PWM 输出功能通过这两位的变化进行更新。仅在 STM 关闭时改变 STIO1 和 STIO0 位的值是很有必要的。若在 STM 运行时改变 STIO1 和 STIO0 的值, PWM 输出的值是无法预料的。

Bit 3 **STOC**: STM 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 输出模式 / 单脉冲输出模式

- 0: 低有效
- 1: 高有效

这是 STM 输出脚输出控制位。它取决于 STM 此时正运行于比较匹配输出模式还是 PWM 输出模式 / 单脉冲输出模式。若 STM 处于定时 / 计数器模式, 则其不受影响。在比较匹配输出模式时, 比较匹配发生前其决定 STM 输出脚的逻辑

电平值。在 PWM 输出模式时，其决定 PWM 信号是高有效还是低有效。在单脉冲输出模式时，其决定当 STON 位由低转高时 STM 输出脚的逻辑电平值。

- Bit 2 **STPOL**: STM 输出极性控制位
0: 同相
1: 反相
此位控制 STM 输出脚的极性。此位为高时 STM 输出脚反相，为低时 STM 输出脚同相。若 STM 处于定时 / 计数器模式时其不受影响。
- Bit 1 **STDPX**: STM PWM 周期 / 占空比控制位
0: CCRP - 周期; CCRA - 占空比
1: CCRP - 占空比; CCRA - 周期
此位决定 CCRA 与 CCRP 寄存器哪个被用于 PWM 波形的周期和占空比控制。
- Bit 0 **STCCLR**: 选择 STM 计数器清零条件位
0: STM 比较器 P 匹配
1: STM 比较器 A 匹配
此位用于选择清除计数器的方法。标准型 TM 包括两个比较器 - 比较器 A 和比较器 P。这两个比较器每个都可以用作清除内部计数器。STCCLR 位设为高，计数器在比较器 A 比较匹配发生时被清除；此位设为低，计数器在比较器 P 比较匹配发生或计数器溢出时被清除。计数器溢出清除的方法仅在 CCRP 被清除为 0 时才能生效。STCCLR 位在 PWM 输出、单脉冲或输入捕捉模式时未使用。

• STMDL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **D7~D0**: STM 计数器低字节寄存器 bit 7~bit 0
STM 10-bit 计数器 bit 7~bit 0

• STMDH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

- Bit 7~2 未定义，读为“0”
Bit 1~0 **D9~D8**: STM 计数器高字节寄存器 bit 1~bit 0
STM 10-bit 计数器 bit 9~bit 8

• STMAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~0 **D7~D0**: STM CCRA 低字节寄存器 bit 7~bit 0
STM 10-bit CCRA bit 7~bit 0

● STMAH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8**: STM CCRA 高字节寄存器 bit 1~bit 0
STM 10-bit CCRA bit 9~bit 8

标准型 TM 工作模式

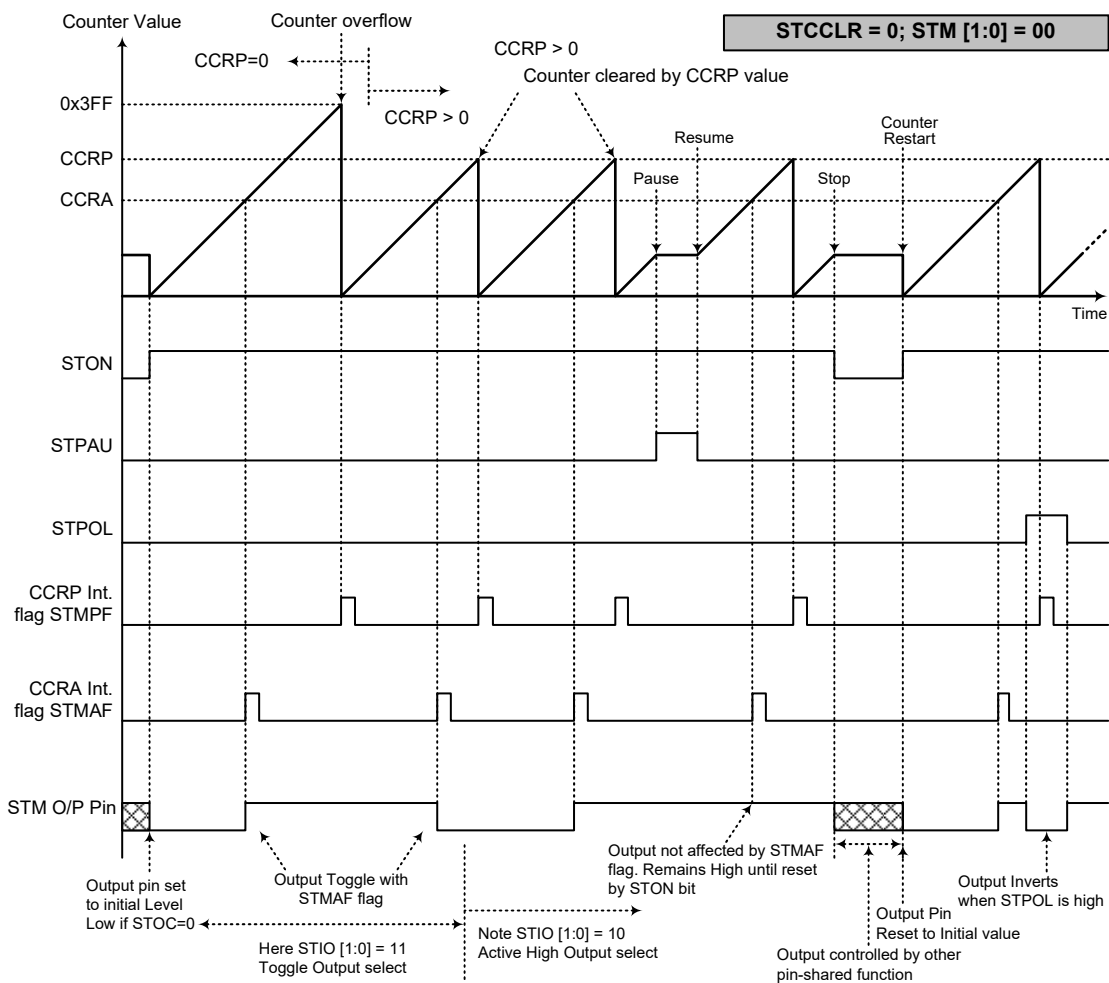
标准型 TM 有五种工作模式，即比较匹配输出模式、PWM 输出模式、单脉冲输出模式、捕捉输入模式或定时 / 计数器模式。通过设置 STMC1 寄存器的 STM1 和 STM0 位选择任意模式。

比较匹配输出模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出、比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 STCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 STMAF 和 STMPF 将分别置位。

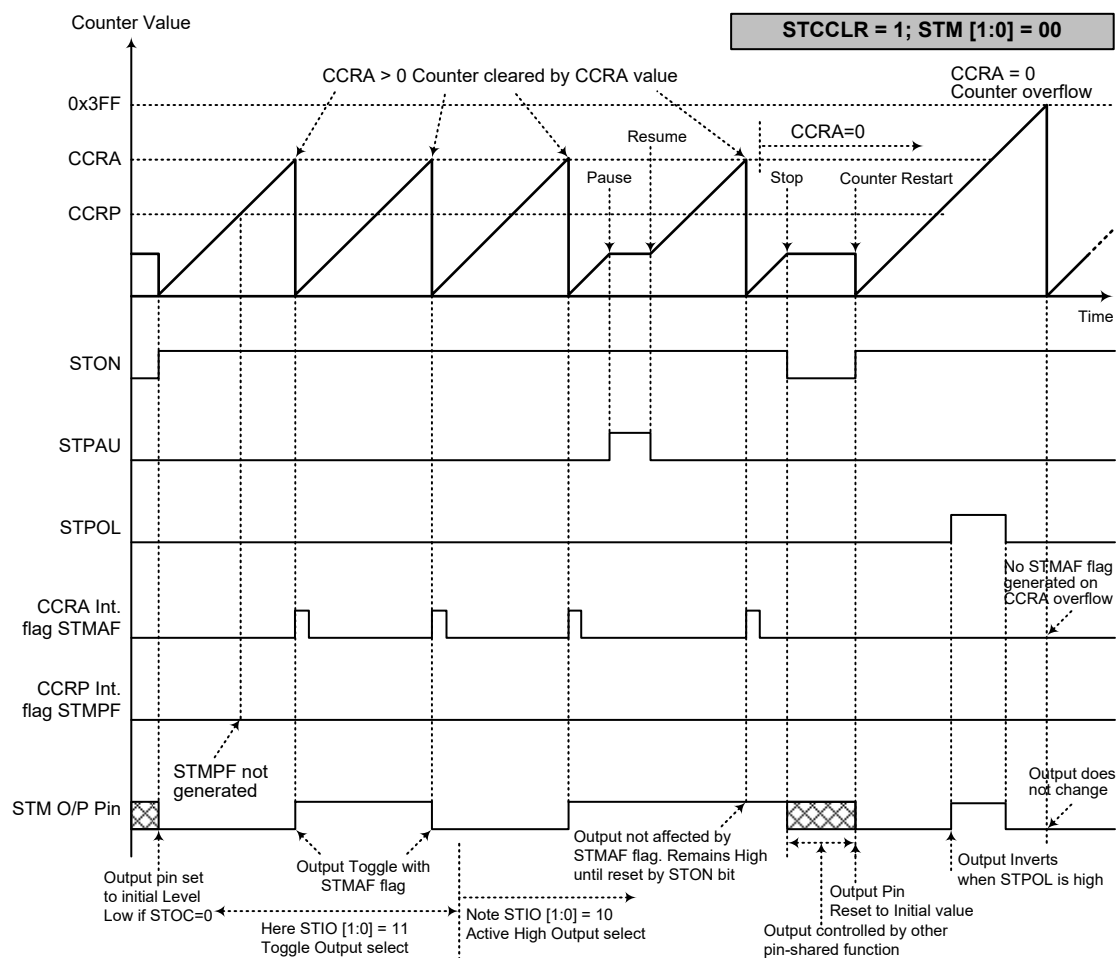
如果 STMC1 寄存器的 STCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅产生 STMAF 中断请求标志。所以当 STCCLR 为高时，不会产生 STMPF 中断请求标志。如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 STMAF 请求标志。

正如该模式名所言，当比较匹配发生后，STM 输出脚状态改变。当比较器 A 比较匹配发生后 STMAF 标志产生时，STM 输出脚状态改变。比较器 P 比较匹配发生时产生的 STMPF 标志不影响 STM 输出脚。STM 输出脚状态改变方式由 STMC1 寄存器中 STIO1 和 STIO0 位决定。当比较器 A 比较匹配发生时，STIO1 和 STIO0 位决定 STM 输出脚输出高，低或翻转当前状态。STM 输出脚初始值，在 STON 位由低到高电平的变化后通过 STOC 位设置。注意，若 STIO1 和 STIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – STCCLR=0

- 注：1. STCCLR=0，比较器 P 匹配将清除计数器
2. STM 输出引脚仅由 STMAF 标志位控制
3. 输出引脚在 STON 上升沿复位至初始值



比较匹配输出模式 – STCCLR=1

- 注：1. STCCLR=1，比较器 A 匹配将清除计数器
2. STM 输出引脚仅由 STMAF 标志位控制
3. 输出引脚在 STON 上升沿复位至初始值
4. 当 STCCLR=1 时，不产生 STMPF 标志位

定时 / 计数器模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 应设置为 11。定时 / 计数器模式操作方式与比较匹配输出模式相同，产生相同的中断标志位。唯一不同的就是在定时 / 计数器模式中未使用 STM 输出引脚。因此，比较匹配输出模式中的描述和时序图可以帮助理解此功能。该模式中未使用的 STM 输出脚用作普通 I/O 脚或其它功能。

PWM 输出模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”，且 STIO1 和 STIO0 位也需要设置为“10”。STM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 STM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 输出模式中，STCCLR 位不影响 PWM 周期。CCRA 和 CCRP 寄存器决定 PWM 波形，一个用来清除内部计数器并控制 PWM 波形的频率，另一个用来控制占空比。哪个寄存器控制频率或占空比取决于 STMC1 寄存器的 STDPX 位。因此，PWM 波形的频率和占空比受 CCRA 和 CCRP 寄存器中的值控制。

当比较器 A 或比较器 P 比较匹配发生时，将产生 CCRA 或 CCRP 中断标志。STMC1 寄存器中的 STOC 位决定 PWM 波形的极性，STIO1 和 STIO0 位使能 PWM 输出或将 STM 输出脚置为逻辑高或逻辑低。STPOL 位对 PWM 输出波形的极性取反。

● 10-bit STM, PWM 输出模式，边沿对齐模式，STDPX=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

若 $f_{sys}=8MHz$ ，TM 时钟源为 $f_{sys}/4$ ，CCRP=100b 且 CCRA=128，

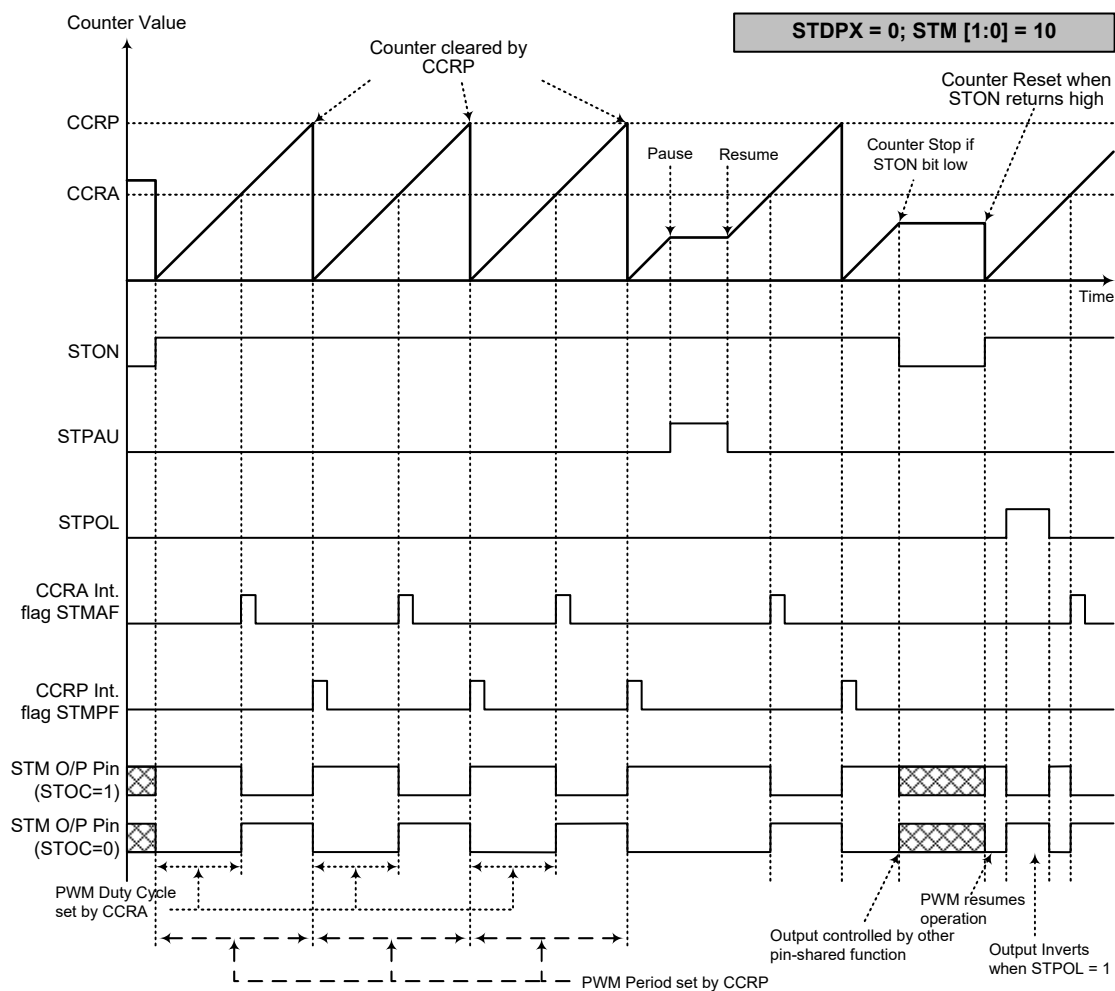
STM PWM 输出频率 = $(f_{sys}/4)/512=f_{sys}/2048=4kHz$ ，周期 = $128/512=25\%$ 。

若 CCRA 所定义的 Duty 值等于或大于 Period 值，则 PWM 输出占空比为 100%。

● 10-bit STM, PWM 输出模式，边沿对齐模式，STDPX=1

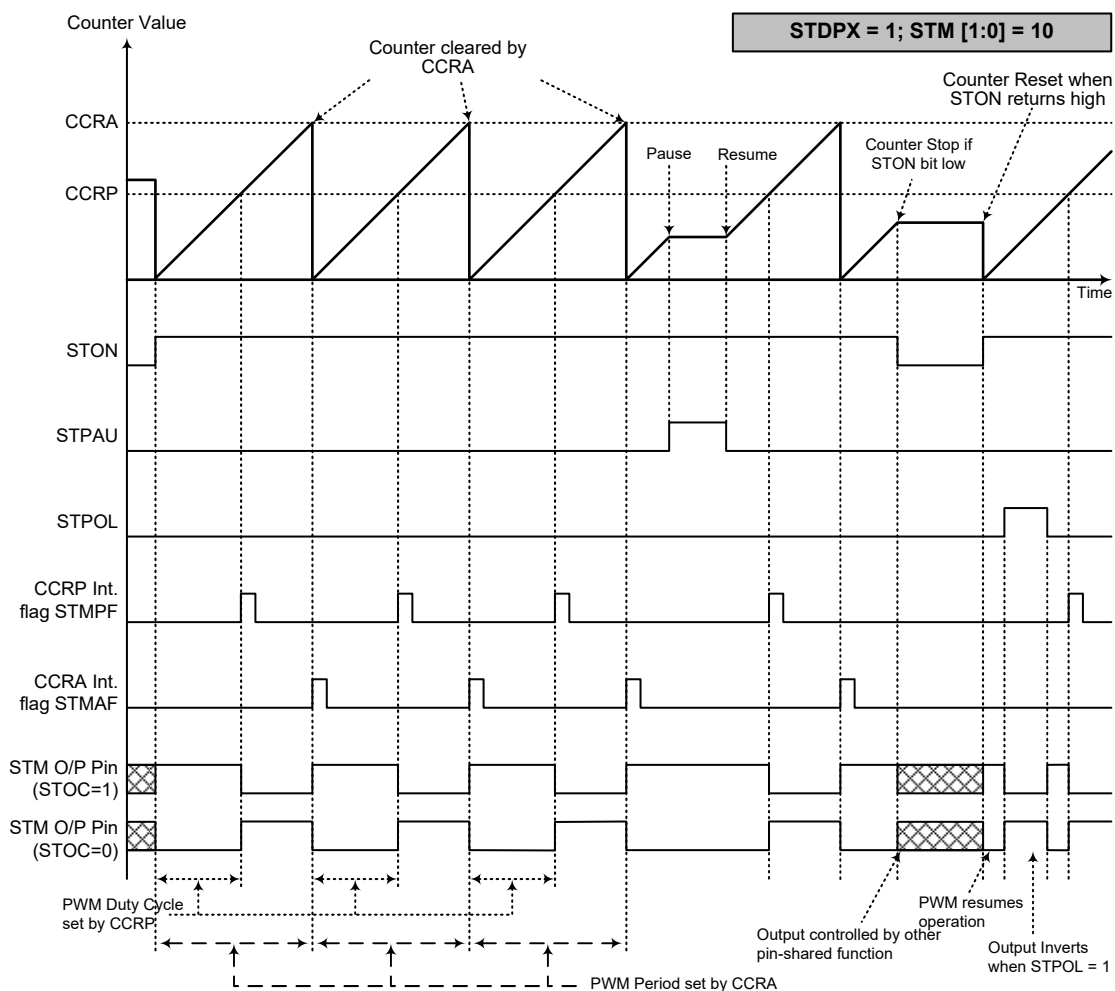
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

PWM 的输出周期由 CCRA 寄存器的值与 STM 的时钟共同决定，PWM 的占空比由 CCRP 的值决定。



PWM 输出模式 – STDPX=0

- 注：1. STDPX=0 – 计数器由 CCRP 清除
2. 计数器清除设置 PWM 周期
3. 即使当 STIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. STCCLR 对 PWM 操作无影响



PWM 输出模式 – STDPX=1

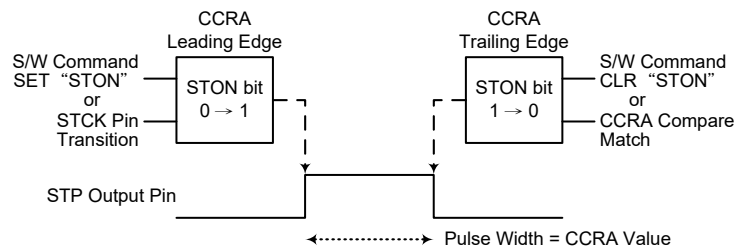
- 注：1. STDPX=1 – 计数器由 CCRA 清除
2. 计数器清除设置 PWM 周期
3. 即使当 STIO[1:0]=00 或 01 时，内部 PWM 功能继续运行
4. STCCLR 对 PWM 操作无影响

单脉冲模式

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“10”，同时 STIO1 和 STIO0 位需要设置为“11”。正如模式名所言，单脉冲输出模式，在 STM 输出脚将产生一个单脉冲输出。

脉冲输出可以通过应用程序控制 STON 位由低到高的转变来触发。而处于单脉冲模式时，STON 位可利用外部 STCK 脚自动由低转变为高，进而开始单脉冲输出。当 STON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 STON 位保持高电平。通过应用程序使 STON 位清零或比较器 A 比较匹配发生时，产生脉冲后沿。

但比较器 A 的比较匹配也会自动清除 STON 位，从而产生单脉冲输出后沿。此时 CCRA 的值可用于控制脉冲宽度。比较器 A 的比较匹配也能产生一个 STM 中断信号。当计数器重新启动，STON 位从低到高转换时，计数器将被复位回 0。在单脉冲模式下，CCRP 寄存器，STCCLR 位和 STDPX 位未使用。



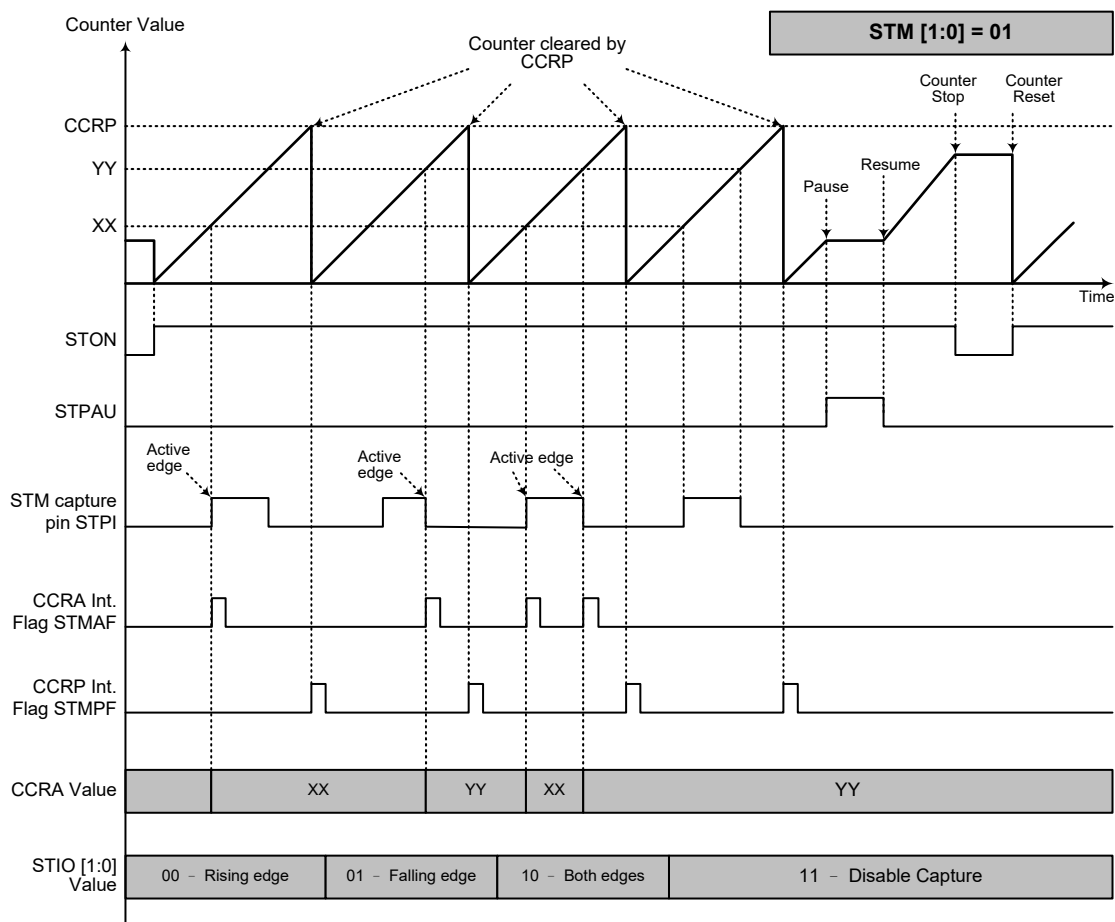
单脉冲产生示意图



- 注: 1. 通过 CCRA 匹配停止计数器
2. CCRP 未使用
3. 通过设 STON 为高触发脉冲
4. 在单脉冲模式下, STIO[1:0] 必须设为“11”且不能被改变。

要工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位需要设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。STP 脚上的外部信号，通过设置 STMC1 寄存器的 STIO1 和 STIO0 位选择有效边沿类型，即上升沿，下降沿或双沿有效。通过应用程序将 STON 位由低到高转变时，计数器启动。

当 STP 脚出现有效边沿转换时，计数器当前值被锁存到 CCRA 寄存器，并产生 STM 中断。无论 STP 引脚发生何种事件，计数器将继续工作直到 STON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 STM 中断。记录 CCRP 溢出中断信号的值可以测量脉宽。通过设置 STIO1 和 STIO0 位选择 STP 引脚为上升沿，下降沿或双沿有效。如果 STIO1 和 STIO0 位都设置为高，无论 STP 引脚发生哪种边沿转换都不会产生捕捉操作，但应注意计数器将会继续运行。STCCLR 和 STDPX 位在此模式中未使用。

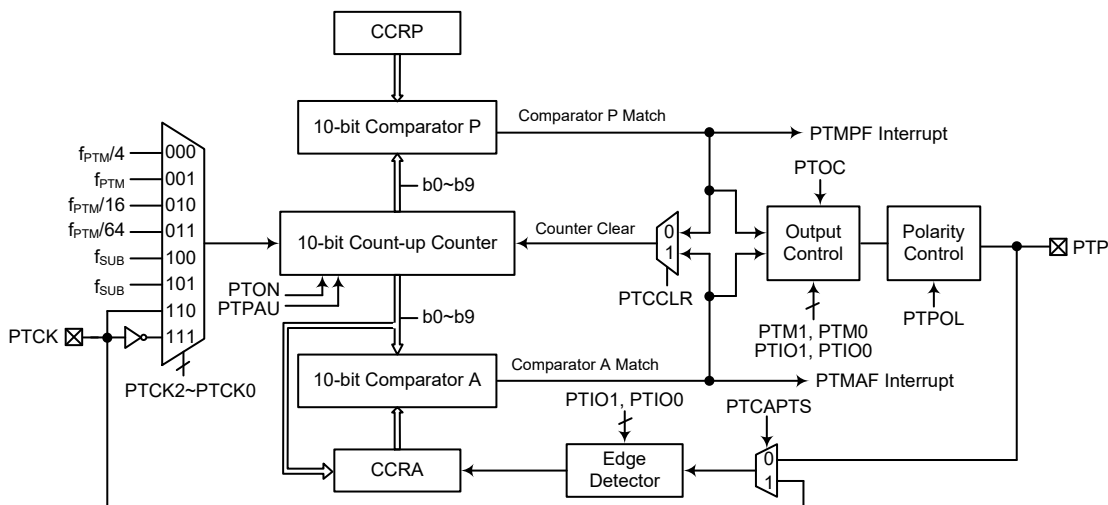


捕捉输入模式

- 注：1. STM[1:0]=01，有效边沿通过 STIO[1:0] 位段设置
2. TM 捕捉输入引脚有效边沿将计数器的值传到 CCRA 中
3. STCCLR 和 STDPX 位未使用
4. 无输出功能 – STOC 和 STPOL 位未使用
5. CCRP 决定计数器的值，且当 CCRP 等于 0 时计数器有一个最大计数值。

周期型 TM – PTM

周期型 TM 包括 5 种工作模式，即比较匹配输出、定时 / 事件计数器、捕捉输入、单脉冲输出和 PWM 输出模式。



注：1. PTM 外部引脚为多种功能共用引脚，因此在使用 PTM 功能之前应该合理配置相应引脚共用功能选择寄存器以选择所需的 PTM 引脚功能。

2. 专用振荡器输出频率 f_{PTM} 为 PTM 提供了时钟。更多振荡器微调和使能控制详见振荡器章节。

周期型 TM 方框图

周期型 TM 操作

周期型 TM 的核心是一个由用户选择的内部或外部时钟源驱动的 10-bit 向上计数器，它还包括两个内部比较器即比较器 A 和比较器 P。这两个比较器将计数器的值与 CCRA 和 CCRP 寄存器中的值进行比较。CCRP 的宽度为 10-bit。

通过应用程序改变 10-bit 计数器值的唯一方法是使 PTON 位发生上升沿跳变清除计数器。此外，计数器溢出或比较匹配也会自动清除计数器。上述条件发生时，通常情况会产生 PTM 中断信号。周期型 TM 可工作在不同的模式，可由包括来自输入脚的不同时钟源驱动，也可以控制输出脚。所有工作模式的设定都是通过设置相关内部寄存器来实现的。

周期型 TM 寄存器介绍

周期型 TM 的所有操作由一系列寄存器控制。一对只读寄存器用来存放 10-bit 内部计数器的值，两对读 / 写寄存器存放 10-bit CCRA 和 CCRP 的值。剩下两个控制寄存器用来设置不同的操作和控制模式。

寄存器名称	位							
	7	6	5	4	3	2	1	0
PTMC0	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
PTMC1	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	PTCAPTS	PTCCLR
PTMDL	D7	D6	D5	D4	D3	D2	D1	D0
PTMDH	—	—	—	—	—	—	D9	D8
PTMAL	D7	D6	D5	D4	D3	D2	D1	D0
PTMAH	—	—	—	—	—	—	D9	D8
PTMRPL	D7	D6	D5	D4	D3	D2	D1	D0
PTMRPH	—	—	—	—	—	—	D9	D8

10-bit 周期型 TM 寄存器列表

● PTMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **PTPAU**: PTM 计数器暂停控制位

0: 运行
1: 暂停

通过设置此位为高可使计数器暂停，清零此位恢复正常计数器操作。当处于暂停条件时，PTM 保持上电状态并继续耗电。当此位由低到高转变时，计数器将保留其剩余值，直到此位再次改变为低电平，并从此值开始继续计数。

Bit 6~4 **PTCK2~PTCK0**: 选择 PTM 计数器时钟位

000: $f_{PTM}/4$
001: f_{PTM}
010: $f_{PTM}/16$
011: $f_{PTM}/64$
100: f_{SUB}
101: f_{SUB}
110: PTCK 上升沿时钟
111: PTCK 下降沿时钟

此三位用于选择 PTM 的时钟源。外部引脚时钟源可被选择在上升沿或下降沿有效。时钟源 f_{PTM} 是 PTM 专用振荡器，PTMOSC，输出频率， f_{SUB} 是其它的内部时钟源，细节方面请参考振荡器章节。

Bit 3 **PTON**: PTM 计数器 On/Off 控制位

0: Off
1: On

此位控制 PTM 的总开关功能。设置此位为高则使能计数器使其运行，清零此位则除能 PTM。清零此位将停止计数器并关闭 PTM 减少耗电。当此位经由低到高转换时，内部计数器将复位清零，当此位由高到低转换时，内部计数器将保持其剩余值直到此位再次转高。

若 PTM 处于比较匹配输出模式、PWM 输出模式或单脉冲输出模式，当 PTON 位经由低到高转换时，PTM 输出脚将复位至 PTOC 位指定的初始值。

Bit 2~0 未定义，读为“0”

● PTMC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	PTCAPTS	PTCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **PTM1~PTM0**: PTM 工作模式选择位

- 00: 比较匹配输出模式
- 01: 捕捉输入模式
- 10: PWM 输出模式或单脉冲输出模式
- 11: 定时 / 计数器模式

这两位为 PTM 设置所需工作模式。为确保操作可靠，应在 PTM1 和 PTM0 位发生任何改变之前关闭 PTM。在定时 / 计数器模式中，PTM 输出引脚状态未定义。

Bit 5~4 **PTIO1~PTIO0**: PTM 外部引脚功能选择位

比较匹配输出模式

- 00: 无改变
- 01: 输出低
- 10: 输出高
- 11: 输出翻转

PWM 输出模式 / 单脉冲输出模式

- 00: PWM 输出无效状态
- 01: PWM 输出有效状态
- 10: PWM 输出
- 11: 单脉冲输出

捕捉输入模式

- 00: 在 PTP 或 PTCK 上升沿输入捕捉
- 01: 在 PTP 或 PTCK 下降沿输入捕捉
- 10: 在 PTP 或 PTCK 双沿输入捕捉
- 11: 输入捕捉除能

定时 / 计数器模式

未使用

这两位用于确定在达到一定条件时 PTM 输出脚如何改变状态。此位段所选择的功能取决于 PTM 当下的工作模式。

在比较匹配输出模式中，当比较器 A 发生比较匹配时，PTIO1~PTIO0 决定了 PTM 输出脚的状态应如何改变。当比较器 A 发生比较匹配时，PTM 输出脚可以设置为切换高，切换低或翻转当前状态。当这两位都为零，则输出将不会发生改变。PTM 输出脚的初始值应通过 PTMC1 寄存器中的 PTOC 位进行设置。注意，由 PTIO1 和 PTIO0 位所确定的输出电平必须与利用 PTOC 位所设置的初始值不同，否则在发生比较匹配时 PTM 输出脚上将不发生改变。在 PTM 输出脚改变状态后，可通过改变 PTON 位的电平从低变高将其复位至初始值。

在 PWM 输出模式中，当发生某种比较匹配情况时，PTIO1~PTIO0 决定了 PTM 输出脚的状态应如何改变。PWM 输出功能通过改变这两位进行更新。在 PTM 关闭后再来改变 PTIO1~PTIO0 的值是很有必要的。若在 PTM 仍运行时改变 PTIO1~PTIO0 的值，所得的 PWM 输出是不可预期的。

Bit 3 **PTOC**: PTM PTP 输出控制位

比较匹配输出模式

- 0: 初始低
- 1: 初始高

PWM 输出模式 / 单脉冲输出模式

- 0: 低有效
- 1: 高有效

此位为 PTM 输出脚的输出控制位。其操作取决于 PTM 被使用在比较匹配输出模式还是 PWM 输出模式 / 单脉冲输出模式。若 PTM 处于定时 / 计数器模式下则不受此位影响。在比较匹配输出模式中，此位决定了在比较匹配发生之前

PTM 输出脚的逻辑电平。在 PWM 输出模式中，此位决定了 PWM 信号是为高有效还是低有效。

Bit 2 **PTPOL:** PTP 输出极性控制位
0: 同相
1: 反相

此位控制 PTP 输出引脚的极性。当此位置高，PTM 输出脚将被反相，而此位为零时则为同相。若 PTM 处于定时 / 计数器模式则对其无影响。

Bit 1 **PTCAPTS:** PTM 输入捕捉触发源选择位
0: 来自 PTP
1: 来自 PTCK

Bit 0 **PTCCLR:** PTM 计数器清除条件选择位
0: PTM 比较器 P 匹配
1: PTM 比较器 A 匹配

此位用于选择清除计数器的方法。记住周期型 TM 包含两个比较器，即比较器 A 和比较器 P，两者中的任意一个都可以清除内部计数器。当 PTCCLR 位置高，在比较器 A 发生比较匹配时计数器将被清除。当此位为低，计数器将在比较器 P 发生比较匹配或是计数器溢出时被清除。只有在 CCRP 位都被清零时才可执行计数器溢出清除方式。PTCCLR 位在 PWM 输出，单脉冲或输入捕捉模式中未使用。

● PTMDL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0:** PTM 计数器低字节寄存器 bit 7~bit 0
PTM 10-bit 计数器 bit 7~bit 0

● PTMDH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8:** PTM 计数器高字节寄存器 bit 1~bit 0
PTM 10-bit 计数器 bit 9~bit 8

● PTMAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0:** PTM CCRA 低字节寄存器 bit 7~bit 0
PTM 10-bit CCRA bit 7~bit 0

● PTMAH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8**: PTM CCRA 高字节寄存器 bit 1~bit 0
PTM 10-bit CCRA bit 9~bit 8

● PTMRPL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**: PTM CCRP 低字节寄存器 bit 7~bit 0
PTM 10-bit CCRP bit 7~bit 0

● PTMRPH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 **D9~D8**: PTM CCRP 高字节寄存器 bit 1~bit 0
PTM 10-bit CCRP bit 9~bit 8

周期型 TM 工作模式

周期型 TM 有五种工作模式，即比较匹配输出模式、PWM 输出模式、单脉冲输出模式、捕捉输入模式或定时 / 计数器模式。通过设置 PTMC1 寄存器的 PTM1 和 PTM0 位选择任意模式。

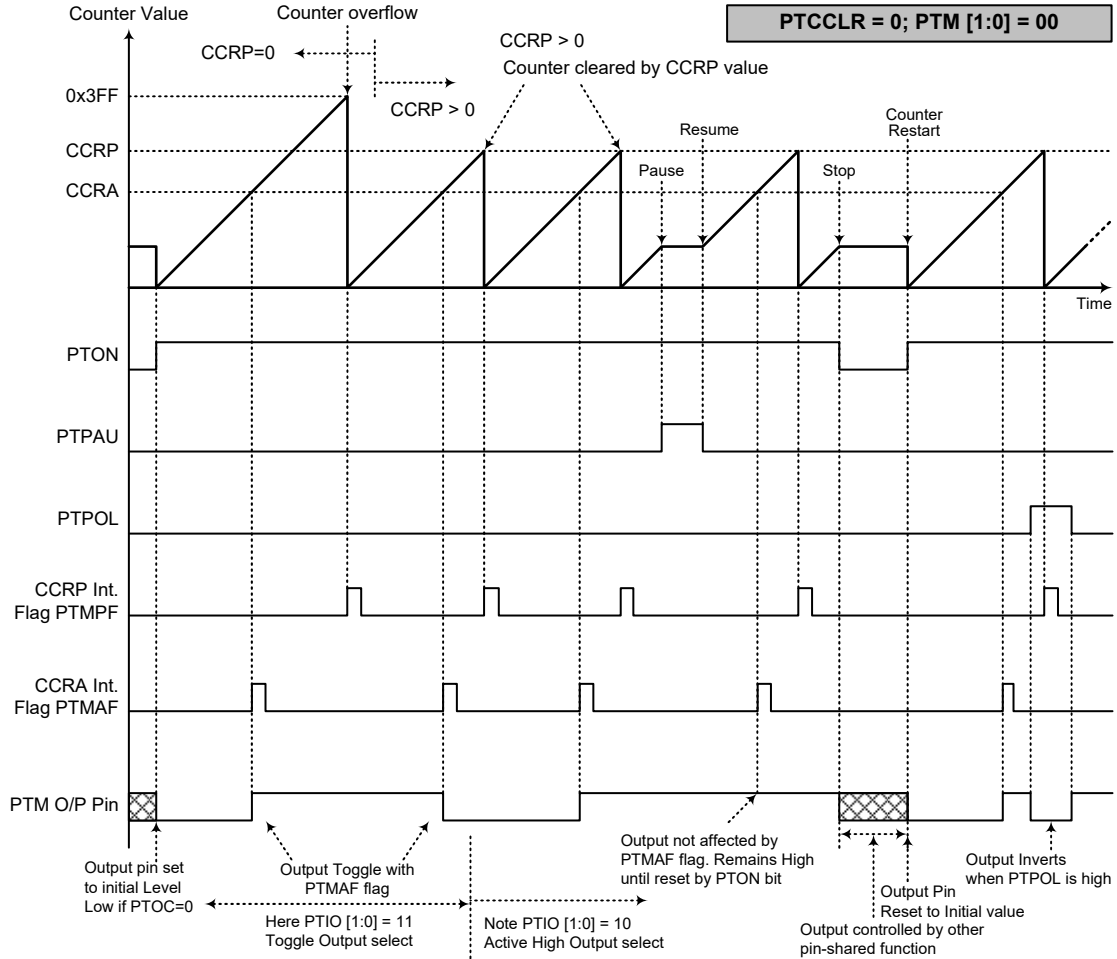
比较匹配输出模式

要工作在此模式，PTMC1 寄存器的 PTM1 和 PTM0 位的值应设为“00”。当工作在该模式，一旦计数器使能并开始计数，有三种方法来清零，分别是：计数器溢出，比较器 A 比较匹配发生和比较器 P 比较匹配发生。当 PTCCLR 位为低，有两种方法清除计数器。一种是比较器 P 比较匹配发生，另一种是 CCRP 所有位设置为零并使得计数器溢出。此时，比较器 A 和比较器 P 的请求标志位 PTMAF 和 PTMPF 将分别置起。

如果 PTMC1 寄存器的 PTCCLR 位设置为高，当比较器 A 比较匹配发生时计数器被清零。此时，即使 CCRP 寄存器的值小于 CCRA 寄存器的值，仅 PTMAF 中断请求标志产生。所以当 PTCCLR 为高时，不会产生 PTMPF 中断请求标志。在比较匹配输出模式中，CCRA 不能被设置为“0”。

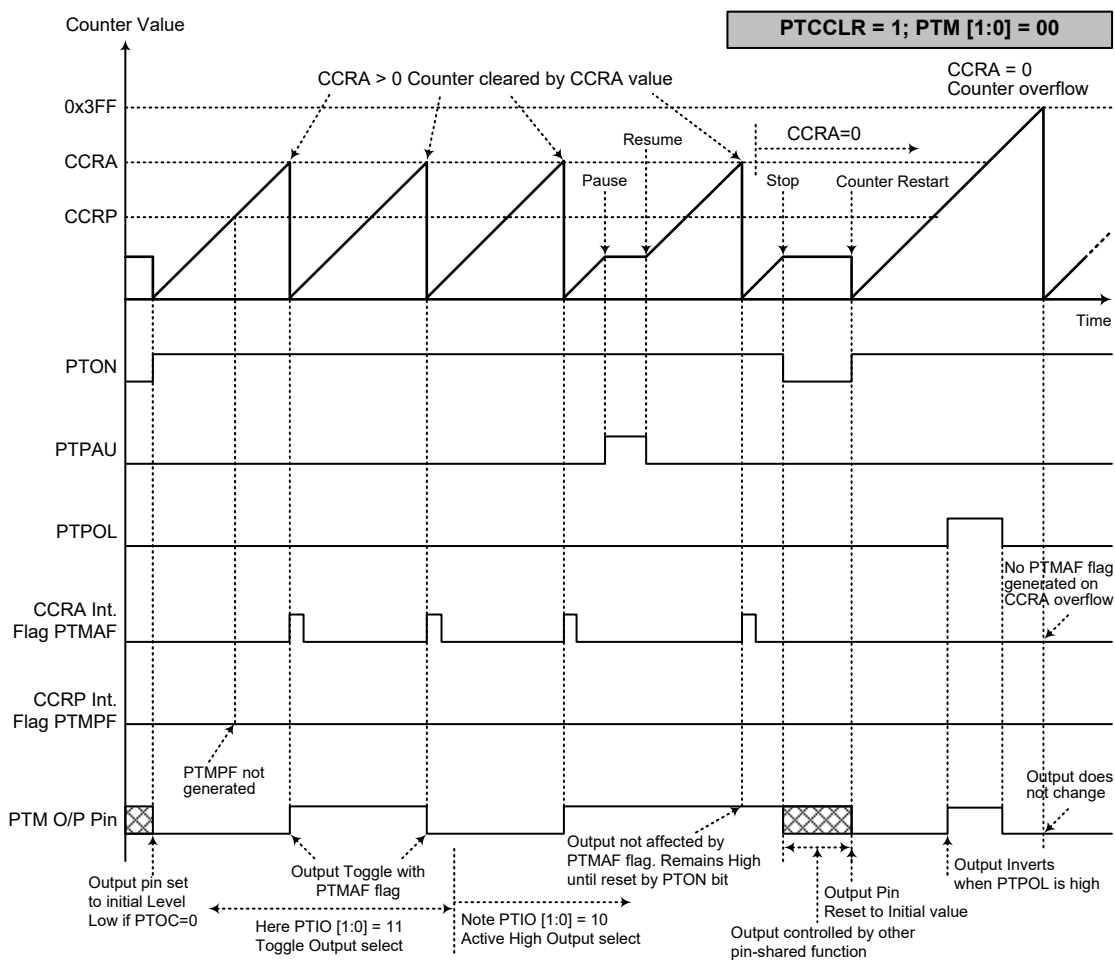
如果 CCRA 被清零，当计数达到最大值 3FFH 时，计数器溢出，而此时不产生 PTMAF 请求标志。

正如该模式名所言，当比较匹配发生后，PTM 输出脚状态改变。当比较器 A 比较匹配发生后 PTMAF 中断请求标志产生时，PTM 输出脚状态改变。比较器 P 比较匹配发生时产生的 PTMPF 标志不影响 PTM 输出脚。PTM 输出脚状态改变方式由 PTMC1 寄存器中 PTIO1 和 PTIO0 位决定。当比较器 A 比较匹配发生时，利用 PTIO1 和 PTIO0 位决定 PTM 输出脚输出高，低或翻转当前状态。PTM 输出脚初始值，在 PTON 位由低到高电平的变化后由 PTOC 位设置。注意，若 PTIO1 和 PTIO0 位同时为 0 时，引脚输出不变。



比较匹配输出模式 – PTCCLR=0

- 注：1. PTCCLR=0，比较器 P 匹配将清除计数器
2. PTM 输出引脚仅由 PTMAF 标志位控制
3. 输出引脚通过 PTON 上升沿复位至初始值



比较匹配输出模式 – PTCCLR=1

- 注：1. PTCCLR=1，比较器 A 匹配将清除计数器
2. PTM 输出引脚仅由 PTMAF 标志位控制
3. 输出引脚通过 PTON 上升沿复位至初始值
4. 当 PTCCLR=1 时不产生 PTMPF 标志位

定时 / 计数器模式

要工作在此模式，PTMC1 寄存器的 PTM1 和 PTM0 位的值需设为“11”。定时 / 计数器模式与比较输出模式操作方式相同，并产生同样的中断请求标志。不同的是，在定时 / 计数器模式下 TM 输出脚未使用。因此，比较匹配输出模式中的描述和时序图可以适用于此功能。该模式中未使用的 TM 输出脚用作普通 I/O 脚或其它功能。

PWM 输出模式

要工作在此模式，PTMC1 寄存器的 PTM1 和 PTM0 位的值需设置为“10”。PTM 的 PWM 功能在马达控制，加热控制，照明控制等方面十分有用。给 PTM 输出脚提供一个频率固定但占空比可调的信号，将产生一个有效值等于 DC 均方根的 AC 方波。

由于 PWM 波形的周期和占空比可调，其波形的选择就较为灵活。在 PWM 输出模式中，PTCCLR 位对 PWM 周期无效。CCRA 和 CCRP 寄存器共同产生 PWM 波形，一个用来清除内部计数器从而控制 PWM 波形频率，另一个则用于控制占空比。因此，PWM 波形频率和占空比可由 CCRA 和 CCRP 寄存器中的值来控制。

当比较器 A 或比较器 P 发生比较匹配时，将产生 CCRA 和 CCRP 的中断标志位。PTMC1 寄存器的 PTOC 位用于选择 PWM 波形所需极性，而 PTIO1 和 PTIO0 位用于使能 PWM 输出或将 PTM 输出脚强制固定为高或低电平。PTPOL 位用于取反 PWM 输出波形的极性。

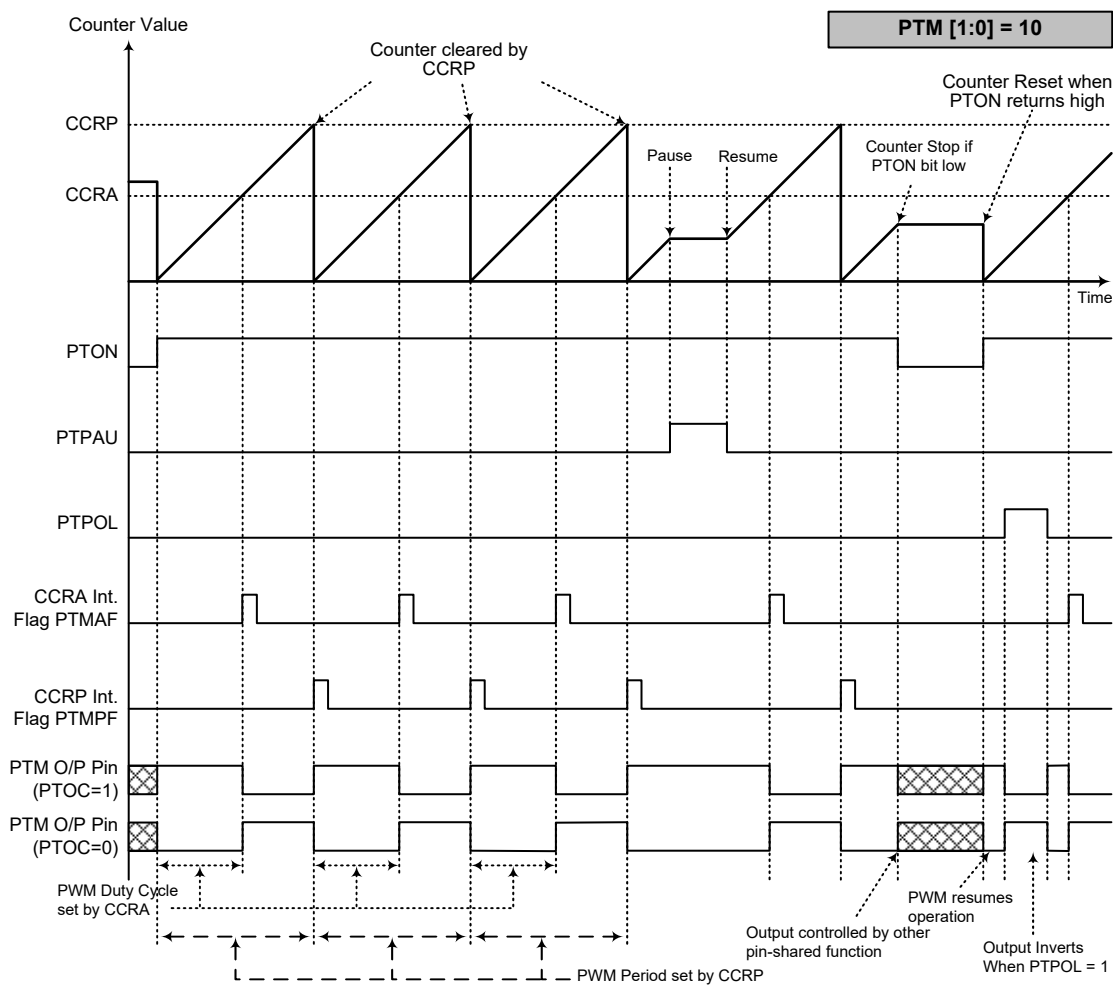
● 10-bit PTM，PWM 输出模式，边沿对齐模式

CCRP	1~1023	0
Period	1~1023	1024
Duty	CCRA	

若 $f_{SYS}=8\text{MHz}$ ，PTM 时钟源选择 $f_{SYS}/4$ ，CCRP=512，CCRA=128

PTM PWM 输出频率 = $(f_{SYS}/4)/512=f_{SYS}/2048=4\text{kHz}$ ，duty=128/512=25%

若由 CCRA 寄存器定义的 Duty 值等于或大于 Period 值，PWM 输出占空比为 100%。



PWM 输出模式

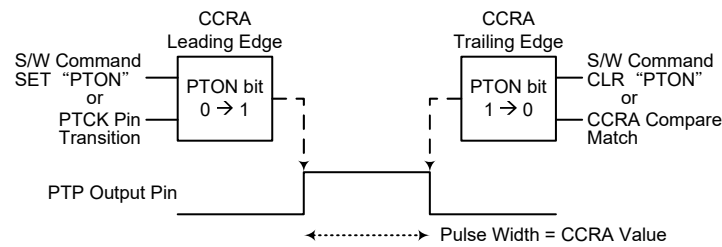
- 注：1. 计数器由 CCRP 清除
2. 计数器清除设置 PWM 周期
3. 当 PTIO[1:0]=00 或 01，内部 PWM 功能继续运行
4. PTCCLR 位对 PWM 操作无影响

单脉冲模式

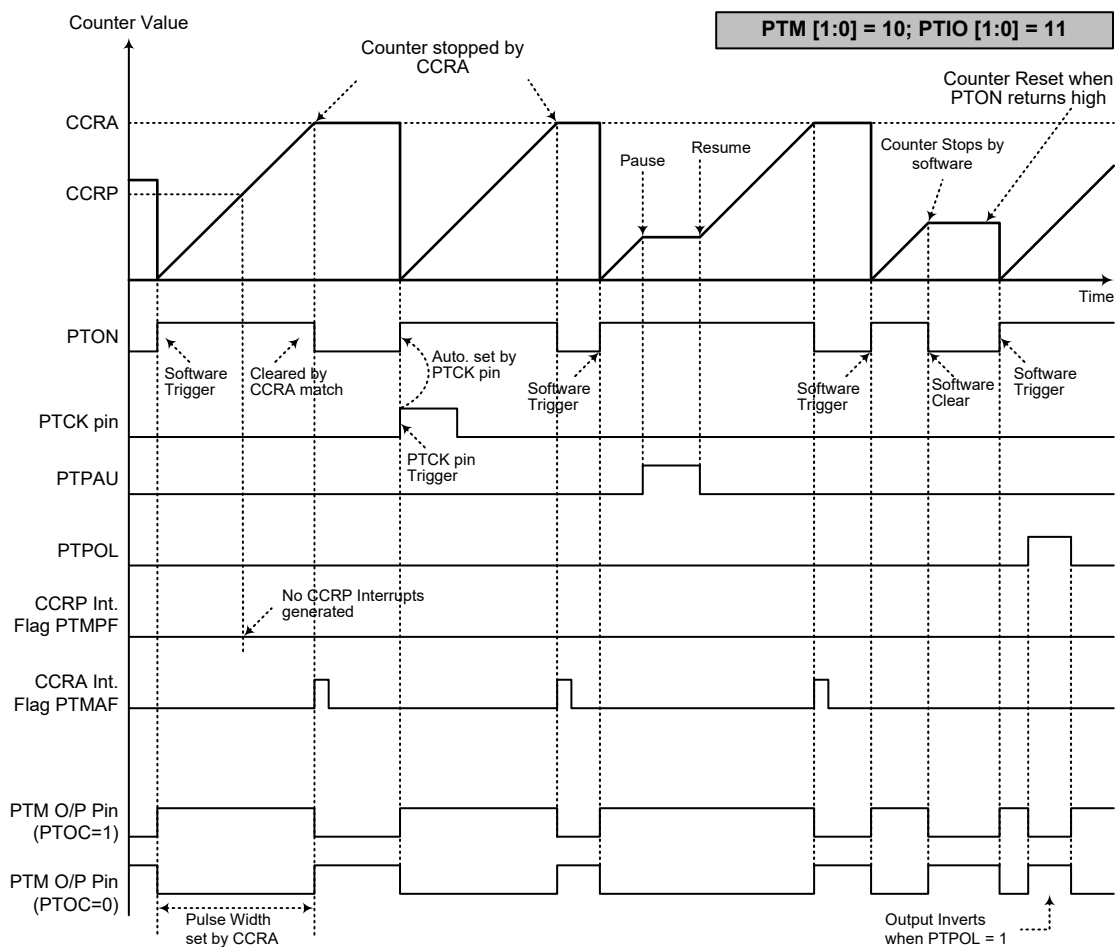
要在此模式，PTMC1 寄存器的 PTM1 和 PTM0 位的值需设置为“10”，且 PTIO1 和 PTIO0 位的值也需设置为“11”。正如模式名所言，单脉冲输出模式，在 PTM 输出脚将产生一个脉冲输出。

通过应用程序控制 PTON 位由低到高的转变来触发脉冲前沿输出。但在单脉冲模式中，PTON 位可利用外部 PTCK 脚而自动由低转变为高，进而启动单脉冲输出。当 PTON 位转变为高电平时，计数器将开始运行，并产生脉冲前沿。当脉冲有效时 PTON 位保持高电平。通过应用程序使 PTON 位清零或比较器 A 比较匹配发生时，产生脉冲后沿。

而比较器 A 比较匹配发生时，会自动清除 PTON 位并产生单脉冲输出后沿。CCRA 的值通过这种方式控制脉冲宽度。比较器 A 比较匹配发生时，会产生 PTM 中断。PTON 位在计数器重启时会发生由低到高的转变，此时计数器才复位至零。在单脉冲模式中，CCRP 寄存器和 PTCCLR 位未使用。



单脉冲生成示意图



单脉冲模式

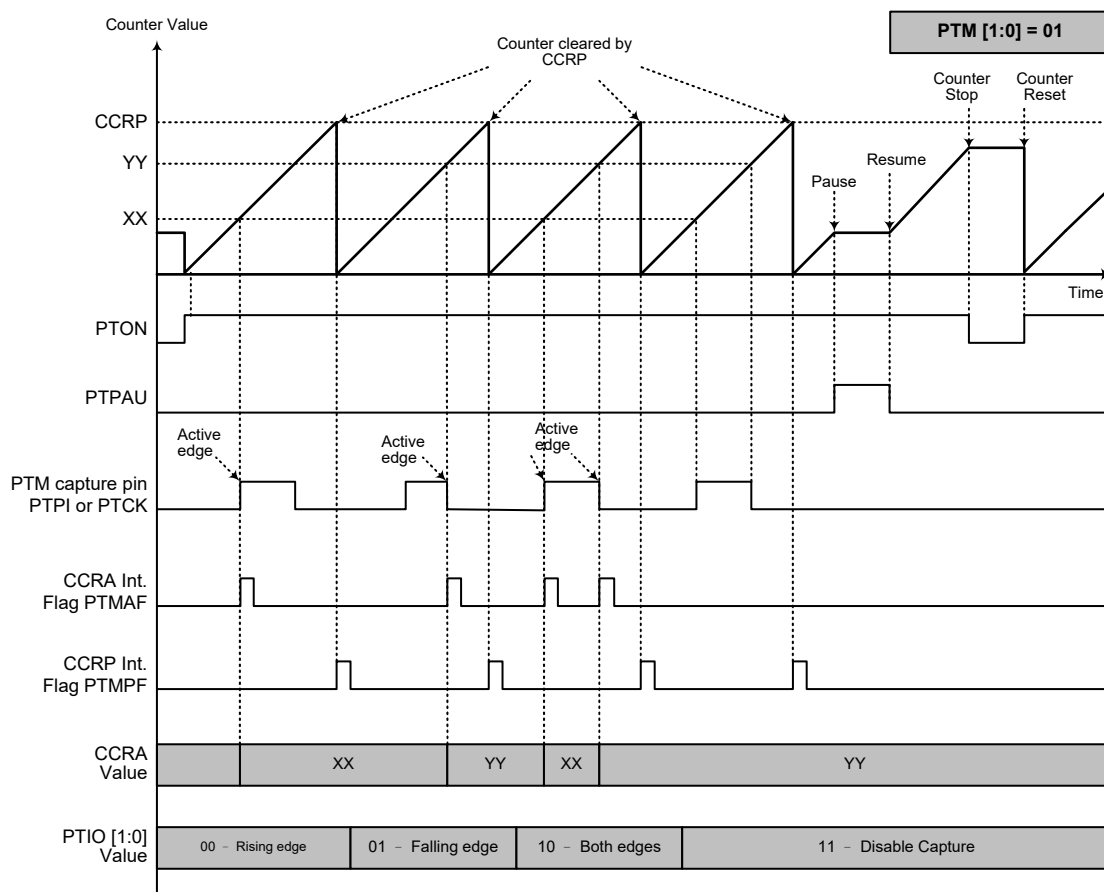
- 注：1. 通过 CCRA 匹配停止计数器
2. CCRP 未使用
3. 通过 PTCK 引脚或设置 PTON 位为高触发脉冲
4. PTCK 脚有效边沿自动将 PTON 位置高
5. 单脉冲模式下，PTIO[1:0] 必须设为“11”且不能改变

捕捉输入模式

要工作在此模式，PTMC1 寄存器的 PTM1 和 PTM0 位的值需设置为“01”。此模式使能外部信号捕捉并保存内部计数器当前值，因此被用于诸如脉冲宽度测量的应用中。PTP 和 PTCK 引脚上的外部信号可通过设置 PTMC1 寄存器的 PTCAPTS 位进行选择。输入引脚有效边沿可为上升沿，下降沿或双沿有效。有效边沿转换类型可通过设置 PTMC1 寄存器的 PTIO1 和 PTIO0 位进行选择。通过应用程序将 PTON 位由低到高转变时，计数器启动。

当 PTP 和 PTCK 引脚出现所需边沿转换时，计数器当前值被锁存到 CCRA 寄存器，并产生 PTM 中断。无论 PTP 和 PTCK 引脚发生什么事件，计数器继续工作直到 PTON 位发生下降沿跳变。当 CCRP 比较匹配发生时计数器复位至零；CCRP 的值通过这种方式控制计数器的最大值。当比较器 P CCRP 比较匹配发生时，也会产生 PTM 中断。计数 CCRP 溢出中断信号的值可以测量长脉宽。PTIO1 和 PTIO0 位可选择 PTP 和 PTCK 引脚上的上升沿，下降沿或双沿为有效触发边沿类型。如果 PTIO1 和 PTIO0 位都设为高，则无论 PTP 和 PTCK 引脚发生什么事件都不会产生捕捉操作，但必须注意，计数器将继续运行。

由于 PTP 和 PTCK 引脚与其它功能共用，则 PTM 工作在输入捕捉模式时需多加注意。这是因为如果引脚被设为输出，那么该引脚上的任何电平转变都可能执行输入捕捉操作。PTCCLR、PTOC 和 PTPOL 位在此模式中未使用。



捕捉输入模式

- 注：1. PTM[1:0]=01，有效边沿通过 PTIO[1:0] 位段设置
2. PTM 捕捉输入引脚有效边沿将计数器的值传到 CCRA 中
3. PTCCLR 位未使用
4. 无输出功能 – PTOC 和 PTPOL 位未使用
5. CCRP 决定计数器的值，且当 CCRP 等于 0 时计数器有一个最大计数值。

A/D 转换器

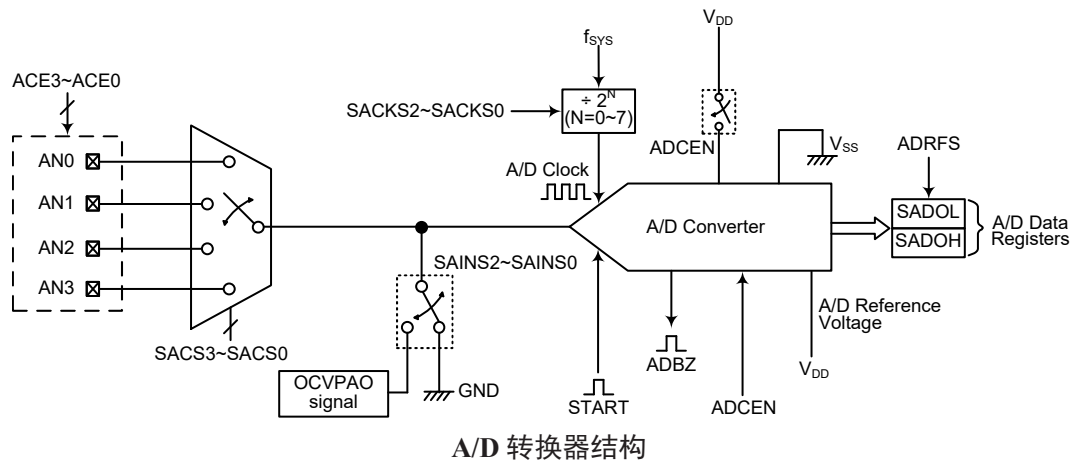
对于大多数电子系统而言，处理现实世界的模拟信号是共同的需求。为了完全由单片机来处理这些信号，首先需要通过 A/D 转换器将模拟信号转换成数字信号。将 A/D 转换器电路集成入单片机，可有效的减少外部器件，随之而来，具有降低成本和减少器件空间需求的优势。

A/D 转换器简介

该单片机包含一个多通道的 A/D 转换器，它们可以直接接入外部模拟信号（来自传感器或其它控制信号）并直接将它们转换成 12-bit 数字量。也可对内部模拟信号，如来自 OCVPAO 的 OCVPAO 信号等进行 A/D 转换。选择转换外部或内部模拟信号由 SAINS2~SAINS0 和 SACS3~SACS0 位控制。注意若选择内部模拟信号，除了设置 SAINS 位段和 SACS 位段外还需合理设置 ACE_n 位。关于 A/D 输入信号的更多详细信息请分别参见“A/D 转换器控制寄存器”和“A/D 转换器输入信号”章节。

外部输入通道	内部输入信号	A/D 输入选择位
AN0~AN3	OCVPAO	SACS3~SACS0 SAINS2~SAINS0 ACE3~ACE0

下图显示了 A/D 转换器内部结构和相关的寄存器。



A/D 转换器寄存器介绍

A/D 转换器的所有工作由四个寄存器控制。一对只读寄存器来存放 12-bit A/D 转换数据的值，剩下的两个控制寄存器设置 A/D 转换器的操作和控制功能。

寄存器名称	位							
	7	6	5	4	3	2	1	0
SADOL(ADRF5=0)	D3	D2	D1	D0	—	—	—	—
SADOL(ADRF5=1)	D7	D6	D5	D4	D3	D2	D1	D0
SADOH(ADRF5=0)	D11	D10	D9	D8	D7	D6	D5	D4
SADOH(ADRF5=1)	—	—	—	—	D11	D10	D9	D8
SADC0	START	ADBZ	ADCEN	ADRF5	SACS3	SACS2	SACS1	SACS0
SADC1	SAINS2	SAINS1	SAINS0	—	—	SACKS2	SACKS1	SACKS0
ACERL	—	—	—	—	ACE3	ACE2	ACE1	ACE0

A/D 转换器寄存器列表

A/D 转换器数据寄存器

对于具有 12-bit A/D 转换器的单片机，需要两个数据寄存器存放转换结果，一个高字节寄存器 SADOH 和一个低字节寄存器 SADOL。在 A/D 转换完毕后，单片机可以直接读取这些寄存器以获得转换结果。由于寄存器只使用了 16 位中的 12 位，其数据存储格式由 SADC0 寄存器的 ADRF5 位控制，如下表所示。D0~D11 是 A/D 转换数据结果位。未使用的位读为“0”。需注意的是，当 A/D 转换器除能时，数据寄存器的值将保持不变。

ADRF5	SADOH								SADOL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

A/D 转换器数据寄存器

A/D 转换器控制寄存器

寄存器 SADC0、SADC1 和 ACERL 用来控制 A/D 转换器的功能和操作。这些 8-bit 寄存器定义包括选择连接至内部 A/D 转换器的模拟通道，数字化数据格式，A/D 时钟源，并控制和监测 A/D 转换器的开始和转换结束状态。寄存器 SADC0 的 SACS3~SACS0 位定义 A/D 转换器外部输入通道编号。寄存器 SADC1 的 SAINS2~SAINS0 位用来决定模拟信号是来自内部模拟信号或外部模拟输入信号。

ACERL 控制寄存器所包含的 ACE3~ACE0 位段用来确定 PB 口中的哪些引脚为 A/D 转换器的模拟输入，哪些引脚不作为 A/D 转换输入。相应的位置高将选中 A/D 转换器输入功能，清零此位则选中 I/O 功能或其它引脚共用功能。当引脚被选择为 A/D 转换器输入，其原来的 I/O 或其它引脚共用功能除能。此外，这些引脚上所连接的内部上拉电阻将自动移除。

● SADC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	START	ADBZ	ADCEN	ADRF5	SACS3	SACS2	SACS1	SACS0
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 **START**: 启动 A/D 转换位
0→1→0: 启动 A/D 转换
此位用于启动 A/D 转换过程。通常此位为低，但如果设为高再被清零，将启动 A/D 转换过程。
- Bit 6 **ADBZ**: A/D 转换忙标志位
0: A/D 转换结束或未开始转换
1: A/D 转换中
该只读标志位用于表明 A/D 转换过程是否完成。当 START 位由低变为高再变为低时，ADBZ 位为高，表明 A/D 转换已启动。A/D 转换结束后，此位被清零。
- Bit 5 **ADCEN**: A/D 转换器功能使能控制位
0: 除能
1: 使能
此位控制 A/D 内部功能。该位被置高将使能 A/D 转换器。如果该位设为低将关闭 A/D 转换器以降低功耗。当 A/D 转换器功能除能时，A/D 数据寄存器 SADOH 和 SADOL 的内容将不变。
- Bit 4 **ADRF5**: A/D 转换器数据格式控制位
0: A/D 转换器数据格式 → SADOH=D[11:4]; SADOL=D[3:0]
1: A/D 转换器数据格式 → SADOH=D[11:8]; SADOL=D[7:0]
此位控制存放在两个 A/D 数据寄存器中的 12-bit A/D 转换结果的格式。细节方面请参考 A/D 转换器数据寄存器章节。
- Bit 3~0 **SACS3~SACS0**: A/D 转换器外部模拟输入通道选择位
0000: AN0
0001: AN1
0010: AN2
0011: AN3
0100~1111: 不存在的通道，输入将浮空。

● SADC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SAINS2	SAINS1	SAINS0	—	—	SACKS2	SACKS1	SACKS0
R/W	R/W	R/W	R/W	—	—	R/W	R/W	R/W
POR	0	0	0	—	—	0	0	0

- Bit 7~5 **SAINS2~SAINS0**: A/D 转换器输入信号选择位
000: 外部来源 – 外部模拟通道输入
001: 内部来源 – OCVPAO (OCVP 电路输出)
010~100: 接地
101~111: 外部来源 – 外部模拟通道输入
若 SAINS2~SAINS0 位段设置为“001”~“100”模拟信号进行转换时必须特别小心。当选择转换内部模拟信号，SACS3~SACS0 位段必须设置为“0100”~“1111”范围内的值。否则外部通道输入将与内部模拟信号连接，将导致不可预期的后果。
- Bit 4~3 未定义，读为“0”
- Bit 2~0 **SACKS2~SACKS0**: A/D 转换器时钟源选择位
000: f_{SYS}
001: $f_{SYS}/2$

010: $f_{SYS}/4$
011: $f_{SYS}/8$
100: $f_{SYS}/16$
101: $f_{SYS}/32$
110: $f_{SYS}/64$
111: $f_{SYS}/128$

这几位用于选择 A/D 转换器的时钟源。

● ACERL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	ACE3	ACE2	ACE1	ACE0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义，读为“0”
 Bit 3 **ACE3**: 定义 PB3 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN3
 Bit 2 **ACE2**: 定义 PB2 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN2
 Bit 1 **ACE1**: 定义 PB1 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN1
 Bit 0 **ACE0**: 定义 PB0 是否为 A/D 输入
 0: 不是 A/D 输入
 1: A/D 输入, AN0

A/D 转换器操作

SADC0 寄存器的 START 位用于启动 A/D 转换。当单片机设定此位从逻辑低到逻辑高，然后再到逻辑低，就会开始一个模数转换周期。

SADC0 寄存器中的 ADBZ 位用于表明模数转换过程是否完成。A/D 转换成功启动后，ADBZ 位被自动置为“1”。在转换周期结束后，ADBZ 位会被清零。此外，中断控制寄存器内相应的 A/D 中断请求标志位也会置位，如果中断使能，就会产生对应的内部中断信号。A/D 内部中断信号将引导程序到相应的 A/D 内部中断入口。如果 A/D 内部中断被禁止，可以让单片机轮询 SADC0 寄存器中的 ADBZ 位，检测此位是否被清除，以作为另一种侦测 A/D 转换周期结束的方法。

A/D 转换器的时钟源为系统时钟 f_{SYS} 或其分频，而分频系数由 SADC1 寄存器中的 SACKS2~SACKS0 位决定。虽然 A/D 时钟源是由系统时钟 f_{SYS} 和 SACKS2~SACKS0 位决定，但可选择的最大 A/D 时钟源则有一些限制。由于允许的 A/D 时钟周期 t_{ADCK} 的范围为 $0.5\mu s \sim 10\mu s$ ，所以选择系统时钟频率时就必须小心。如果系统时钟速度为 4MHz 时，SACKS2~SACKS0 位不能设为“000”、“110”或“111”。必须保证设置的 A/D 转换时钟周期不小于时钟周期的最小值或不大于时钟周期最大值，否则将会产生不准确的 A/D 转换值。使用者可以参考下面的表格，被标上星号 * 的数值是不允许的。因为它们的 A/D 转换时钟周期不在规定的范围内。

f _{sys}	A/D 时钟周期 (t _{ADCK})							
	SACKS [2:0]=000 (f _{sys})	SACKS [2:0]=001 (f _{sys} /2)	SACKS [2:0]=010 (f _{sys} /4)	SACKS [2:0]=011 (f _{sys} /8)	SACKS [2:0]=100 (f _{sys} /16)	SACKS [2:0]=101 (f _{sys} /32)	SACKS [2:0]=110 (f _{sys} /64)	SACKS [2:0]=111 (f _{sys} /128)
1MHz	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *	128μs *
2MHz	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *
4MHz	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *
8MHz	125ns *	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *
12MHz	83ns *	167ns *	333ns *	667ns	1.33μs	2.67μs	5.33μs	10.67μs *

A/D 时钟周期范例

SADC0 寄存器的 ADCEN 位用于控制 A/D 转换电路电源的开启 / 关闭。该位必须置高以开启 A/D 转换器电源。当设置 ADCEN 位为高开启 A/D 转换器内部电路时，在 A/D 转换成功初始化前需一段延时。如时序图所示。即使通过设置相关引脚共用控制位，选择无引脚作为 A/D 输入，如果 ADCEN 设为“1”，那么仍然会产生功耗。因此当未使用 A/D 转换器功能时，在功耗敏感的应用中建议设置 ADCEN 为低以减少功耗。

A/D 转换器的参考电压可以来自正电源引脚 VDD。模拟输入值一定不能超过参考电压值。

A/D 转换器输入信号

A/D 转换器的所有模拟输入引脚与 I/O 引脚及其它功能共用。ACERL 寄存器中的 ACE3~ACE0 位决定了外部引脚是设置为 A/D 转换器模拟通道输入还是其它功能。如果引脚的对应位 ACE3~ACE0 设为高，相应引脚设置为 A/D 转换器模拟通道输入那么该引脚原引脚功能除能。通过这种方式，引脚的功能可由程序来控制，灵活地切换引脚功能。如果将引脚设为 A/D 输入，则通过寄存器编程设置的所有上拉电阻会自动断开。请注意，端口控制寄存器不需要为使能 A/D 输入而先设定为输入模式，当引脚共用选择位使能 A/D 输入时，端口控制寄存器的状态将被重置。

可通过配置 SAINS2~SAINS0 位段，选择 OCVF 电路输出的一个内部模拟信号连接到 A/D 转换器。若 SAINS2~SAINS0 位段设为“000”或“101~111”，则所要转换信号为外部模拟通道信号，SACS3~SACS0 位段可决定选择哪个外部通道的信号进行转换。若 SAINS2~SAINS0 位段设置为“001”，则选择来自 OCVF 功能的 OCVPAO 信号进行转换。若 SAINS2~SAINS0 设置为“010”~“011”，则 A/D 转换器输入接到地。当选择内部模拟信号转换时必须小心。若选择内部模拟信号，SACS3~SACS0 位段必须配置选择浮空状态，否则外部通道输入将与内部模拟信号连接，从而导致不可预期的后果。

SAINS[2:0]	SACS[3:0]	输入信号	描述
000, 101~111	0000~0011	AN0~AN3	外部引脚模拟输入
	0100~1111	—	浮空
001	0110~1111	OCVPAO	来自 OCVF 功能的 OCVPAO 信号
010~100	0110~1111	GND	接地

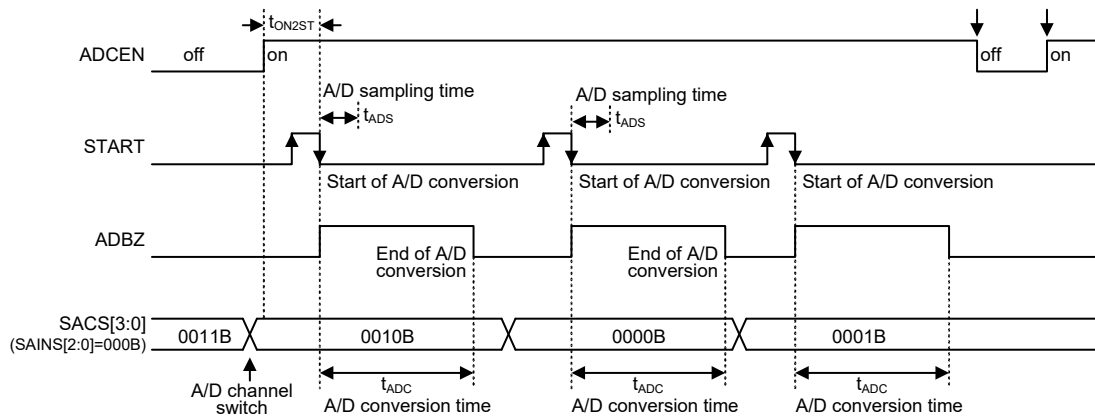
A/D 转换器输入信号选择

A/D 转换率和时序图

一个完整的 A/D 转换包含两部分，数据采样和数据转换。数据采样时间定义为 t_{ADS} ，需 4 个 A/D 时钟周期，而数据转换需 12 个 A/D 时钟周期。所以一个完整的 A/D 转换时间 t_{ADC} 一共需要 16 个 A/D 时钟周期。

最大 A/D 转换率 = A/D 时钟周期 / 16

下列时序图表示模数转换过程中不同阶段的图形与时序。由应用程序控制开始 A/D 转换过程后，单片机的内部硬件就会开始进行转换，在这个过程中，程序可以继续其它功能。A/D 转换时间为 $16t_{ADCK}$ ， t_{ADCK} 为 A/D 时钟周期。



A/D 转换时序图 - 外部模拟通道输入

A/D 转换步骤

下面概述实现 A/D 转换过程的各个步骤。

- 步骤 1
通过 SADC1 寄存器中的 SACKS2~SACKS0 位，选择所需的 A/D 转换器转换时钟。
- 步骤 2
将 SADC0 寄存器中的 ADCEN 位置高以使能 A/D 转换器。
- 步骤 3
通过 SADC1 寄存器中的 SAINS2~SAINS0 位段，选择连接至内部 A/D 转换器的信号。
选择外部通道输入，前往步骤 4。
选择内部模拟信号，前往步骤 5。
- 步骤 4
若配置 SAINS 位段将外部通道输入信号选作 A/D 输入信号，相应引脚应先通过配置 ACERL 寄存器的 ACE3~ACE0 位选择。配置 SACS 位段选择所需模拟通道。完成此步骤后，前往步骤 6。
- 步骤 5
在配置 SAINS 位段选择内部模拟信号作为 A/D 输入信号之前，SACS 位段需预先设置为“0100”到“1111”范围内的值以断开外部通道输入。接着通过 SAINS 位段选择所需内部模拟信号。完成此步骤后，前往步骤 6。
- 步骤 6
设置 SADC0 寄存器的 ADRFS 位选择 A/D 转换器输出数据格式。

- 步骤 7

如果要使用 A/D 转换中断，则中断控制寄存器需要正确地设置，以确保 A/D 中断功能是有效的。总中断控制位 EMI 以及 A/D 转换器中断位 ADE 需要预先置位为“1”。

- 步骤 8

现在可以通过设定 START 位从低到高再到低，开始 A/D 转换的过程。

- 步骤 9

若 A/D 转换正在进行，ADBZ 标志将置高。A/D 转换结束后，ADBZ 标志置低，并可从 SADOH 和 SADOL 寄存器读取输出数据。

注：若使用轮询 SADC0 寄存器中的 ADBZ 位的方法以检测模数转换过程是否完成，上述中断使能步骤可以忽略。

编程注意事项

在单片机工作时，如果 A/D 转换器未使用，通过设置 SADC0 寄存器中的 ADCEN 为低，关闭 A/D 内部电路以减少电源功耗。此时，不考虑输入脚的模拟电压，内部 A/D 转换器电路不产生功耗。如果 A/D 转换器输入脚用作普通 I/O 脚，必须特别注意，输入电压为无效逻辑电平也可能增加功耗。

A/D 转换功能

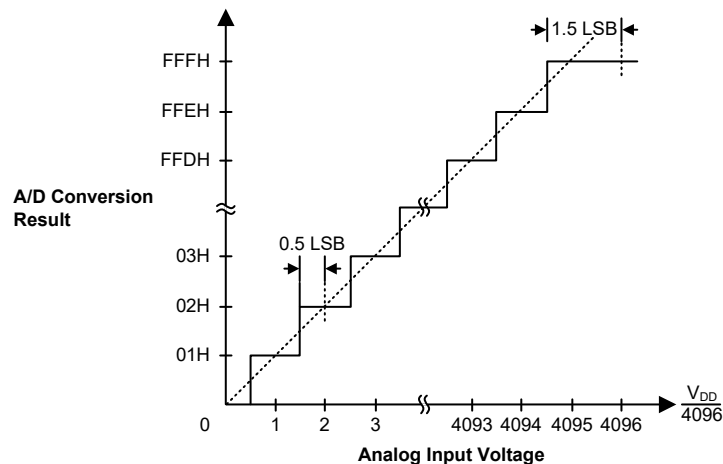
该单片机含有一个 12-bit A/D 转换器，它们转换的最大值可达 FFFH。由于模拟输入最大值等于实际 A/D 转换器参考电压， V_{DD} ，因此每一位可表示 $V_{DD}/4096$ 的模拟输入值。

$$1 \text{ LSB} = V_{DD} \div 4096$$

通过下面的等式可估算 A/D 转换器输入电压值：

$$\text{A/D 输入电压} = \text{A/D 数字输出值} \times V_{DD} \div 4096$$

下图显示 A/D 转换器模拟输入值和数字输出值之间理想的转换功能。除了数字化数值 0，其后的数字化数值会在精确点之前的 0.5 LSB 处改变，而数字化数值的最大值将在 V_{DD} 之前的 1.5 LSB 处改变。



理想的 A/D 转换功能

A/D 程序范例

下面两个范例程序用来说明怎样设置和实现 A/D 转换。第一个范例是轮询 SADC0 寄存器中的 ADBZ 位来判断 A/D 转换何时完成，而第二个范例则使用中断的方式判断。

范例 1：使用轮询 ADBZ 的方式来检测转换结束

```
clr ADE                                ; disable ADC interrupt
mov a,03H
mov SADC1,a                            ; select fsys/8 as A/D conversion clock
mov a,01H                              ; setup ACERL to configure pin AN0
mov ACERL,a
mov a,20H
mov SADC0,a                            ; enable and connect AN0 channel to A/D converter
:
start_conversion:
clr START                              ; high pulse on start bit to initiate conversion
set START                              ; reset A/D
clr START                              ; start A/D
:
polling_EOC:
sz ADBZ                                ; poll the SADC0 register ADBZ bit to detect end
                                        ; of A/D conversion
jmp polling_EOC                        ; continue polling
:
mov a,SADOL                            ; read low byte conversion result value
mov ADRL_buffer,a                      ; save result to user defined register
mov a,SAD0H                            ; read high byte conversion result value
mov ADRH_buffer,a                      ; save result to user defined register
:
jmp start_conversion                    ; start next A/D conversion
```

范例 2：使用中断的方式来检测转换结束

```
clr ADE                                ; disable ADC interrupt
mov a,03H
mov SADC1,a                            ; select fsys/8 as A/D conversion clock
mov a,01H                              ; setup ACERL to configure pin AN0
mov ACERL,a
mov a,20H
mov SADC0,a                            ; enable and connect AN0 channel to A/D converter
:
Start_conversion:
clr START                              ; high pulse on START bit to initiate conversion
set START                              ; reset A/D
clr START                              ; start A/D
clr ADF                                ; clear ADC interrupt request flag
set ADE                                ; enable ADC interrupt
set EMI                                ; enable global interrupt
:
:
ADC_ISR:                               ; ADC interrupt service routine
mov acc_stack,a                        ; save ACC to user defined memory
mov a,STATUS
mov status_stack,a                    ; save STATUS to user defined memory
:
mov a,SADOL                            ; read low byte conversion result value
mov adrl_buffer,a                     ; save result to user defined register
```


较。OCVPCI 引脚上的电压亦是与 8-bit D/A 转换器提供的参考电压进行比较。

通过 OCVPC1 寄存器中的 OCVPCPS 位可选择 OCVPAO 输出信号或 OCVPCI 输入信号与 D/A 转换器输出电压进行比较。8-bit D/A 转换器的电源可以由外部电源引脚 VDD 或 VREF 提供，这由 OCVPC1 寄存器中的 OCVPVRS 位决定。比较器输出 OCVPCOUT 将先经过一定去抖动时间周期的滤波，去抖动时钟周期由 OCVPC2 寄存器中的 OCVPDEB2~OCVPDEB0 位段选择。所获得的滤波后的 OCVP 数字比较器输出 OCVPO 表明了是否出现用户定义的电流或电压条件。

若 OCVPSPOL 位清零且比较器输入将 OCVPO 位电平由低强制拉高，或是 OCVPSPOL 位置高且比较器输入将 OCVPO 位电平由高强制拉低，若相关中断控制位使能，则将产生相应的中断。值得注意的是，只有 OCVPINT 上升沿才能触发 OCVP 中断请求，所以 OCVPSPOL 位事先必须按照用户应用要求进行正确的配置。OCVP 电路中的比较器还具有迟滞功能，由 OCVPCHY 位控制。

注意去抖动时钟 f_{DEB} 来自于系统时钟 f_{SYS} 。A/D 转换器也可通过 OCVPAO 信号读取到运算放大器输出电压。D/A 转换器输出电压由 OCVPDA 寄存器控制，且定义如下：

$$DAC V_{OUT} = (DAC \text{ 参考电压} / 256) \times OCVPDA[7:0]$$

OCVP 控制寄存器

OCVP 功能的所有操作由一系列的寄存器控制。CTRL2 寄存器用于选择 OCVPAI0 输入信号源引脚。一个寄存器用于为 OCVP 电路提供参考电压。两个寄存器用于运算放大器和比较器输入失调校准。剩下的三个寄存器为控制寄存器，用于控制 OCVP 功能，D/A 转换器参考电压选择，开关 on/off 控制，PGA 增益选择，比较器同相输入选择，比较器去抖动时间，比较器迟滞功能和比较器输出极性控制等。关于输入失调电压校准步骤的更多细节描述，参见相应的输入失调校准章节。

寄存器名称	位							
	7	6	5	4	3	2	1	0
CTRL2	FPTMEN	FPTMF	—	—	—	—	OCVPS1	OCVPS0
OCVPDA	D7	D6	D5	D4	D3	D2	D1	D0
OCVPC0	OCVPSW7	OCVPSW6	OCVPSW5	OCVPSW4	OCVPSW3	OCVPSW2	OCVPSW1	OCVPSW0
OCVPC1	OCVPEN	OCVPCHY	OCVPO	OCVPSPOL	—	—	OCVPCPS	OCVPVRS
OCVPC2	—	OCVPG2XEN	OCVPG2	OCVPG1	OCVPG0	OCVPDEB2	OCVPDEB1	OCVPDEB0
OCVPOCAL	OCVPOOFM	OCVPORSP	OCVPOOF5	OCVPOOF4	OCVPOOF3	OCVPOOF2	OCVPOOF1	OCVPOOF0
OCVPCCAL	OCVPCOUT	OCVPCOFM	OCVPCRSF	OCVPCOF4	OCVPCOF3	OCVPCOF2	OCVPCOF1	OCVPCOF0

OCVP 寄存器列表

• CTRL2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	FPTMEN	FPTMF	—	—	—	—	OCVPS1	OCVPS0
R/W	R/W	R	—	—	—	—	R/W	R/W
POR	1	0	—	—	—	—	0	0

Bit 7 **FPTMEN**: PTMOSC 振荡器使能控制位
详见振荡器章节。

Bit 6 **FPTMF**: PTMOSC 振荡器稳定标志位
详见振荡器章节。

- Bit 5~2 未定义，读为“0”
- Bit 1~0 **OCVPS1~OCVPS0**: OCVPAI0 信号输入源引脚选择位
 00: PB0
 01: PB1
 10: PB2
 11: PB3

● **OCVPC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OCVPSW7	OCVPSW6	OCVPSW5	OCVPSW4	OCVPSW3	OCVPSW2	OCVPSW1	OCVPSW0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	0	0	1	0	0	0

- Bit 7 **OCVPSW7**: OCVP 开关 S7 on/off 控制位
 0: Off
 1: On
- Bit 6 **OCVPSW6**: OCVP 开关 S6 on/off 控制位
 0: Off
 1: On
- Bit 5 **OCVPSW5**: OCVP 开关 S5 on/off 控制位
 0: Off
 1: On
- Bit 4 **OCVPSW4**: OCVP 开关 S4 on/off 控制位
 0: Off
 1: On
- Bit 3 **OCVPSW3**: OCVP 开关 S3 on/off 控制位
 0: Off
 1: On
- Bit 2 **OCVPSW2**: OCVP 开关 S2 on/off 控制位
 0: Off
 1: On
- Bit 1 **OCVPSW1**: OCVP 开关 S1 on/off 控制位
 0: Off
 1: On
- Bit 0 **OCVPSW0**: OCVP 开关 S0 on/off 控制位
 0: Off
 1: On

● **OCVPC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OCVPEN	OCVPCHY	OCVPO	OCVPSPOL	—	—	OCVPCPS	OCVPVRS
R/W	R/W	R/W	R	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

- Bit 7 **OCVPEN**: OCVP 功能使能控制位
 0: 除能
 1: 使能
 当此位清零，OCVP 所有操作将除能，比较器输出 OCVPCOUT 等于 0。
- Bit 6 **OCVPCHY**: OCVP 比较器迟滞功能控制位
 0: 除能
 1: 使能

- Bit 5 **OCVPO**: OCV 比较器去抖动输出
OCVPO 为 OCVPCOUT 位输出的去抖动后输出。
- Bit 4 **OCVPSPOL**: OCV 去抖动后输出信号极性控制位
0: 同相
1: 反相
- Bit 3~2 未定义, 读为 “0”
- Bit 1 **OCVPCPS**: OCV 比较器同相输入选择位
0: 来自 OCVPAO 信号与或 OPAMP 输出
1: 来自 OCVPCI 引脚
- Bit 0 **OCVPVRS**: OCV D/A 转换器参考电压选择位
0: 来自 V_{DD}
1: 来自 VREF 引脚

• OCVPC2 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	OCVPG2XEN	OCVPG2	OCVPG1	OCVPG0	OCVPDEB2	OCVPDEB1	OCVPDEB0
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 未定义, 读为 “0”
- Bit 6 **OCVPG2XEN**: R2/R1 比率加倍使能控制位
0: 除能 ($R1=13.2k\Omega$)
1: 使能 ($R1=6.6k\Omega$)
当此位置 1, 由 OCVPG2~OCVPG0 位段所选择的 R2/R1 比率将加倍。
- Bit 5~3 **OCVPG2~OCVPG0**: PGA R2/R1 比率选择位
当 OCVPG2XEN=0:
000: $R2/R1=1$
001: $R2/R1=4$
010: $R2/R1=6$
011: $R2/R1=10$
100: $R2/R1=15$
101: $R2/R1=25$
110: $R2/R1=40$
111: $R2/R1=65$
当 OCVPG2XEN=1:
000: $R2/R1=2$
001: $R2/R1=8$
010: $R2/R1=12$
011: $R2/R1=20$
100: $R2/R1=30$
101: $R2/R1=50$
110: $R2/R1=80$
111: $R2/R1=130$
反相和同相模式下的 PGA 增益计算公式将在“输入电压范围”章节内介绍。
当增益由此位段决定时需用到内部电阻 R1 和 R2。这就意味着开关 S4 或 S5, 以及开关 S7 应为 on, 否则无法保证增益的精确。
- Bit 2~0 **OCVPDEB2~OCVPDEB0**: OCV 比较器输出去抖动时间控制位
000: 旁路, 无去抖动
001: $(1\sim2)\times t_{DEB}$
010: $(3\sim4)\times t_{DEB}$
011: $(7\sim8)\times t_{DEB}$
100: $(15\sim16)\times t_{DEB}$
101: $(31\sim32)\times t_{DEB}$

110: $(63 \sim 64) \times t_{DEB}$
111: $(127 \sim 128) \times t_{DEB}$
注: $f_{DEB} = f_{SYS}$, $t_{DEB} = 1/f_{DEB}$

• OCVPDA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 OCVP D/A 转换器数据寄存器 bit 7~bit 0
OCVP D/A 转换器输出 = (DAC 参考电压 / 256) $\times$ OCVPDA[7:0]

• OCVPOCAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	OCVPOOFM	OCVPORSP	OCVPOOF5	OCVPOOF4	OCVPOOF3	OCVPOOF2	OCVPOOF1	OCVPOOF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	0	0	0	0	0

Bit 7 **OCVPOOFM**: OCVP 运算放大器工作模式选择位
0: 正常工作模式
1: 失调校准模式
此位用来选择 OCVP 工作模式。要选择运算放大器输入失调校准模式, OCVPSW7~OCVPSW0 位段必须先设为 28H, OCVPOOFM 位必须设为 1, 接着 OCVPCOFM 清零。失调校准步骤详情参见“运算放大器输入失调校准”章节。

Bit 6 **OCVPORSP**: OCVP 运算放大器输入失调电压校准参考选择位
0: 选择运算放大器反相输入
1: 选择运算放大器同相输入

Bit 5~0 **OCVPOOF5~OCVPOOF0**: OCVP 运算放大器输入失调电压校准值
此 6-bit 位段用于执行运算放大器输入失调电压校准操作, 用于该操作的值可以重复存入此位段。更多详情参见“运算放大器输入失调电压校准”章节。

• OCVPCCAL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	OCVPCOUT	OCVPCOFM	OCVPCRSP	OCVPCOF4	OCVPCOF3	OCVPCOF2	OCVPCOF1	OCVPCOF0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	0	0	0

Bit 7 **OCVPCOUT**: OCVP 比较器输出
0: 同相输入电压 < DAC 输出电压
1: 同相输入电压 > DAC 输出电压
此位用于说明同相输入电压是否大于 DAC 输出电压。若 OCVPCOUT 置 1, 同相输入电压大于 DAC 输出电压。反之, 同相输入电压小于 DAC 输出电压。

Bit 6 **OCVPCOFM**: OCVP 比较器工作模式选择位
0: 正常工作模式
1: 失调校准模式
此位用于选择 OCVP 比较器工作模式。要选择比较器输入失调校准模式, OCVPSW7~OCVPSW0 位段必须先设置为 28H, OCVPCOFM 位必须置 1, 接着 OCVPOOFM 位清零。失调校准步骤详情参见“比较器输入失调校准”章节。

Bit 5 **OCVPCRSP**: OCVP 比较器输入失调电压校准参考选择位
0: 反相输入作为参考输入
1: 同相输入作为参考输入

- Bit 4~0 **OCVPCOF4~OCVPCOF0**: OCV 比较器输入失调电压校准值
此 5-bit 位段用于执行比较器输入失调电压校准操作，用于该操作的值可以重复存入此位段。更多详情参见“比较器输入失调电压校准”章节。

输入电压范围

为了操作的灵活性，在不同的 PGA 工作模式下，输入电压可以为正或者为负。正输入或负输入电压的 PGA 输出基于不同的公式计算如下：

- 输入电压 $V_{IN} > 0$ 时，PGA 工作在同相模式下，PGA 输出可由以下公式获得：

$$V_{OUT} = (1 + R_2/R_1) \times V_{IN}$$

- 当 PGA 工作在同相模式下，将作为单位增益缓冲器。若 OCVPSW6~OCVPSW4 位段的值设为“000”，

- PGA 增益将为“1”，此时 PGA 作为单位增益缓冲器。内部开关 S6、S5 和 S4 将为 off，PGA 输出电压为：

$$V_{OUT} = V_{IN}$$

- 若 S3 和 S4 on，输入结点是 OCVPAI0。输入电压 $0 > V_{IN} > -0.4V$ 时，工作在反相模式下的 PGA 以及 PGA 输出可由以下公式获得。注意，若输入电压为负，其值不可小于 -0.4V，否则会漏电流。

$$V_{OUT} = -(R_2/R_1) \times V_{IN}$$

失调校准

要工作在 OCV 电路输入失调校准模式，OCVPSW7~OCVPSW0 位段必须先设置为 28H。除相关控制位设定外，运算放大器和比较器的输入失调校准步骤相似。

运算放大器输入失调校准

步骤 1. 设置 OCVPSW[7:0]=28H (S3 和 S5 on，其它开关 off)，OCVPOOFM=1，OCVPCOFM=0 且 OCVPORSP=1，OCVPCPS=0，OCVP 将工作在运算放大器输入失调校准模式。

步骤 2. 设置 OCVPDA[7:0]=40H

步骤 3. 设置 OCVPOOF[5:0]=000000，此时开始读 OCVPCOUT 位。

步骤 4. OCVPOOF[5:0] 的值加一，读 OCVPCOUT 位。

若 OCVPCOUT 位状态不变，则重复步骤 4 直到 OCVPCOUT 位状态改变。

若 OCVPCOUT 位状态改变，则记录下 OCVPOOF 值为 VOOS1，前往步骤 5。

步骤 5. 设置 OCVPOOF[5:0]=111111，此时开始读 OCVPCOUT 位。

步骤 6. OCVPOOF[5:0] 的值减一，读 OCVPCOUT 位。

若 OCVPCOUT 位状态未改变，重复步骤 6 直到 OCVPCOUT 位状态改变。

若 OCVPCOUT 位状态改变，则记录下 OCVPOOF 值为 VOOS2，前往步骤 7。

步骤 7. 将运算放大器输入失调校准值 VOOS，重复存入 OCVPOOF[5:0] 位段。此时失调校准步骤完成。

$$V_{OOS} = (V_{OOS1}/V_{OOS2})/2$$

比较器输入失调校准

在失调校准前，应通过设置 OCVPCHY 位为零使得迟滞电压为零。

步骤 1. 设置 OCVPSW[7:0]=28H, OCVPCOFM=1, OCVPOOFM=0 且 OCVPCRSP=0, OCVPC 将工作在比较器输入失调校准模式。

步骤 2. 设置 OCVPDA[7:0]=40H

步骤 3. 设置 OCVPCOF[4:0]=00000, 此时开始读 OCVPCOUT 位。

步骤 4. OCVPCOF[4:0] 的值加一，读 OCVPCOUT 位。

若 OCVPCOUT 位状态不变，则重复步骤 4 直到 OCVPCOUT 位状态改变。

若 OCVPCOUT 位状态改变，则记录下 OCVPCOF 值为 VCOS1，前往步骤 5。

步骤 5. 设置 OCVPCOF[4:0]=11111，此时开始读 OCVPCOUT 位。

步骤 6. OCVPCOF[4:0] 的值减一，读 OCVPCOUT 位。

若 OCVPCOUT 位状态未改变，重复步骤 6 直到 OCVPCOUT 位状态改变。

若 OCVPCOUT 位状态改变，则记录下 OCVPCOF 值为 VCOS2，前往步骤 7。

步骤 7. 将比较器输入失调校准值 VCOS，重复存入 OCVPCOF[4:0] 位段。此时失调校准步骤完成。

$$V_{COS} = (V_{COS1} / V_{COS2}) / 2$$

触控按键功能

该单片机提供 4 个触控按键功能。该触控按键功能完全内部集成不需外接元件，通过内部寄存器对其进行简单的操作。

触控按键结构

触控按键引脚与 I/O 引脚共用。通过对应的引脚共用功能控制寄存器的位来选择此功能。按键被分成一个组，称为一个模块。该模块为独立的一组包含四个触控按键且每个模块有各自的振荡器。该模块具有单独的控制逻辑电路和配套的寄存器系列。

按键总数	触控按键	共用 I/O 引脚
4	KEY1~KEY4	PA7, PA4, PA1, PA3

触控按键结构

触控按键寄存器定义

触控按键模块 0 包含 4 个触控按键功能，且都有其配套的寄存器。以下表格显示了触控按键模块 0 的寄存器系列。

寄存器名称	说明
TKTMR	触控按键 8-bit 时隙计数器预载寄存器
TKC0	触控按键功能控制寄存器 0
TKC1	触控按键功能控制寄存器 1
TK16DL	触控按键功能 16-bit 计数器低字节
TK16DH	触控按键功能 16-bit 计数器高字节
TKM016DL	触控按键模块 0 16-bit C/F 计数器低字节

寄存器名称	说明
TKM016DH	触控按键模块 0 16-bit C/F 计数器高字节
TKM0ROL	触控按键模块 0 参考振荡器电容选择低字节
TKM0ROH	触控按键模块 0 参考振荡器电容选择高字节
TKM0C0	触控按键模块 0 控制寄存器 0
TKM0C1	触控按键模块 0 控制寄存器 1

触控按键功能寄存器定义

寄存器名称	位							
	7	6	5	4	3	2	1	0
TKTMR	D7	D6	D5	D4	D3	D2	D1	D0
TKC0	—	TKRCOV	TKST	TKCFOV	TK16OV	—	TK16S1	TK16S0
TKC1	—	—	—	—	—	—	TKFS1	TKFS0
TK16DL	D7	D6	D5	D4	D3	D2	D1	D0
TK16DH	D15	D14	D13	D12	D11	D10	D9	D8
TKM016DL	D7	D6	D5	D4	D3	D2	D1	D0
TKM016DH	D15	D14	D13	D12	D11	D10	D9	D8
TKM0ROL	D7	D6	D5	D4	D3	D2	D1	D0
TKM0ROH	—	—	—	—	—	—	D9	D8
TKM0C0	M0MXS1	M0MXS0	M0DFEN	M0FILEN	M0SOFC	M0SOF2	M0SOF1	M0SOF0
TKM0C1	M0TSS	—	M0ROEN	M0KOEN	M0K4EN	M0K3EN	M0K2EN	M0K1EN

触控按键功能寄存器列表

• TKTMR 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0:** 触控按键 8-bit 时隙计数器预载寄存器

触控按键时隙计数器预载寄存器用于确定触控按键时隙溢出时间。时隙单元周期为 32 个时隙时钟周期，通过一个 5-bit 计数器获得。因此，时隙计数器溢出时间可由下面的等式算出。

时隙计数器溢出时间 = $(256 - \text{TKTMR}[7:0]) \times 32t_{\text{TSC}}$ ，此处的 t_{TSC} 为时隙计数器时钟周期。

• TKC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	TKRCOV	TKST	TKCFOV	TK16OV	—	TK16S1	TK16S0
R/W	—	R/W	R/W	R/W	R/W	—	R/W	R/W
POR	—	0	0	0	0	—	0	0

Bit 7 未定义，读为“0”

Bit 6 **TKRCOV:** 触控按键时隙计数器溢出标志位

0: 无溢出

1: 溢出

此位可通过应用程序读 / 写。当触控按键时隙计数器溢出将此位置为“1”时，

相应的触控按键中断请求标志位也会同时置位。然而若是通过应用程序将此位设置为“1”时，相应的触控按键中断请求标志位不会受到影响。因此，该位不能通过应用程序置位，但必须通过应用程序清零。

如果时隙计数器溢出，TKRCOV 位及触控按键中断请求标志位 TKMF 将被置位同时所有模块按键振荡器和参考振荡器自动停止。触控按键模块 0 的 16 位 C/F 计数器、触控按键功能 16 位计数器、5 位时隙单位周期计数器和 8 位时隙定时计数器也都会自动关闭。

Bit 5 **TKST**: 触控按键检测开启控制位

0: 检测停止或无操作

0→1: 启动检测

当该位为“0”时，触控按键模块 0 的 16 位 C/F 计数器、触控按键功能 16 位计数器和 5 位时隙单位周期计数器会自动清零但 8 位可编程时隙定时计数器不会被清零。当该位由 0→1 时，16 位 C/F 计数器、触控按键功能 16 位计数器、5 位时隙单位周期计数器和 8 位时隙定时计数器都会自动开启，并使能按键振荡器和参考振荡器以驱动相应的计数器。

Bit 4 **TKCFOV**: 触控按键模块 0 16 位 C/F 计数器溢出标志位

0: 无溢出

1: 溢出

该位由触控按键模块 0 16 位 C/F 计数器溢出置位，必须通过应用程序清零。

Bit 3 **TK16OV**: 触控按键功能 16 位计数器溢出标志位

0: 无溢出

1: 溢出

该位由触控按键功能 16 位计数器溢出置位，必须通过应用程序清零。

Bit 2 未定义，读为“0”

Bit 1~0 **TK16S1~TK16S0**: 触控按键功能 16-bit 计数器时钟源选择位

00: f_{SYS}

01: $f_{SYS}/2$

10: $f_{SYS}/4$

11: $f_{SYS}/8$

• TKC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	TKFS1	TKFS0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

Bit 7~2 未定义，读为“0”

Bit 1~0 **TKFS1~TKFS0**: 触控按键振荡器和参考振荡器频率选择位

00: 1MHz

01: 3MHz

10: 7MHz

11: 11MHz

• TK16DL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**: 触控按键功能 16-bit 计数器低字节内容

● TK16DH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 D15~D8: 触控按键功能 16-bit 计数器高字节内容

该寄存器对用于存储触控按键功能 16-bit 计数器值。该 16-bit 计数器用于校准参考振荡器或按键振荡器频率。如果触控按键时隙计数器溢出，此 16-bit 计数器将停止，但不会改变当前值。当 TKST 位为“0”时，该寄存器对被清零。

● TKM016DL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0: 触控按键模块 0 16-bit C/F 计数器低字节内容

● TKM016DH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 D15~D8: 触控按键模块 0 16-bit C/F 计数器高字节内容

该寄存器对用于存储触控按键模块 0 16-bit C/F 计数器值。如果触控按键时隙计数器溢出，该 16-bit C/F 计数器将停止，但不会改变当前值。当 TKST 位为“0”时，该寄存器对被清零。

● TKM0ROL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0: 触控按键模块 0 参考振荡器内部电容选择

● TKM0ROH 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定义，读为“0”

Bit 1~0 D9~D8: 触控按键模块 0 参考振荡器内部电容选择

该寄存器对用于存储触控按键模块 0 参考振荡器电容值。

参考振荡器内接电容值 = (TKMRO[9:0]×50pF)/1024。

● TKM0C0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	M0MXS1	M0MXS0	M0DFEN	M0FILEN	M0SOFC	M0SOF2	M0SOF1	M0SOF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7~6 **M0MXS1~M0MXS0**: 触控按键模块 0 多路复用按键选择
00: KEY1
01: KEY2
10: KEY3
11: KEY4
- Bit 5 **M0DFEN**: 触控按键模块 0 倍频功能控制位
0: 除能
1: 使能
此位用于控制触控按键振荡器频率的倍频功能。当此位置“1”，按键振荡器频率将为原来的倍频。
- Bit 4 **M0FILEN**: 触控按键模块 0 滤波器功能控制
0: 除能
1: 使能
- Bit 3 **M0SOFC**: 触控按键模块 0 C/F 振荡器跳频功能选择位
0: 软件处理跳频功能，由 M0SOF2~M0SOF0 位决定
1: 硬件处理跳频功能，M0SOF2~M0SOF0 位无作用
该位用来选择触控按键振荡器跳频功能控制方式，当此位置 1，按键振荡器跳频功能由硬件电路控制，而不受 M0SOF2~M0SOF0 位影响。
- Bit 2~0 **M0SOF2~M0SOF0**: 触控按键模块 0 参考和按键振荡器跳频选择位
000: 1.020MHz
001: 1.040MHz
010: 1.059MHz
011: 1.074MHz
100: 1.085MHz
101: 1.099MHz
110: 1.111MHz
111: 1.125MHz
这些位用于为跳频功能选择触控按键振荡器频率。应注意的是这些位仅在 M0SOFC 位清零时有效。
上述频率会随着外部或内部电容值的不同而变化。若触控按键振荡器频率选择 1MHz，用户选择其它频率时，可依此比例调整。

● TKM0C1 寄存器

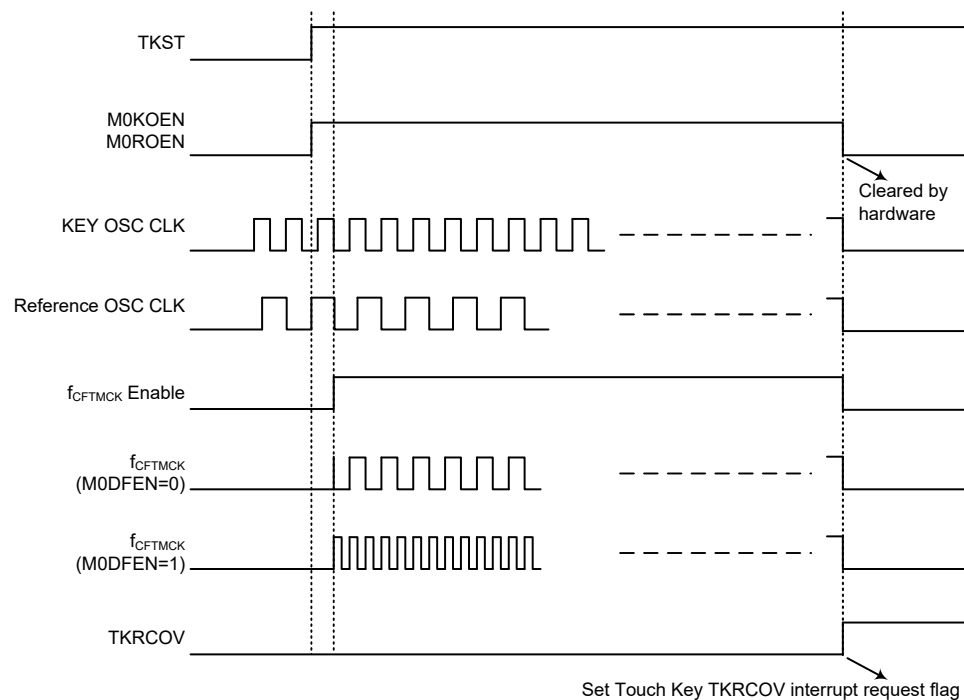
Bit	7	6	5	4	3	2	1	0
Name	M0TSS	—	M0ROEN	M0KOEN	M0K4EN	M0K3EN	M0K2EN	M0K1EN
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	—	0	0	0	0	0	0

- Bit 7 **M0TSS**: 触控按键模块 0 时隙计数器时钟源选择位
0: 触控按键模块 0 参考振荡器
1: $f_{sys}/4$
- Bit 6 未定义，读为“0”
- Bit 5 **M0ROEN**: 触控按键模块 0 参考振荡器使能控制位
0: 除能
1: 使能
该位用于使能触控按键模块 0 参考振荡器。
若选择使用参考振荡器，在将 TKST 位设置为低到高前，必须先使能参考振荡器。

Bit 4	M0KOEN: 触控按键模块 0 按键振荡器使能控制位 0: 除能 1: 使能 该位用于使能触控按键模块 0 按键振荡器。 若使能相应的按键进行扫描时, 在将 TKST 位设置为低到高前, 必须先使能按键振荡器。
Bit 3	M0K4EN: 触控按键模块 0 KEY 4 使能控制 0: 除能 1: 使能
Bit 2	M0K3EN: 触控按键模块 0 KEY 3 使能控制 0: 除能 1: 使能
Bit 1	M0K2EN: 触控按键模块 0 KEY 2 使能控制 0: 除能 1: 使能
Bit 0	M0K1EN: 触控按键模块 0 KEY 1 使能控制 0: 除能 1: 使能

触控按键操作

手指接近或接触到触控面板时, 面板的电容量会增大, 电容量的变化会轻微改变内部感应振荡器的频率, 通过测量频率的变化可以感知触控动作。参考时钟通过内部可编程分频器能够产生一个固定的时间周期。在这个时间周期内, 通过在此固定时间周期内对感应振荡器产生的时钟周期计数, 可确定触控按键的动作。



触控按键模 0 块时序图

触控按键模块 0 包含四个与 I/O 引脚共用的触控按键 KEY1~KEY4。通过相关的引脚共用功能控制寄存器位可选择触控按键引脚功能。触控按键具有独立的感应振荡器，因此触控按键模块 0 包含四个感应振荡器。

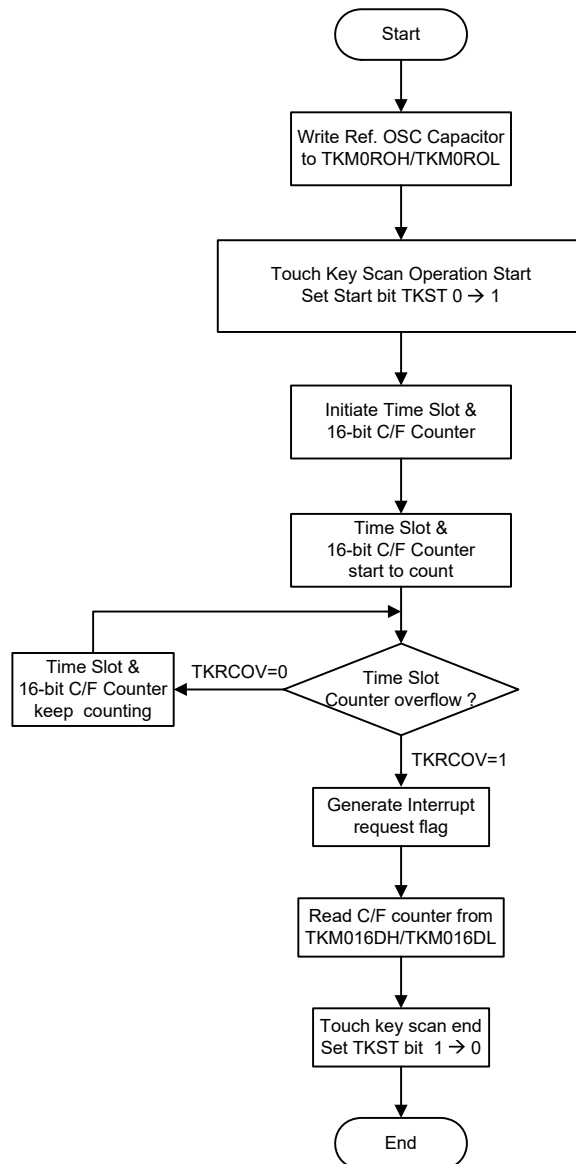
在参考时钟固定的时间间隔内，感应振荡器产生的时钟周期数是可以测量的。测到的周期数可以用于判断触控动作是否有效发生。在此固定的时间间隔最后，会产生一个触控按键中断信号。

当 TKST 位被清零时，触控按键模块 0 的 16 位 C/F 计数器、16 位计数器和 5 位时隙单位周期计数器会自动清零，而 8 位可编程时隙计数器不清零，由用户设置溢出时间。当 TKST 位置高时，16 位 C/F 计数器、16 位计数器、5 位时隙单位周期计数器和 8 位时隙定时计数器会自动开启。

如果时隙计数器溢出，触控按键模块 0 的按键振荡器和参考振荡器都会自动停止且 16 位 C/F 计数器、16 位计数器、5 位时隙单位周期计数器和 8 位时隙定时计数器会自动停止。时隙计数器时钟源可通过 TKM0C1 寄存器 MTSS 位选择来自参考振荡器或 $f_{SYS}/4$ 。通过设置 TKM0C1 寄存器中的 M0ROEN 位和 M0KOEN 为“1”，可使能参考振荡器和按键振荡器。

当触控按键模块 0 的时隙计数器溢出，产生触控按键中断。这里所有的触控按键是指已使能的触控按键。

触控按键扫描操作流程



触控按键扫描操作流程

触控按键中断

触控按键只有一个中断。当触控按键模块 0 的时隙计数器溢出时，才会产生触控按键中断，注意，此处提到的触控按键是指已被使能的触控按键。此时 16 位 C/F 计数器、16 位计数器，5 位时隙单位周期计数器和 8 位时隙定时计数器会自动清零。

若触控按键模块 0 的 16 位 C/F 计数器溢出，16 位 C/F 计数器溢出标志位将置高，此中断标志位不会自动复位，必须通过应用程序将其清零。

若 16 位计数器溢出，16 位计数器溢出标志位 TK16OV 将置高，此中断标志位不会自动复位，必须通过应用程序将其清零。触控按键中断的详细说明见中断部分。

编程注意事项

相关寄存器设置后，将 TKST 位由低电平变为高电平会启动触控按键检测程序初始化。此时所有相关的振荡器将同步使能。当计数器溢出时，时隙计数器标志位 TKRCOV 将变为高电平。计数器溢出发生时，会产生一个中断信号。

当外部触控按键的大小和布局确定时，其相关的电容将决定感应振荡器的频率。

低电压检测 – LVD

该单片机具有低电压检测功能，即 LVD。该功能使能用于监测电源电压 V_{DD} ，若电源电压低于一定值可提供一个警告信号。此功能在电池类产品中非常有用，在电池电压较低时产生警告信号。低电压检测也可产生中断信号。

LVD 寄存器

低电压检测功能由 LVDC 寄存器控制。VLVD2~VLVD0 位用于选择 8 个固定电压中的一个参考点。LVDO 位被置位时低电压情况发生，若 LVDO 位为低表明 V_{DD} 电压工作在当前所设置低电压水平值之上。LVDEN 位用于控制低电压检测功能的开启 / 关闭，设置此位为高使能低电压检测功能，反之，关闭内部低电压检测电路。低电压检测会有一定的功耗，在不使用时可考虑关闭此功能，此举在功耗要求严格的电池供电应用中值得考虑。

• LVDC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	—	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

Bit 7~6 未定义，读为“0”

Bit 5 **LVDO**: LVD 输出标志位

0: 未检测到低电压

1: 检测到低电压

Bit 4 **LVDEN**: 低电压检测控制位

0: 除能

1: 使能

Bit 3 未定义，读为“0”

Bit 2~0 **VLVD2~VLVD0**: 选择 LVD 电压位

000: 2.0V

001: 2.2V

010: 2.4V

011: 2.7V

100: 3.0V

101: 3.3V

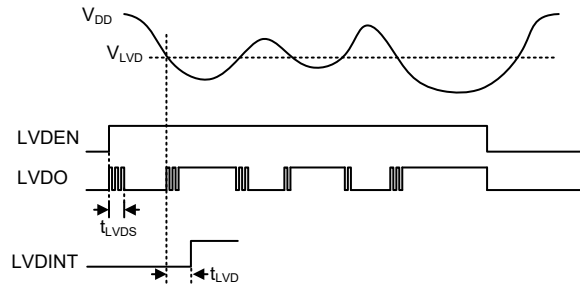
110: 3.6V

111: 4.0V

LVD 操作

通过比较电源电压 V_{DD} 与存储在 LVDC 寄存器中的预置电压值的结果，低电压检测功能工作。其设置的范围为 2.0V~4.0V。当电源电压 V_{DD} 低于预置电压值时，LVDO 位被置为高，表明低电压产生。低电压检测功能由一个自动使能的参考电压提供。低电压检测器使能后，读取 LVDO 位前，电路稳定需要一定的延时 t_{LVDs} 。注意， V_{DD} 电压可能上升或下降比较缓慢，在 V_{LVD} 电压值附近时，

LVDO 位可能有多种变化。



LVD 操作

低电压检测器也有自己的中断功能，也是属于多功能中断的一种，它是除了轮询 LVDO 位之外的另一种检测低电压的方法。检测到低电压，置位 LVDO 并延时 t_{LVD} 后，中断产生。此种情况下，若 V_{DD} 降至小于 LVD 预置电压值时，中断请求标志位 LVF 将被置位，中断产生，单片机将从空闲或休眠模式中被唤醒。若不要求低电压检测的唤醒功能使能，在单片机进入空闲或休眠模式前应将 LVF 标志置为高。

中断

中断是单片机的一个重要功能。当外部事件或内部功能如定时器模块有效，并且产生中断时，系统会暂时中止当前的程序而转到执行相对应的中断服务程序。该单片机提供多个外部中断和内部中断功能，外部中断由 INT0 和 INT1 引脚动作产生，而内部中断由各种内部功能，如定时器模块、过电流 / 电压保护功能 (OCVP)、时基、LVD、EEPROM、触控按键功能和 A/D 转换器产生。

中断寄存器

中断控制基本上是在一定单片机条件发生时设置请求标志位，应用程序中中断使能位的设置是通过位于特殊功能数据存储单元中的一系列寄存器控制的，如下表所示。中断寄存器分为三类。第一类是 INTC0~INTC2 寄存器，用于设置基本的中断；第二类是 MFIO~MFI3 寄存器，用于设置多功能中断；最后一类是 INTEG 寄存器，用于设置外部中断边沿触发类型。

寄存器中含有中断控制位和中断请求标志位。中断控制位用于使能或除能各种中断，中断请求标志位用于存放当前中断请求的状态。它们都按照特定的模式命名，前面表示中断类型的缩写，紧接着是中断的序号 (可选)，后面跟着的字母 “E” 代表使能 / 除能位，“F” 代表请求标志位。

功能	使能位	请求标志	注释
总中断	EMI	—	—
OCVP	OCVPE	OCVPF	—
INTn 引脚	INTnE	INTnF	n=0~1
多功能	MFnE	MFnF	n=0~3
时基	TBnE	TBnF	n=0~1
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
触控按键模块 0	TKME	TKMF	—
A/D 转换器	ADE	ADF	—

功能	使能位	请求标志	注释
定时器模块	CTMnAE	CTMnAF	n=0~2
	CTMnPE	CTMnPF	
	STMAE	STMAF	—
	STMPE	STMPF	
	PTMAE	PTMAF	—
	PTMPE	PTMPF	

中断寄存器位命名模式

寄存器名称	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	MF0F	INT0F	OCVPF	MF0E	INT0E	OCVPE	EMI
INTC1	TB0F	MF3F	MF2F	MF1F	TB0E	MF3E	MF2E	MF1E
INTC2	ADF	TKMF	INT1F	TB1F	ADE	TKME	INT1E	TB1E
MF10	CTM1AF	CTM1PF	CTM0AF	CTM0PF	CTM1AE	CTM1PE	CTM0AE	CTM0PE
MF11	CTM2AF	CTM2PF	STMAF	STMPF	CTM2AE	CTM2PE	STMAE	STMPE
MF12	—	—	PTMAF	PTMPF	—	—	PTMAE	PTMPE
MF13	—	—	DEF	LVF	—	—	DEE	LVE

中断寄存器列表

• INTEG 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定义，读为“0”

Bit 3~2 **INT1S1~INT1S0**: INT1 脚中断有效边沿控制位

00: 除能中断
01: 上升沿中断
10: 下降沿中断
11: 双沿中断

Bit 1~0 **INT0S1~INT0S0**: INT0 脚中断有效边沿控制位

00: 除能中断
01: 上升沿中断
10: 下降沿中断
11: 双沿中断

• INTC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	MF0F	INT0F	OCVPF	MF0E	INT0E	OCVPE	EMI
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 未定义，读为“0”
- Bit 6 **MF0F**: 多功能中断 0 请求标志位
0: 无请求
1: 中断请求
- Bit 5 **INT0F**: 外部中断 0 请求标志位
0: 无请求
1: 中断请求
- Bit 4 **OCVPF**: 过电流 / 电压保护中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **MF0E**: 多功能中断 0 控制位
0: 除能
1: 使能
- Bit 2 **INT0E**: 外部中断 0 控制位
0: 除能
1: 使能
- Bit 1 **OCVPE**: 过电流 / 电压保护中断控制位
0: 除能
1: 使能
- Bit 0 **EMI**: 总中断控制位
0: 除能
1: 使能

• INTC1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TB0F	MF3F	MF2F	MF1F	TB0E	MF3E	MF2E	MF1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **TB0F**: 时基 0 中断请求标志位
0: 无请求
1: 中断请求
- Bit 6 **MF3F**: 多功能中断 3 请求标志位
0: 无请求
1: 中断请求
- Bit 5 **MF2F**: 多功能中断 2 请求标志位
0: 无请求
1: 中断请求
- Bit 4 **MF1F**: 多功能中断 1 请求标志位
0: 无请求
1: 中断请求
- Bit 3 **TB0E**: 时基 0 中断控制位
0: 除能
1: 使能

- Bit 2 **MF3E**: 多功能中断 3 控制位
0: 除能
1: 使能
- Bit 1 **MF2E**: 多功能中断 2 控制位
0: 除能
1: 使能
- Bit 0 **MF1E**: 多功能中断 1 控制位
0: 除能
1: 使能

● **INTC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	ADF	TKMF	INT1F	TB1F	ADE	TKME	INT1E	TB1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **ADF**: A/D 转换器中断请求标志位
0: 无请求
1: 中断请求
- Bit 6 **TKMF**: 触控按键中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 **INT1F**: 外部中断 1 请求标志位
0: 无请求
1: 中断请求
- Bit 4 **TB1F**: 时基 1 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **ADE**: A/D 转换器中断控制位
0: 除能
1: 使能
- Bit 2 **TKME**: 触控按键中断控制位
0: 除能
1: 使能
- Bit 1 **INT1E**: 外部中断 1 控制位
0: 除能
1: 使能
- Bit 0 **TB1E**: 时基 1 中断控制位
0: 除能
1: 使能

● MF10 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTM1AF	CTM1PF	CTM0AF	CTM0PF	CTM1AE	CTM1PE	CTM0AE	CTM0PE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **CTM1AF**: CTM1 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 6 **CTM1PF**: CTM1 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 **CTM0AF**: CTM0 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **CTM0PF**: CTM0 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **CTM1AE**: CTM1 比较器 A 匹配中断控制位
0: 除能
1: 使能
- Bit 2 **CTM1PE**: CTM1 比较器 P 匹配中断控制位
0: 除能
1: 使能
- Bit 1 **CTM0AE**: CTM0 比较器 A 匹配中断控制位
0: 除能
1: 使能
- Bit 0 **CTM0PE**: CTM0 比较器 P 匹配中断控制位
0: 除能
1: 使能

● MF11 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CTM2AF	CTM2PF	STMAF	STMPF	CTM2AE	CTM2PE	STMAE	STMPE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

- Bit 7 **CTM2AF**: CTM2 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 6 **CTM2PF**: CTM2 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 5 **STMAF**: STM 比较器 A 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **STMPF**: STM 比较器 P 匹配中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 **CTM2AE**: CTM2 比较器 A 匹配中断控制位
0: 除能
1: 使能

- Bit 2 **CTM2PE**: CTM2 比较器 P 匹配中断控制位
0: 除能
1: 使能
- Bit 1 **STMAE**: STM 比较器 A 匹配中断控制位
0: 除能
1: 使能
- Bit 0 **STMPE**: STM 比较器 P 匹配中断控制位
0: 除能
1: 使能

● **MF12 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	PTMAF	PTMPF	—	—	PTMAE	PTMPE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义，读为“0”
- Bit 5 **PTMAF**: PTM 比较器 A 匹配请求标志位
0: 无请求
1: 中断请求
- Bit 4 **PTMPF**: PTM 比较器 P 匹配请求标志位
0: 无请求
1: 中断请求
- Bit 3~2 未定义，读为“0”
- Bit 1 **PTMAE**: PTM 比较器 A 匹配中断控制位
0: 除能
1: 使能
- Bit 0 **PTMPE**: PTM 比较器 P 匹配中断控制位
0: 除能
1: 使能

● **MF13 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	DEF	LVF	—	—	DEE	LVE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

- Bit 7~6 未定义，读为“0”
- Bit 5 **DEF**: 数据 EEPROM 中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **LVF**: LVD 中断请求标志位
0: 无请求
1: 中断请求
- Bit 3~2 未定义，读为“0”
- Bit 1 **DEE**: 数据 EEPROM 中断控制位
0: 除能
1: 使能
- Bit 0 **LVE**: LVD 中断控制位
0: 除能
1: 使能

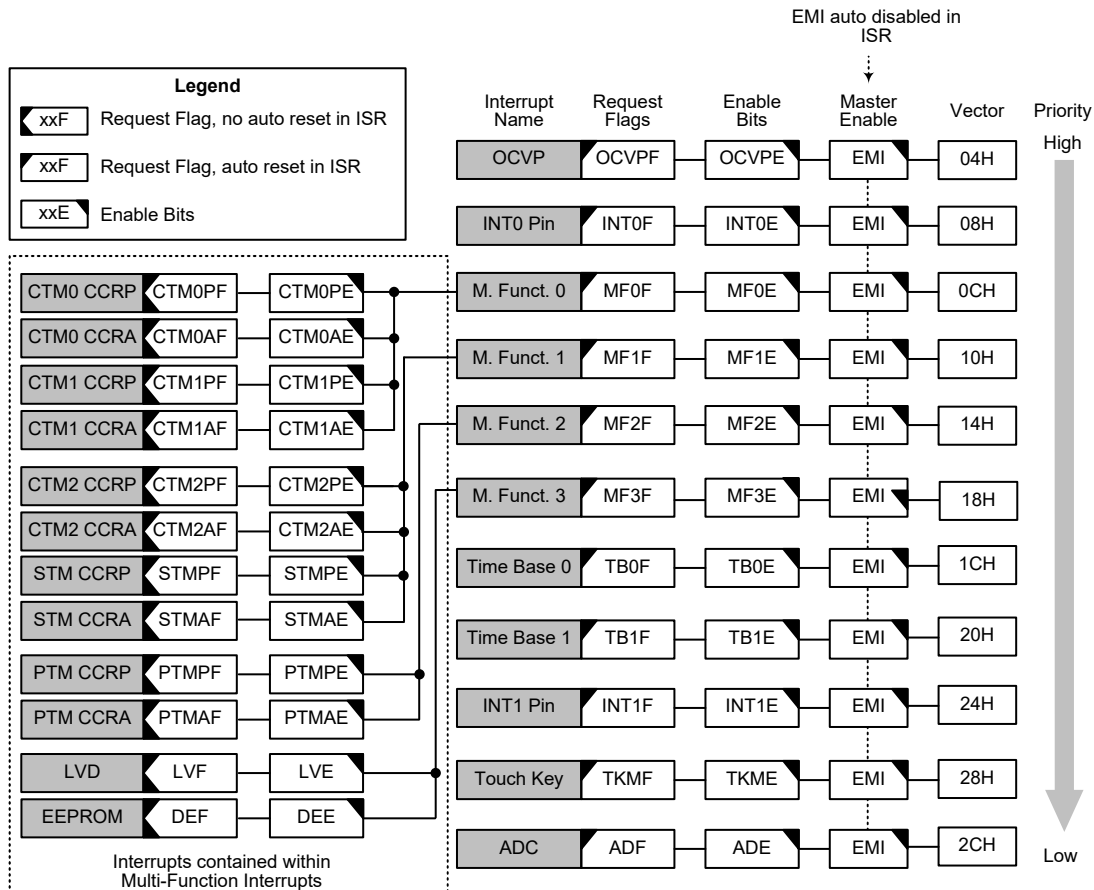
中断操作

若中断事件条件产生，如一个 TM 比较器 P、比较器 A 匹配等等，相关中断请求标志将置起。中断标志产生后程序是否会跳转至相关中断向量执行是由中断使能位的条件决定的。若使能位为“1”，程序将跳至相关中断向量中执行；若使能位为“0”，即使中断请求标志置起中断也不会发生，程序也不会跳转至相关中断向量执行。若总中断使能位为“0”，所有中断都将除能。

当中断发生时，PC 中下条指令的地址将被压入堆栈。相应的中断向量地址加载至 PC。系统将从此向量取得下条指令。中断向量处通常为“JMP”指令，以跳转到相应的中断服务程序，此处程序存放着相应中断的代码。中断服务程序必须以“RETI”指令返回至主程序，以继续执行原来的程序。

各个中断使能位以及相应的请求标志位，以优先级的次序显示在下图。一些中断源有自己的向量，但是有些中断却共用多功能中断向量。一旦中断子程序被响应，系统将自动清除 EMI 位，所有其它的中断将被屏蔽，这个方式可以防止任何进一步的中断嵌套。其它中断请求可能发生在此期间，虽然中断不会立即响应，但是中断请求标志位会被记录。

如果某个中断服务子程序正在执行时，有另一个中断要求立即响应，那么 EMI 位应在程序进入中断子程序后置位，以允许此中断嵌套。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到 SP 减少为止。如果要求立刻动作，则堆栈必须避免成为储满状态。请求同时发生时，执行优先级如下流程图所示。所有被置起的中断请求标志都可把单片机从休眠或空闲模式中唤醒，若要防止唤醒动作发生，在单片机进入休眠或空闲模式前应相应的标志置起。



外部中断

通过 INT0~INT1 引脚上的信号变化可控制外部中断。当触发沿选择位设置好触发类型，INT0~INT1 引脚的状态发生变化，外部中断请求标志 INT0F~INT1F 被置位时外部中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和相应外部中断使能位 INT0E~INT1E 需先被置位。此外，必须使用 INTEG 寄存器使能外部中断功能并选择触发沿类型。外部中断引脚和普通 I/O 口共用，如果相应寄存器中的中断使能位被置位，并且通过引脚共用寄存器选择外部中断脚，此引脚将被作为外部中断脚使用。此时该引脚必须通过设置端口控制寄存器和相关引脚共用选择位，将该引脚设置为输入口。当中断使能，堆栈未满并且外部中断脚状态改变，将调用外部中断向量子程序。当响应外部中断服务子程序时，中断请求标志位 INT0F~INT1F 会自动复位且 EMI 位会被清零以除能其它中断。注意，即使此引脚被用作外部中断输入，其上拉电阻仍保持有效。

寄存器 INTEG 被用于选择有效的边沿类型，来触发外部中断。可以选择上升沿还是下降沿或双沿触发都产生外部中断。注意寄存器 INTEG 也可以用来除能外部中断功能。

OCVP 中断

当 OCVP 电路检测到过电流或过电压情况时，其中断请求标志位 OCVPF 置位，OCVP 中断请求产生。若要跳转到相应中断向量地址，总中断控制位 EMI 和 OCVP 中断使能位 OCVPE 需先被设置。当中断使能，堆栈未满且发生过电压或过电流情况时，将调用 OCVP 中断向量子程序。当响应 OCVP 中断服务子程序时，EMI 将自动清除以除能其它中断，OCVP 中断请求标志位也将自动清除。

A/D 转换器中断

A/D 转换动作的结束控制 A/D 转换器中断。当 A/D 转换器中断请求标志 ADF 被置位，即 A/D 转换过程完成时，中断请求发生。若要跳转到相应中断向量地址，总中断控制位 EMI、A/D 转换器中断使能位 ADE 需先被置位。当中断使能，堆栈未满且 A/D 转换动作结束时，将调用 A/D 转换器中断向量子程序。当 A/D 转换器中断响应时，ADF 标志将自动清除，EMI 将被自动清零以除能其它中断。

多功能中断

此系列单片机中有多达 4 种多功能中断，与其它中断不同，这些没有独立源，但由其它现有的中断源构成，即 TM 中断、LVD 中断和 EEPROM 中断。

当多功能中断中任何一种中断请求标志 MFnF 被置位，多功能中断请求产生。当所包含的任一功能产生中断请求标志，多功能中断标志将置位。若要跳转到相应的中断向量地址，当多功能中断使能，堆栈未满，包括在多功能中断中的任意一个中断发生时，将调用多功能中断向量中的一个子程序。当响应中断服务子程序时，相关的多功能请求标志位会自动复位且 EMI 位会自动清零以除能其它中断。

但必须注意的是，在中断响应时，虽然多功能中断标志会自动复位，但多功能中断源的请求标志位，即 TM 中断、LVD 中断和 EEPROM 中断的请求标志位不会自动复位，必须由应用程序清零。

时基中断

时基中断提供一个固定周期的中断信号，由各自的定时器功能产生溢出信号控制。当出现这种情况时其各自的中断请求标志 TB0F 或 TB1F 被置位，中断请求发生。若要跳转到其相应的中断向量地址，总中断使能位 EMI 和时基使能位 TB0E 或 TB1E 需先被置位。当中断使能，堆栈未满且时基溢出时，将调用它

们各自的中断向量子程序。当响应中断服务子程序时，相应的中断请求标志位 TB0F 或 TB1F 会自动复位且 EMI 位会被清零以除能其它中断。

时基中断的目的是提供一个固定周期的中断信号。其时钟源来自内部时钟源 f_{TB} 。 f_{TB} 输入时钟首先经过分频器，分频率由程序设置 TBC 寄存器相关位获取合适的分频值以提供更长的时基中断周期。相应的控制时基中断周期的时钟源可通过 TBC 寄存器的 TBCK 位选择。

• TBC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TBON	TBCK	TB11	TB10	—	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	1	1	—	1	1	1

Bit 7 **TBON**: TB0 和 TB1 控制位

0: 除能
1: 使能

Bit 6 **TBCK**: f_{TB} 时钟源选择位

0: f_{TBC}
1: $f_{SYS}/4$

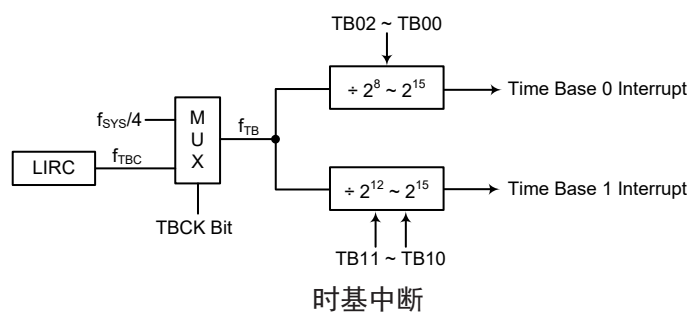
Bit 5~4 **TB11~TB10**: 时基 1 溢出周期选择位

00: $2^{12}/f_{TB}$
01: $2^{13}/f_{TB}$
10: $2^{14}/f_{TB}$
11: $2^{15}/f_{TB}$

Bit 3 未定义，读为“0”

Bit 2~0 **TB02~TB00**: 时基 0 溢出周期选择位

000: $2^8/f_{TB}$
001: $2^9/f_{TB}$
010: $2^{10}/f_{TB}$
011: $2^{11}/f_{TB}$
100: $2^{12}/f_{TB}$
101: $2^{13}/f_{TB}$
110: $2^{14}/f_{TB}$
111: $2^{15}/f_{TB}$



EEPROM 中断

EEPROM 中断属于多功能中断。当 EEPROM 写周期结束，EEPROM 中断请求标志位 DEF 将置位，EEPROM 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI，EEPROM 中断使能位 DEE 以及相关的多功能中断使能位 MF3E 需先被置位。当中断使能，堆栈未满足且一个 EEPROM 写周期结束时，将调用相应的 EEPROM 中断向量子程序。当相应 EEPROM 中断服务时，

EMI 位将自动清零以除能其它中断，但只有多功能中断请求标志位将自动清除。由于 DEF 标志位无法自动清除，必须在应用程序中手动清除。

LVD 中断

LVD 中断属于多功能中断。当低电压检测功能检测到一个低电压时，LVD 中断请求标志 LVF 被置位，LVD 中断请求产生。若要程序跳转到相应中断向量地址，总中断控制位 EMI、低电压中断使能位 LVE 和相应多功能中断使能位 MF3E 需先被置位。当中断使能，堆栈未满且低电压条件发生时，可跳转至相关多功能中断向量子程序中执行。当 LVD 中断响应，EMI 将被自动清零以除能其它中断，多功能中断请求标志也可自动清除。由于 LVD 中断请求标志位 LVF 无法自动清除，必须在应用程序中手动清除。

触控按键模块中断

触控按键中断由时隙计数器溢出来控制。当触控按键模块中断请求标志位 TKMF 被置位，即触控按键中的时隙计数器溢出，触控按键模块中断请求发生。若要跳转到相应中断向量地址，总中断使能位 EMI 和触控按键模块使能位 TKME 必须先被置位，中断使能，堆栈未满且触控按键时隙计数器溢出时，将调用位于触控按键中断向量处的子程序。当响应中断服务子程序时，中断请求标志位 TKMF 会被自动复位且 EMI 位会被清零以除能其它中断。

TM 中断

简易型、标准型和周期型 TM 各有两个中断，都属于多功能中断。不同类型的 TM 都有两个中断请求标志位 xTMnPF 和 xTMnAF 及两个使能位 xTMnPE 和 xTMnAE。当 TM 比较器 P、A 匹配情况发生时，相应 TM 中断请求标志被置位，TM 中断请求产生。

若要程序跳转到相应中断向量地址，总中断控制位 EMI、相应 TM 中断使能位和相关多功能中断使能位 MFnE 需先被置位。当中断使能，堆栈未满且 TM 比较器匹配情况发生时，可跳转至相关多功能中断向量子程序中执行。当 TM 中断响应，EMI 将被自动清零以除能其它中断，相关 MFnF 标志也可自动清除，由于 TM 中断请求标志位无法自动清除，必须在应用程序中手动清除。

中断唤醒功能

每个中断都具有将处于休眠或空闲模式的单片机唤醒的能力。当中断请求标志由低到高转换时唤醒动作产生，其与中断是否使能无关。因此，尽管单片机处于休眠或空闲模式且系统振荡器停止工作，如有外部中断脚上产生外部边沿跳变、低电压都可能导致其相应的中断标志被置位，由此产生中断，因此必须注意避免伪唤醒情况的发生。若中断唤醒功能被除能，单片机进入休眠或空闲模式前相应中断请求标志应被置起。中断唤醒功能不受中断使能位的影响。

编程注意事项

通过禁止相关中断使能位，可以屏蔽中断请求，然而，一旦中断请求标志位被设定，它们会被保留在中断控制寄存器内，直到相应的中断服务子程序执行或请求标志位被软件指令清除。

多功能中断中所含中断相应程序执行时，多功能中断请求标志 MF0F~MF3F 可以自动清零，但各自的请求标志需在应用程序中手动清除。

建议在中断服务子程序中不要使用“CALL 子程序”指令。中断通常发生在不可预料的情况或是需要立刻执行的某些应用。假如只剩下一层堆栈且没有控制好中断，当“CALL 子程序”在中断服务子程序中执行时，将破坏原来的控制

序列。

所有中断在休眠或空闲模式下都具有唤醒功能，当中断请求标志发生由低到高的转变时都可产生唤醒功能。若要避免相应中断产生唤醒动作，在单片机进入休眠或空闲模式前需先将相应请求标志置为高。

当进入中断服务程序，系统仅将程序计数器的内容压入堆栈，如果中断服务程序会改变状态寄存器或其它的寄存器的内容而破坏控制流程，应事先将这些数据保存起来。

若从中断子程序中返回可执行 RET 或 RETI 指令。除了能返回至主程序外，RETI 指令还能自动设置 EMI 位为高，允许进一步中断。RET 指令只能返回至主程序，清除 EMI 位，除能进一步中断。

配置选项

配置选项在烧写程序时写入芯片。通过 HT-IDE 的软件开发环境，使用者在开发过程中可以选择配置选项。由于使用的是硬件工具将配置选项烧入单片机，之后无法再通过应用程序修改。所有的选项必须按系统的需要定义，具体内容可参考下表：

序号	选项
振荡器选择	
1	HIRC 频率选择： 1. 4MHz 2. 8MHz 3. 12MHz

[illegible]

指令集

简介

任何单片机成功运作的核心在于它的指令集，此指令集为一组程序指令码，用来指导单片机如何去执行指定的工作。在 Holtek 单片机中，提供了丰富且灵活的指令，共超过六十条，程序设计者可以事半功倍地实现它们的应用。

为了更加容易理解各种各样的指令码，接下来按功能分组介绍它们。

指令周期

大部分的操作均只需要一个指令周期来执行。分支、调用或查表则需要两个指令周期。一个指令周期相当于四个系统时钟周期，因此如果在 8MHz 的系统时钟振荡器下，大部分的操作将在 0.5 μ s 中执行完成，而分支或调用操作则将在 1 μ s 中执行完成。虽然需要两个指令周期的指令通常指的是 JMP、CALL、RET、RETI 和查表指令，但如果牵涉到程序计数器低字节寄存器 PCL 也将多花费一个周期去加以执行。即指令改变 PCL 的内容进而导致直接跳转至新地址时，需要多一个周期去执行，例如“CLR PCL”或“MOV PCL, A”指令。对于跳转指令必须注意的是，如果比较的结果牵涉到跳转动作将多花费一个周期，如果没有则需一个周期即可。

数据的传送

单片机程序中数据传送是使用最为频繁的操作之一，使用三种 MOV 的指令，数据不但可以从寄存器转移至累加器（反之亦然），而且能够直接移动立即数到累加器。数据传送最重要的应用之一是从输入端口接收数据或传送数据到输出端口。

算术运算

算术运算和数据处理是大部分单片机应用所必需具备的能力，在 Holtek 单片机内部的指令集中，可直接实现加与减的运算。当加法的结果超出 255 或减法的结果少于 0 时，要注意正确的处理进位和借位的问题。INC、INCA、DEC 和 DECA 指令提供了对一个指定地址的值加一或减一的功能。

逻辑和移位运算

标准逻辑运算例如 AND、OR、XOR 和 CPL 全都包含在 Holtek 单片机内部的指令集中。大多数牵涉到数据运算的指令，数据的传送必须通过累加器。在所有逻辑数据运算中，如果运算结果为零，则零标志位将被置位，另外逻辑数据运用形式还有移位指令，例如 RR、RL、RRC 和 RLC 提供了向左或向右移动一位的方法。不同的移位指令可满足不同的应用需要。移位指令常用于串行端口的程序应用，数据可从内部寄存器转移至进位标志位，而此位则可被检验，移位运算还可应用在乘法与除法的运算组成中。

分支和控制转换

程序分支是采取使用 JMP 指令跳转至指定地址或使用 CALL 指令调用子程序的形式，两者之不同在于当子程序被执行完毕后，程序必须马上返回原来的地址。这个动作是由放置在子程序里的返回指令 RET 来实现，它可使程序跳回 CALL 指令之后的地址。在 JMP 指令中，程序则只是跳到一个指定的地址而已，并不需如 CALL 指令般跳回。一个非常有用的分支指令是条件跳转，跳转条件是由数据存储器或定位来加以决定。遵循跳转条件，程序将继续执行下一条指令或略过且跳转至接下来的指令。这些分支指令是程序走向的关键，跳转条件可能是外部开关输入，或是内部数据位的值。

位运算

提供数据存储器中单个位的运算指令是 Holtek 单片机的特性之一。这特性对于输出端口位的设置尤其有用，其中个别的位或端口的引脚可以使用“SET [m].i”或“CLR [m].i”指令来设定其为高位或低位。如果没有这特性，程序设计师必须先读入输入口的 8 位数据，处理这些数据，然后再输出正确的新数据。这种读入 - 修改 - 写出的过程现在则被位运算指令所取代。

查表运算

数据的储存通常由寄存器完成，然而当处理大量固定的数据时，它的存储量常常造成对个别存储器的不便。为了改善此问题，Holtek 单片机允许在程序存储器中建立一个表格作为数据可直接存储的区域，只需要一组简易的指令即可对数据进行查表。

其它运算

除了上述功能指令外，其它指令还包括用于省电的“HALT”指令和使程序在极端电压或电磁环境下仍能正常工作的看门狗定时器控制指令。这些指令的使用则请查阅相关的章节。

指令集概要

下表中说明了按功能分类的指令集，用户可以将该表作为基本的指令参考。

惯例

x: 立即数
m: 数据存储器地址
A: 累加器
i: 第 0~7 位
addr: 程序存储器地址

助记符	说明	指令周期	影响标志位
算术运算			
ADD A,[m]	ACC 与数据存储器相加，结果放入 ACC	1	Z, C, AC, OV
ADDM A,[m]	ACC 与数据存储器相加，结果放入数据存储器	1 注	Z, C, AC, OV
ADD A, x	ACC 与立即数相加，结果放入 ACC	1	Z, C, AC, OV
ADC A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	1	Z, C, AC, OV
ADCM A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	1 注	Z, C, AC, OV
SUB A, x	ACC 与立即数相减，结果放入 ACC	1	Z, C, AC, OV
SUB A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z, C, AC, OV
SUBM A,[m]	ACC 与数据存储器相减，结果放入数据存储器	1 注	Z, C, AC, OV
SBC A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z, C, AC, OV
SBCM A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	1 注	Z, C, AC, OV
DAA [m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	1 注	C
逻辑运算			
AND A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	1	Z
OR A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	1	Z
XOR A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	1	Z
ANDM A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	1 注	Z
ORM A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	1 注	Z
XORM A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	1 注	Z
AND A, x	ACC 与立即数做“与”运算，结果放入 ACC	1	Z
OR A, x	ACC 与立即数做“或”运算，结果放入 ACC	1	Z
XOR A, x	ACC 与立即数做“异或”运算，结果放入 ACC	1	Z
CPL [m]	对数据存储器取反，结果放入数据存储器	1 注	Z
CPLA [m]	对数据存储器取反，结果放入 ACC	1	Z
递增和递减			
INCA [m]	递增数据存储器，结果放入 ACC	1	Z
INC [m]	递增数据存储器，结果放入数据存储器	1 注	Z
DECA [m]	递减数据存储器，结果放入 ACC	1	Z
DEC [m]	递减数据存储器，结果放入数据存储器	1 注	Z
移位			
RRA [m]	数据存储器右移一位，结果放入 ACC	1	无
RR [m]	数据存储器右移一位，结果放入数据存储器	1 注	无
RRCA [m]	带进位将数据存储器右移一位，结果放入 ACC	1	C
RRC [m]	带进位将数据存储器右移一位，结果放入数据存储器	1 注	C
RLA [m]	数据存储器左移一位，结果放入 ACC	1	无

助记符	说明	指令周期	影响标志位
RL [m]	数据存储器左移一位，结果放入数据存储器	1 注	无
RLCA [m]	带进位将数据存储器左移一位，结果放入 ACC	1	C
RLC [m]	带进位将数据存储器左移一位，结果放入数据存储器	1 注	C
数据传送			
MOV A,[m]	将数据存储器送至 ACC	1	无
MOV [m],A	将 ACC 送至数据存储器	1 注	无
MOV A, x	将立即数送至 ACC	1	无
位运算			
CLR [m].i	清除数据存储器的位	1 注	无
SET [m].i	置位数据存储器的位	1 注	无
转移			
JMP addr	无条件跳转	2	无
SZ [m]	如果数据存储器为零，则跳过下一条指令	1 注	无
SZA [m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	1 注	无
SZ [m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	1 注	无
SNZ [m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	1 注	无
SIZ [m]	递增数据存储器，如果结果为零，则跳过下一条指令	1 注	无
SDZ [m]	递减数据存储器，如果结果为零，则跳过下一条指令	1 注	无
SIZA [m]	递增数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1 注	无
SDZA [m]	递减数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1 注	无
CALL addr	子程序调用	2	无
RET	从子程序返回	2	无
RET A, x	从子程序返回，并将立即数放入 ACC	2	无
RETI	从中断返回	2	无
查表			
TABRD [m]	读取特定页或当前页的 ROM 内容，并送至数据存储器 and TBLH	2 注	无
TABRDL [m]	读取最后页的 ROM 内容，并送至数据存储器 and TBLH	2 注	无
其它指令			
NOP	空指令	1	无
CLR [m]	清除数据存储器	1 注	无
SET [m]	置位数据存储器	1 注	无
CLR WDT	清除看门狗定时器	1	TO, PDF
SWAP [m]	交换数据存储器的高低字节，结果放入数据存储器	1 注	无
SWAPA [m]	交换数据存储器的高低字节，结果放入 ACC	1	无
HALT	进入暂停模式	1	TO, PDF

- 注: 1. 对跳转指令而言，如果比较的结果牵涉到跳转即需 2 个周期，如果没有发生跳转，则只需一个周期。
2. 任何指令若要改变 PCL 的内容将需要 2 个周期来执行。

指令定义

ADC A, [m]	Add Data Memory to ACC with Carry
指令说明	将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C
ADCM A, [m]	Add ACC to Data Memory with Carry
指令说明	将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。
功能表示	$[m] \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C
ADD A, [m]	Add Data Memory to ACC
指令说明	将指定的数据存储器和累加器内容相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C
ADD A, x	Add immediate data to ACC
指令说明	将累加器和立即数相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + x$
影响标志位	OV、Z、AC、C
ADDM A, [m]	Add ACC to Data Memory
指令说明	将指定的数据存储器和累加器内容相加，结果存放到指定的数据存储器。
功能表示	$[m] \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C
AND A, [m]	Logical AND Data Memory to ACC
指令说明	将累加器中的数据和指定数据存储器内容做逻辑与，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "AND" } [m]$
影响标志位	Z

AND A, x	Logical AND immediate data to ACC
指令说明	将累加器中的数据和立即数做逻辑与，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ “AND” } x$
影响标志位	Z
ANDM A, [m]	Logical AND ACC to Data Memory
指令说明	将指定数据存储器内容和累加器中的数据做逻辑与，结果存放到数据存储器。
功能表示	$[m] \leftarrow ACC \text{ “AND” } [m]$
影响标志位	Z
CALL addr	Subroutine call
指令说明	无条件地调用指定地址的子程序，此时程序计数器先加 1 获得下一个要执行的指令地址并压入堆栈，接着载入指定地址并从新地址继续执行程序，由于此指令需要额外的运算，所以为一个 2 周期的指令。
功能表示	$Stack \leftarrow Program\ Counter + 1$ $Program\ Counter \leftarrow addr$
影响标志位	无
CLR [m]	Clear Data Memory
指令说明	将指定数据存储器的内容清零。
功能表示	$[m] \leftarrow 00H$
影响标志位	无
CLR [m].i	Clear bit of Data Memory
指令说明	将指定数据存储器的第 i 位内容清零。
功能表示	$[m].i \leftarrow 0$
影响标志位	无
CLR WDT	Clear Watchdog Timer
指令说明	WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。
功能表示	WDT cleared $TO \ \& \ PDF \leftarrow 0$
影响标志位	TO、PDF

CPL [m]	Complement Data Memory
指令说明	将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1。
功能表示	$[m] \leftarrow \overline{[m]}$
影响标志位	Z
CPLA [m]	Complement Data Memory with result in ACC
指令说明	将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或 0 变 1，而结果被储存回累加器且数据存储器中的内容不变。
功能表示	$ACC \leftarrow \overline{[m]}$
影响标志位	Z
DAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
指令说明	将累加器中的内容转换为 BCD (二进制转成十进制) 码。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对原值加“6”，否则原值保持不变；如果高四位的值大于“9”或 C=1，那么 BCD 调整就执行对原值加“6”。
	BCD 转换实质上是根据累加器和标志位执行 00H, 06H, 60H 或 66H 的加法运算，结果存放到数据存储器。只有进位标志位 C 受影响，用来指示原始 BCD 的和是否大于 100，并可以进行双精度十进制数的加法运算。
功能表示	$[m] \leftarrow ACC + 00H$ 或 $[m] \leftarrow ACC + 06H$ 或 $[m] \leftarrow ACC + 60H$ 或 $[m] \leftarrow ACC + 66H$
影响标志位	C
DEC [m]	Decrement Data Memory
指令说明	将指定数据存储器内容减 1。
功能表示	$[m] \leftarrow [m] - 1$
影响标志位	Z
DECA [m]	Decrement Data Memory with result in ACC
指令说明	将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器的内容不变。
功能表示	$ACC \leftarrow [m] - 1$
影响标志位	Z

HALT	Enter power down mode
指令说明	此指令终止程序执行并关掉系统时钟，RAM 和寄存器的内容保持原状态，WDT 计数器和分频器被清“0”，暂停标志位 PDF 被置位 1，WDT 溢出标志位 TO 被清 0。
功能表示	$TO \leftarrow 0$ $PDF \leftarrow 1$
影响标志位	TO、PDF
INC [m]	Increment Data Memory
指令说明	将指定数据存储器的内容加 1。
功能表示	$[m] \leftarrow [m] + 1$
影响标志位	Z
INCA [m]	Increment Data Memory with result in ACC
指令说明	将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。
功能表示	$ACC \leftarrow [m] + 1$
影响标志位	Z
JMP addr	Jump unconditionally
指令说明	程序计数器的内容无条件地由被指定的地址取代，程序由新的地址继续执行。当新的地址被加载时，必须插入一个空指令周期，所以此指令为 2 个周期的指令。
功能表示	$Program\ Counter \leftarrow addr$
影响标志位	无
MOV A, [m]	Move Data Memory to ACC
指令说明	将指定数据存储器的内容复制到累加器。
功能表示	$ACC \leftarrow [m]$
影响标志位	无
MOV A, x	Move immediate data to ACC
指令说明	将 8 位立即数载入累加器。
功能表示	$ACC \leftarrow x$
影响标志位	无
MOV [m], A	Move ACC to Data Memory
指令说明	将累加器的内容复制到指定的数据存储器。
功能表示	$[m] \leftarrow ACC$
影响标志位	无

NOP	No operation
指令说明	空操作，接下来顺序执行下一条指令。
功能表示	无操作
影响标志位	无
ORA, [m]	Logical OR Data Memory to ACC
指令说明	将累加器中的数据和指定的数据存储器内容逻辑或，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z
ORA, x	Logical OR immediate data to ACC
指令说明	将累加器中的数据和立即数逻辑或，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "OR" } x$
影响标志位	Z
ORM A, [m]	Logical OR ACC to Data Memory
指令说明	将存在指定数据存储器中的数据和累加器逻辑或，结果放到数据存储器。
功能表示	$[m] \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z
RET	Return from subroutine
指令说明	将堆栈寄存器中的程序计数器值恢复，程序由取回的地址继续执行。
功能表示	Program Counter ← Stack
影响标志位	无
RET A, x	Return from subroutine and load immediate data to ACC
指令说明	将堆栈寄存器中的程序计数器值恢复且累加器载入指定的立即数，程序由取回的地址继续执行。
功能表示	Program Counter ← Stack $ACC \leftarrow x$
影响标志位	无

RETI	Return from interrupt
指令说明	将堆栈寄存器中的程序计数器值恢复且中断功能通过设置 EMI 位重新使能。EMI 是控制中断使能的主控制位。如果在执行 RETI 指令之前还有中断未被响应，则这个中断将在返回主程序之前被响应。
功能表示	Program Counter $\leftarrow$ Stack
影响标志位	EMI $\leftarrow$ 1 无
RL [m]	Rotate Data Memory left
指令说明	将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
功能表示	[m].(i+1) $\leftarrow$ [m].i (i=0~6) [m].0 $\leftarrow$ [m].7
影响标志位	无
RLA [m]	Rotate Data Memory left with result in ACC
指令说明	将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，结果送到累加器，而指定数据存储器的内容保持不变。
功能表示	ACC.(i+1) $\leftarrow$ [m].i (i=0~6) ACC.0 $\leftarrow$ [m].7
影响标志位	无
RLC [m]	Rotate Data Memory Left through Carry
指令说明	将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位。
功能表示	[m].(i+1) $\leftarrow$ [m].i (i=0~6) [m].0 $\leftarrow$ C C $\leftarrow$ [m].7
影响标志位	C
RLC A [m]	Rotate Data Memory left through Carry with result in ACC
指令说明	将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。
功能表示	ACC.(i+1) $\leftarrow$ [m].i (i=0~6) ACC.0 $\leftarrow$ C C $\leftarrow$ [m].7
影响标志位	C

RR [m]	Rotate Data Memory right
指令说明	将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow [m].0$
影响标志位	无
RRA [m]	Rotate Data Memory right with result in ACC
指令说明	将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow [m].0$
影响标志位	无
RRC [m]	Rotate Data Memory right through Carry
指令说明	将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
RRCA [m]	Rotate Data Memory right through Carry with result in ACC
指令说明	将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位，移位结果送回累加器，但是指定数据寄存器的内容保持不变。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
SBC A, [m]	Subtract Data Memory from ACC with Carry
指令说明	将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C

SBCM A, [m]	Subtract Data Memory from ACC with Carry and result in Data Memory
指令说明	将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$[m] \leftarrow ACC - [m] - \overline{C}$
影响标志位	OV、Z、AC、C
SDZ [m]	Skip if Decrement Data Memory is 0
指令说明	将指定的数据存储器的内容减 1，判断是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$[m] \leftarrow [m] - 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC
指令说明	将指定数据存储器内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$ACC \leftarrow [m] - 1$ ，如果 $ACC=0$ 跳过下一条指令执行
影响标志位	无
SET [m]	Set Data Memory
指令说明	将指定数据存储器的每一位设置为 1。
功能表示	$[m] \leftarrow FFH$
影响标志位	无
SET [m].i	Set bit of Data Memory
指令说明	将指定数据存储器的第 i 位置位为 1。
功能表示	$[m].i \leftarrow 1$
影响标志位	无

SIZ [m]	Skip if increment Data Memory is 0
指令说明	将指定的数据存储器的内容加 1，判断是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$[m] \leftarrow [m] + 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无
SIZA [m]	Skip if increment Data Memory is zero with result in ACC
指令说明	将指定数据存储器的内容加 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$ACC \leftarrow [m] + 1$ ，如果 $ACC=0$ 跳过下一条指令执行
影响标志位	无
SNZ [m].i	Skip if bit i of Data Memory is not 0
指令说明	判断指定数据存储器的第 i 位，若不为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果为 0，则程序继续执行下一条指令。
功能表示	如果 $[m].i \neq 0$ ，跳过下一条指令执行
影响标志位	无
SUB A, [m]	Subtract Data Memory from ACC
指令说明	将累加器的内容减去指定的数据存储器的数据，把结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C
SUBM A, [m]	Subtract Data Memory from ACC with result in Data Memory
指令说明	将累加器的内容减去指定数据存储器的数据，结果存放到指定的数据存储器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$[m] \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C

SUB A, x	Subtract immediate Data from ACC
指令说明	将累加器的内容减去立即数，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - x$
影响标志位	OV、Z、AC、C
SWAP [m]	Swap nibbles of Data Memory
指令说明	将指定数据存储器的低 4 位和高 4 位互相交换。
功能表示	$[m].3 \sim [m].0 \leftrightarrow [m].7 \sim [m].4$
影响标志位	无
SWAPA [m]	Swap nibbles of Data Memory with result in ACC
指令说明	将指定数据存储器的低 4 位与高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。
功能表示	$ACC.3 \sim ACC.0 \leftarrow [m].7 \sim [m].4$ $ACC.7 \sim ACC.4 \leftarrow [m].3 \sim [m].0$
影响标志位	无
SZ [m]	Skip if Data Memory is 0
指令说明	指定数据存储器的内容会先被读出，后又被重新写入指定数据存储器内。判断指定数据存储器的内容是否为 0，若为 0，则程序跳过下一条指令执行。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	如果 $[m]=0$ ，跳过下一条指令执行
影响标志位	无
SZA [m]	Skip if Data Memory is 0 with data movement to ACC
指令说明	将指定数据存储器内容复制到累加器，并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$ACC \leftarrow [m]$ ，如果 $[m]=0$ ，跳过下一条指令执行
影响标志位	无

SZ [m].i	Skip if bit i of Data Memory is 0
指令说明	判断指定数据存储器的第 i 位是否为 0，若为 0，则跳过下一条指令。由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	如果 [m].i=0，跳过下一条指令执行
影响标志位	无
TABRD [m]	Read table (specific page or current page) to TBLH and Data Memory
指令说明	将表格指针 (TBHP 和 TBLP，若无 TBHP 则仅 TBLP) 所指的程序代码低字节移至指定数据存储器且将高字节移至 TBLH。
功能表示	[m] ← 程序代码 (低字节) TBLH ← 程序代码 (高字节)
影响标志位	无
TABRDL [m]	Read table (last page) to TBLH and Data Memory
指令说明	将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定数据存储器且将高字节移至 TBLH。
功能表示	[m] ← 程序代码 (低字节) TBLH ← 程序代码 (高字节)
影响标志位	无
XOR A, [m]	Logical XOR Data Memory to ACC
指令说明	将累加器的数据和指定的数据存储器内容逻辑异或，结果存放到累加器。
功能表示	ACC ← ACC “XOR” [m]
影响标志位	Z
XORM A, [m]	Logical XOR ACC to Data Memory
指令说明	将累加器的数据和指定的数据存储器内容逻辑异或，结果放到数据存储器。
功能表示	[m] ← ACC “XOR” [m]
影响标志位	Z
XOR A, x	Logical XOR immediate data to ACC
指令说明	将累加器的数据与立即数逻辑异或，结果存放到累加器。
功能表示	ACC ← ACC “XOR” x
影响标志位	Z

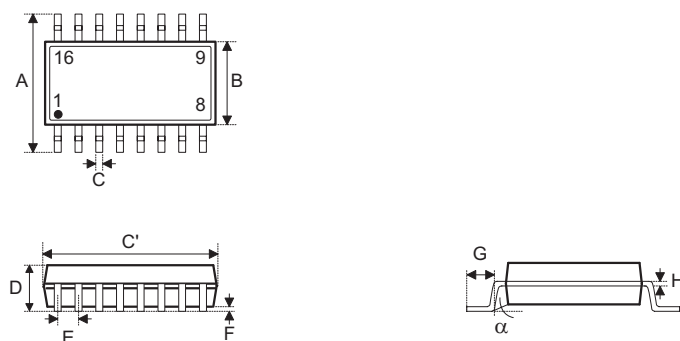
封装信息

请注意，这里提供的封装信息仅作为参考。由于这个信息经常更新，提醒用户咨询 [Holtek 网站](#) 以获取最新版本的[封装信息](#)。

封装信息的相关内容如下所示，点击可链接至 Holtek 网站相关信息页面。

- 封装信息 (包括外形尺寸、包装带和卷轴规格)
- 封装材料信息
- 纸箱信息

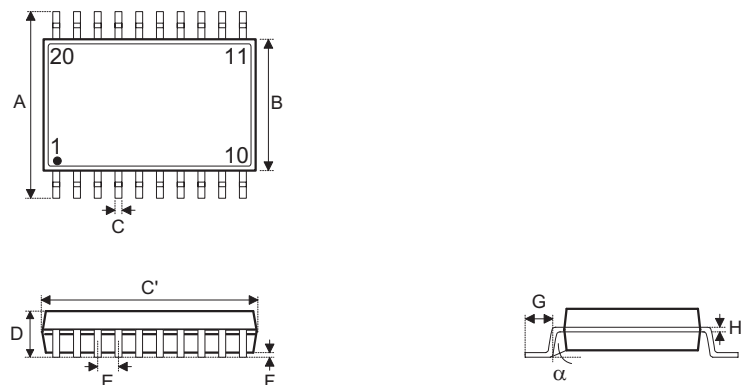
16-pin NSOP (150mil) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.390 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符号	尺寸 (单位: mm)		
	最小值	典型值	最大值
A	—	6.00 BSC	—
B	—	3.90 BSC	—
C	0.31	—	0.51
C'	—	9.90 BSC	—
D	—	—	1.75
E	—	1.27 BSC	—
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°

20-pin NSOP (150mil) 外形尺寸



符号	尺寸 (单位: inch)		
	最小值	典型值	最大值
A	0.228	0.236	0.244
B	0.146	0.154	0.161
C	0.009	—	0.012
C'	0.382	0.390	0.398
D	—	—	0.069
E	—	0.032 BSC	—
F	0.002	—	0.009
G	0.020	—	0.031
H	0.008	—	0.010
α	0°	—	8°

符号	尺寸 (单位: mm)		
	最小值	典型值	最大值
A	5.80	6.00	6.20
B	3.70	3.90	4.10
C	0.23	—	0.30
C'	9.70	9.90	10.10
D	—	—	1.75
E	—	0.80 BSC	—
F	0.05	—	0.23
G	0.50	—	0.80
H	0.21	—	0.25
α	0°	—	8°

Copyright© 2022 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

本文件出版时 HOLTEK 已针对所载信息为合理注意，但不保证信息准确无误。文中提到的信息仅是提供作为参考，且可能被更新取代。HOLTEK 不担保任何明示、默示或法定的，包括但不限于适合商品化、令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。HOLTEK 就文中提到的信息及该信息之应用，不承担任何法律责任。此外，HOLTEK 并不推荐将 HOLTEK 的产品使用在会由于故障或其他原因而可能会对人身安全造成危害的地方。HOLTEK 特此声明，不授权将产品使用于救生、维生或安全关键零部件。在救生 / 维生或安全应用中使用 HOLTEK 产品的风险完全由买方承担，如因该等使用导致 HOLTEK 遭受损害、索赔、诉讼或产生费用，买方同意出面进行辩护、赔偿并使 HOLTEK 免受损害。HOLTEK (及其授权方，如适用) 拥有本文件所提供信息 (包括但不限于内容、数据、示例、材料、图形、商标) 的知识产权，且该信息受著作权法和其他知识产权法的保护。HOLTEK 在此并未明示或暗示授予任何知识产权。HOLTEK 拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息，请与我们联系。