

深圳宇凡微电子有限公司
8-Bit Single-Chip Microcontrollers

YF2712J

用户手册 (Ver 1.1)

2019 年 6 月修订



目 录

1 芯片简介	6
1.1 功能特性.....	6
1.2 引脚分配.....	6
1.3 引脚说明.....	7
1.4 系统框图.....	8
2 存储器结构	9
2.1 程序存储器区.....	9
2.2 数据存储器区.....	9
2.2.1 RAM 数据寄存器区.....	10
2.2.2 系统寄存器列表.....	10
2.2.3 系统寄存器说明.....	10
2.2.4 系统寄存器位定义.....	10
3 功能模块	12
3.1 系统寄存器.....	12
3.1.1 082H-R (工作寄存器和 ROM 查表数据缓存器)	12
3.1.2 083H-Z (专用寄存器)	12
3.1.3 084H-Y (专用寄存器)	12
3.1.4 086H-PFLAG (ROM 页和特殊标志寄存器)	12
3.1.5 087H-IRCCAL (IRC 频率修调寄存器)	13
3.1.6 0AEH-P4CON (P4 配置控制寄存器)	13
3.1.7 0AFH-VREF (ADC 参考电压寄存器)	13
3.1.8 0B1H-ADM (ADC 模式寄存器)	14
3.1.9 0B2H-ADB (ADC 数据缓存器)	14
3.1.10 0B3H-ADR (ADC 精度选择寄存器)	14
3.1.11 0B4H-ADT (ADC 零漂寄存器)	15
3.1.12 0B8H-P0M (P0M 方向控制寄存器)	15
3.1.13 0BFH-PEDGE (P00 模式控制寄存器)	15
3.1.14 0C4H-P4M (P4M 模式控制寄存器)	15
3.1.15 0C5H-P5M (P5M 模式控制寄存器)	15
3.1.16 0C8H-INTRQ (中断请求寄存器)	15
3.1.17 0C9H-INTEN (中断使能寄存器)	16
3.1.18 0CAH-OSCM (振动模式寄存器)	16
3.1.19 0CCH-WDTR (看门狗清零寄存器)	16
3.1.20 0CDH-TC0R (TC0 自动装载数据缓存器)	16
3.1.21 0CEH-PCL (程序计数器低位)	17
3.1.22 0CFH-PCH (程序计数器高位)	17
3.1.23 0D0H-P0 (P0 数据寄存器)	17
3.1.24 0D4H-P4 (P4 数据寄存器)	17
3.1.25 0D5H-P5 (P5 数据寄存器)	17
3.1.26 0D8H-T0M (TC 加速和 TC0 唤醒功能寄存器)	17
3.1.27 0DAH-TC0M (TC0 模式寄存器)	18
3.1.28 0DBH-TC0C (TC0 计数寄存器)	18
3.1.29 0DCH-TC1M (TC1 模式寄存器)	19

3.1.30 ODDH-TC1C (TC1 计数寄存器)	20
3.1.31 ODEH-TC1R (TC1 自动装载数据缓存器)	20
3.1.32 ODFH-STKP (堆栈指针)	21
3.1.33 OE0H-P0UR (P0 上拉电阻控制寄存器)	21
3.1.34 OE4H-P4UR (P4 上拉电阻控制寄存器)	21
3.1.35 OE5H-P5UR (P5 上拉电阻控制寄存器)	21
3.1.36 OE7H-@YZ (间接寻址寄存器)	21
3.1.37 OF8H~OFFH-STK0~STK3 (堆栈寄存器)	21
3.2 GPIO 功能模块	22
3.2.1 GPIO 寄存器说明	22
0B8H (P0M 方向控制寄存器)	22
0C4H (P4M 方向控制寄存器)	22
0C5H (P5M 方向控制寄存器)	22
0D0H (P0 数据寄存器)	22
0D4H (P4 数据寄存器)	22
0D5H (P5 数据寄存器)	23
0E0H (P0UR 上拉控制寄存器)	23
0E4H (P4UR 上拉控制寄存器)	23
0E5H (P5UR 上拉控制寄存器)	23
3.3 TCC 定时器功能模块	24
3.3.1 看门狗定时器	24
0CCH (WDTR 数据寄存器)	24
3.3.2 定时/计数器 TC0	25
3.3.2.1 概述	25
3.3.2.2 TC0 操作	25
3.3.2.3 TC0M 模式寄存器	26
3.3.2.4 TC0X8/TC0GN 标志	27
3.3.2.5 TC0C 计数寄存器	27
3.3.2.6 TC0R 自动装载寄存器	28
3.3.2.7 TC0 事件计数器	29
3.3.2.8 TC0 时钟频率输出	29
3.3.2.9 脉冲宽度调制 PWM	30
3.3.3 定时/计数器 TC1	31
3.3.3.1 概述	31
3.3.3.2 TC1 操作	31
3.3.3.3 TC1M 模式寄存器	32
3.3.3.4 TC1X8/TC1GN 标志	33
3.3.3.5 TC1C 计数寄存器	33
3.3.3.6 TC1R 自动装载寄存器	34
3.3.3.7 TC1 事件计数器	35
3.3.3.8 TC1 时钟频率输出	35
3.3.3.9 脉冲宽度调制 PWM	35
3.4 ADC 数模转换功能模块	37
3.4.1 概述	37
3.4.2 ADC 模式数据寄存器	37

0B1H (ADM 控制寄存器)	37
0B2H (ADB 数据寄存器)	38
0B3H (ADR 控制寄存器)	38
3.4.3 ADC 引脚配置寄存器	38
0AEH (P4CON 端口配置控制寄存器)	38
3.4.4 ADC 参考电压寄存器	39
0AFH (VREFH 基准电压控制寄存器)	39
3.4.5 ADC 数模转换应用电路说明	39
3.4.6 ADC 模数转换设置说明	40
3.5 LVD 电压检测功能模块	41
3.5.1 LVD 电压检测寄存器说明	41
086H (PFLAG 控制寄存器)	41
3.6 中断功能模块	42
3.6.1 中断请求使能寄存器	42
0C9H (INTEN 控制寄存器)	42
3.6.2 中断请求寄存器	43
0C8H (INTRQ 控制寄存器)	43
3.6.3 GIE 全局中断	43
0DFH (STKP 控制寄存器)	43
3.6.4 INT0 (P0.0) 中断	43
0BFH (PEDGE 控制寄存器)	44
3.6.5 INT1 (P0.1) 中断	44
3.6.6 TCO 中断	44
3.6.7 TC1 中断	44
3.6.8 ADC 中断	45
3.6.9 中断处理说明	45
3.7 复位功能模块	47
3.7.1 复位功能概述	47
0B6H (PFLAG 控制寄存器)	47
3.7.2 上电复位	48
3.7.3 WDT 看门狗复位	48
3.7.4 掉电复位	48
3.7.5 工作频率与 LVR 低压复位关系	49
3.8 系统时钟功能模块	50
3.8.1 外部晶体振荡器/陶瓷谐振器 (XT)	50
3.8.2 内部 RC 振荡器模式 (IRC)	50
3.8.3 OSCM 寄存器	51
3.9 系统工作模式	52
3.9.1 概述	52
3.9.2 普通模式	52
3.9.3 低速模式	53
3.9.4 睡眠模式	53
3.9.5 绿色模式	53
3.9.6 控制宏	54
3.9.7 系统唤醒	54

4 CODE OPTION 寄存器	56
5 芯片电气特性	57
5.1 芯片极限参数.....	57
5.2 芯片直流参数.....	57
6 封装类型	58
7 封装尺寸	59
7.1 16PIN 封装尺寸.....	59
7.3 14PIN 封装尺寸.....	61
7.3 8PIN 封装尺寸.....	63

1 芯片简介

1.1 功能特性

CPU 配置

- 2K×16-Bit OTP ROM
- 128×8-Bit SRAM
- 4 级堆栈空间
- 2 级可编程低电压检测 (LVD)
2.4V/3.6V
- 3 级可编程低电压复位 (LVR)
2.0V/2.4V/3.6V
- 工作电流小于 1mA (16MHz/5V)
- 工作电流 5μA (32kHz/3V)
- 休眠电流小于 1μA (休眠模式)

I/O 配置

- 3 组双向 I/O 端口:P0, P4, P5
- 14 个 I/O 引脚
(P04 仅漏极开路输出)
- 唤醒端口:P00~P06 电平触发
- 13 个可编程上拉 I/O 引脚
(另 P04 上拉可通过 OPTION 配置)
- 外部中断:P0.0, P0.1

工作电压

- 工作电压范围:
1.8V~5.5V (0°C~70°C)
2.3V~5.5V (-40°C~85°C)

系统时钟

- 内置低速振荡电路:
16KHz (3V)
32KHz (5V)
- 内置高速振荡电路:
4MHz, 8MHz, 16MHz, 32MHz
- 外置高速振荡电路:
XT 高达 16MHz
- 指令周期分频选择:

1T/2T/4T/8T/16T

外围模块

- 5 通道 12Bit ADC 模拟转换器
- 1 通道 ADC 内部 VDD 输入检测
- 内置 ADC 基准 (2V, 3V, 4V)
- 8Bit*2 定时器/计数器
- 8Bit*2 PWM/Buzzer 输出

中断源

- 3 个内部中断
TC0/TC1/ADC
- 2 个外部中断
INT0/INT1

工作模式

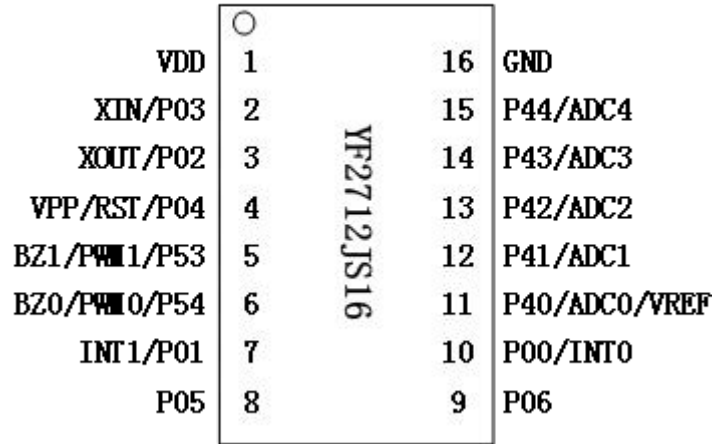
- 普通模式:高/低速时钟同时工作
- 低速模式:只有低速时钟工作
- 睡眠模式:高低速时钟都停止工作
- 绿色模式:由 TC0 周期性唤醒

封装类型

- YF2712JPD (SOP16);
- YF2712JM (SOP14);
- YF2712JD (SOP8);
- YF2712JPB (DIP16);
- YF2712JN (DIP14);
- YF2712JB (DIP8)

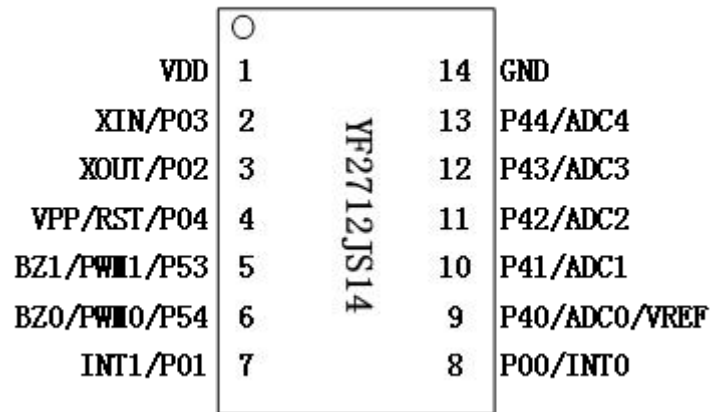
1.2 引脚分配

Top View



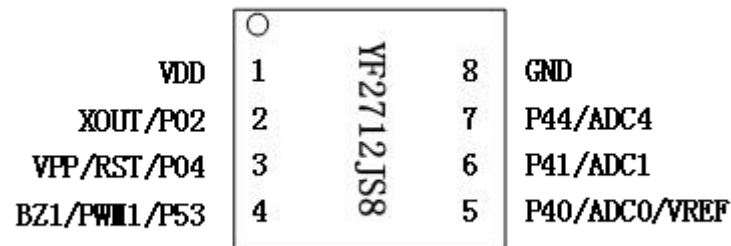
YF2712J-16PIN 脚位图

Top View



YF2712J-14PIN 脚位图

Top View

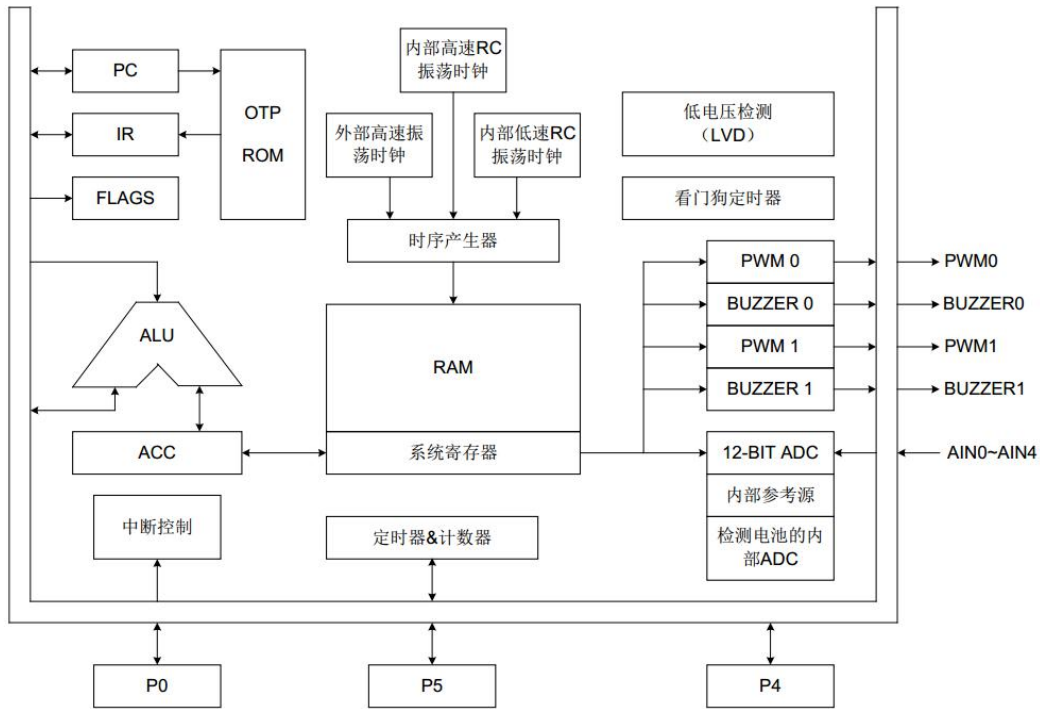


YF2712J-8PIN 脚位图

1.3 引脚说明

序号	管脚名	I/O	功能描述
P03	P03	I/O	GPIO, 可编程上拉、端口唤醒
	XIN	I	外部振荡器输入口
P02	P02	I/O	GPIO, 可编程上拉、端口唤醒
	XOUT	O	外部振荡器输出口
P04	P04	I/O	GPI, 端口唤醒, 可漏极开漏输出, 可 OPTION 配置上拉
	RST	I	外部复位输入口, 低电平触发
	VPP	I	外部编程高压输入口
P53	P53	I/O	GPIO, 可编程上拉
	PWM1	O	PWM1 输出
	BZ1	O	Buzzer1 输出
P54	P54	I/O	GPIO, 可编程上拉
	PWM0	O	PWM0 输出
	BZ0	O	Buzzer0 输出
P01	P01	I/O	GPIO, 可编程上拉、端口唤醒
	INT1	I	外部中断输入口
P05	P05	I/O	GPIO, 可编程上拉
P06	P06	I/O	GPIO, 可编程上拉
P00	P00	I/O	GPIO, 可编程上拉、端口唤醒
	INT0	I	外部中断输入口
P40	P40	I/O	GPIO, 可编程上拉
	ADC0	I	外部 ADC 输入检测口
	VREF	I	外部 ADC 基准电压输入口
P41	P41	I/O	GPIO, 可编程上拉
	ADC1	I	ADC 输入检测口
P42	P42	I/O	GPIO, 可编程上拉
	ADC2	I	ADC 输入检测口
P43	P43	I/O	GPIO, 可编程上拉
	ADC3	I	ADC 输入检测口
P44	P44	I/O	GPIO, 可编程上拉
	ADC4	I	ADC 输入检测口
	VDD	--	电源
	GND	--	地

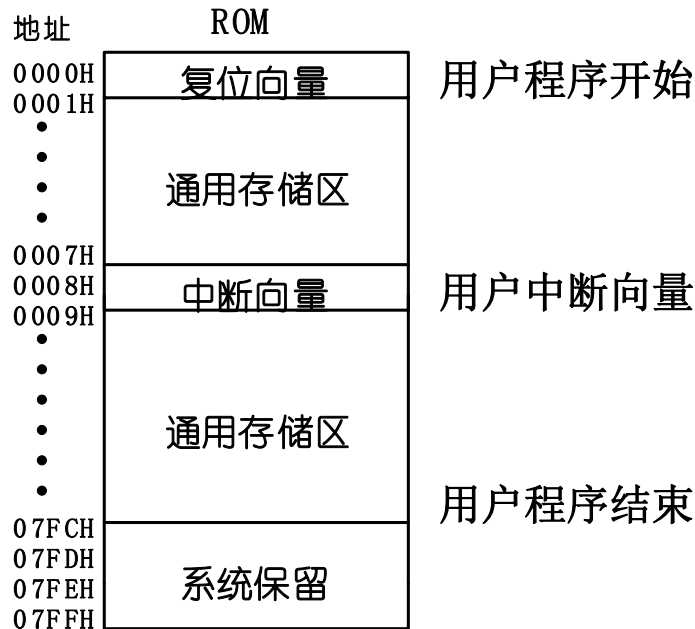
1.4 系统框图



系统结构框图

2 存储器结构

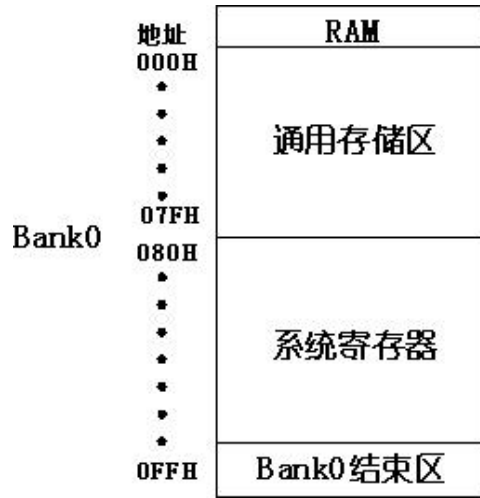
2.1 程序存储器区



程序存储器区结构图

2.2 数据存储器区

2.2.1 RAM 数据寄存器区



2.2.2 系统寄存器列表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8			R	Z	Y		PFLAG	IRC CAL								
9																
A															P4CON	VREF
B		ADM	ADB	ADR	ADT				POM							PEDGE
C					P4M	P5M			INTRQ	INTEN	OSCM		WDTR	TCOR	PCL	PCH
D	P0				P4	P5			TOM		TCOM	TCOC	TC1M	TC1C	TC1R	STKP
E	POUR				P4UR	P5UR		@YZ								
F									STK3L	STK3H	STK2L	STK2H	STK1L	STK1H	STK0L	STK0H

2.2.3 系统寄存器说明

R =工作寄存器和 ROM 查表数据缓存器
PFLAG =ROM 页和特殊标志寄存器
VREF =ADC 参考电压寄存器
ADB =ADC 数据缓存器
PEDGE =P00 模式控制寄存器
INTRQ =中断请求寄存器
OSCM =振动模式寄存器
TCOR =TC0 自动装载数据缓存器
Pn =Pn 数据缓存器
TCOM =TC0 模式寄存器
TC1M =TC1 模式寄存器
TC1R =TC1 自动装载数据缓存器
PnUR =Pn 上拉电阻控制寄存器
STK0~STK3 =堆栈寄存器
IRCCAL=IRC 频率修调寄存器

Y, Z =专用寄存器
@YZ =间接寻址寄存器
P4CON =P4 配置控制寄存器
ADM =ADC 模式寄存器
ADR =ADC 精度选择寄存器
ADT =ADC 零漂寄存器
PnM =Pn 模式控制寄存器
INTEN =中断使能寄存器
WDTR =看门狗清零寄存器
PCH&L =程序计数器
TOM =TC 加速和 TC0 唤醒功能寄存器
TCOC =TC0 计数寄存器
TC1C =TC1 计数寄存器
STKP =堆栈指针

2.2.4 系统寄存器位定义

地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	注释
082H	RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0	R/W	R
083H	ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0	R/W	Z
084H	YBIT7	YBIT6	YBIT5	YBIT4	YBIT3	YBIT2	YBIT1	YBIT0	R/W	Y
086H	NT0	NPD	LVD36	LVD24		C	DC	Z	R/W	PFLAG
0AEH				P4CON4	P4CON3	P4CON2	P4CON1	P4CON0	R/W	P4CON
0AFH	EVHENB						VHS1	VHS0	R/W	VREF
0B1H	ADENB	ADS	EOC	GCHS		CHS2	CHS1	CHS0	R/W	ADM
0B2H	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	R	ADB
0B3H		ADCKS1		ADCKS0	ADB3	ADB2	ADB1	ADB0	R/W	ADR
0B4H	ADTS1	ADTS0		ADT4	ADT3	ADT2	ADT1	ADT0	R/W	ADT
0B8H		P06M	P05M		P03M	P02M	P01M	P00M	R/W	P0M
0BFH				P00G1	P00G0				R/W	PEDGE
0C4H				P44M	P43M	P42M	P41M	P40M	R/W	P4M
0C5H				P54M	P53M					
0C8H	ADCIRQ	TC1IRQ	TC0IRQ				P01IRQ	P00IRQ	R/W	INTRQ
0C9H	ADC1EN	TC11EN	TC01EN				P011EN	P001EN	R/W	INTEN
0CAH				CPUM1	CPUM0	CLKMD	STPHX		R/W	OSCM
0CCH	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0	W	WDTR
0CDH	TCOR7	TCOR6	TCOR5	TCOR4	TCOR3	TCOR2	TCOR1	TCOR0	W	TCOR
0CEH	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	R/W	PCL
0CFH							PC9	PC8	R/W	PCH
0D0H		P06	P05	P04	P03	P02	P01	P00	R/W	P0
0D4H				P44	P43	P42	P41	P40	R/W	P4
0D5H				P54	P53				R/W	P5
0D8H					TC1X8	TC0X8	TC0GN		R/W	TOM
0DAH	TC0ENB	TC0rate	TC0rate	TC0rate	TC0CKS	ALOAD0	TC0OUT	PWM0OUT	R/W	TCOM
0DBH	TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0	R/W	TCOC
0DCH	TC1ENB	TC1rate	TC1rate	TC1rate	TC1CKS	ALOAD1	TC1OUT	PWM1OUT	R/W	TC1M
0DDH	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0	R/W	TC1C
0DEH	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0	W	TC1R
0DFH	GIE					STKPB2	STKPB1	STKPB0	R/W	STKP
0E0H		P06R	P05R		P03R	P02R	P01R	P00R	W	P0UR
0E4H				P44R	P43R	P42R	P41R	P40R	W	P4UR
0E5H				P54R	P53R				W	P5UR
0E7H	@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0	R/W	@YZ
0F8H	S3PC7	S3PC6	S3PC5	S3PC4	S3PC3	S3PC2	S3PC1	S3PC0	R/W	STK3L
0F9H							S3PC9	S3PC8	R/W	STK3H
0FAH	S2PC7	S2PC6	S2PC5	S2PC4	S2PC3	S2PC2	S2PC1	S2PC0	R/W	STK2L
0FBH							S2PC9	S2PC8	R/W	STK2H
0FCH	S1PC7	S1PC6	S1PC5	S1PC4	S1PC3	S1PC2	S1PC1	S1PC0	R/W	STK1L
0FDH							S1PC9	S1PC8	R/W	STK1H
0FEH	S0PC7	S0PC6	S0PC5	S0PC4	S0PC3	S0PC2	S0PC1	S0PC0	R/W	STK0L
0FFH							S0PC9	S0PC8	R/W	STK0H

3 功能模块

3.1 系统寄存器

3.1.1 082H-R (工作寄存器和 ROM 查表数据缓存器)

8 位缓存器 R 主要有以下两个功能:

1. 作为工作寄存器使用;
2. 存储执行查表指令后的高字节数据。(执行 MOVC 指令, 指定 ROM 单元的高字节数据会被存入 R 寄存器而低字节数据则存入 ACC。)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0

3.1.2 083H-Z (专用寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0

3.1.3 084H-Y (专用寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
YBIT7	YBIT6	YBIT5	YBIT4	YBIT3	YBIT2	YBIT1	YBIT0

寄存器 Y 和 Z 都是 8 位缓存器, 主要用途如下:

1. 普通工作寄存器;
2. RAM 数据寻址指针@YZ;
3. 配合指令 MOVC 对 ROM 数据进行查表。

3.1.4 086H-PFLAG (ROM 页和特殊标志寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NT0	NPD	LVD36	LVD24	--	C	DC	Z

Bit<7:6>: NP0/NPD-复位状态标识位

NT0	NPD	复位情况	说明
0	0	看门狗复位	看门狗溢出
0	1	保留	-
1	0	上电及 LVD 复位	电源电压低于 LVD 检测值
1	1	外部复位	外部复位引脚检测到低电平

任何一种复位情况都需要一定的响应时间, 系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器, 完成复位所需要的时间也不同。因此, VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的起振时间最短, 晶体振荡器的起振时间则较长。在用户终端使用的过程中, 应注意考虑主机对上电复位时间的要求。

Bit<5>: LVD36-3.6VLVD 工作电压标志位, LVD 编译选项为 LVD_H 时有效

0: 系统工作电压 VDD 超过 3.6V, 低电压检测器没有工作

1: 系统工作电压 VDD 低于 3.6V, 说明此时低电压检测器已处于监控状态

Bit<4>: LVD36-2.4VLVD 工作电压标志位, LVD 编译选项为 LVD_H 时有效

0: 系统工作电压 VDD 超过 2.4V, 低电压检测器没有工作

1: 系统工作电压 VDD 低于 2.4V, 说明此时低电压检测器已处于监控状态

LVD	LVD 编译选项		
	LVD_L	LVD_M	LVD_H
2.0V 复位	有效	有效	有效
2.4V 标志	-	有效	-
2.4V 复位	-	-	有效
3.6V 标志	-	-	有效

LVD_L: 如果 $VDD < 2.0V$, 系统复位;

LVD24 和 LVD36 标志位无意义。

LVD_M: 如果 $VDD < 2.0V$, 系统复位;

LVD24: 如果 $VDD > 2.4V$, $LVD24 = 0$; 如果 $VDD \leq 2.4V$, $LVD24 = 1$;
LVD36 标志位无意义。

LVD_H: 如果 $VDD < 2.4V$, 系统复位;

LVD36: 如果 $VDD > 3.6V$, $LVD36 = 0$; 如果 $VDD \leq 3.6V$, $LVD36 = 1$;

注:

(1) LVD 复位结束后, LVD24 和 LVD36 都将被清零;

(2) LVD 2.4V 和 LVD 3.6V 检测电平值仅作为设计参考, 不能用作芯片工作电压值的精确检测。

3.1.5 087H-IRCCAL (IRC 频率修调寄存器)

3.1.6 0AEH-P4CON (P4 配置控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0

Bit<4:0>: P4CON<4:0>-P4 ADC 端口配置控制位

0: P4 端口可以作为模拟输入 (ADC 输入) 引脚或者 GPIO 引脚

1: P4 端口只能作为模拟输入引脚, 不能作为 GPIO 引脚

注: 当 P4.n 作为普通 I/O 口而不是 ADC 输入引脚时, P4CON.n 必须置为 0, 否则 P4.n 的普通 I/O 信号不能正常输出。

3.1.7 0AFH-VREF (ADC 参考电压寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EVHENB	--	--	--	--	--	VHS1	VHS0

Bit<7>: EVHENB-ADC 内部参考电压控制位

0: 允许 ADC 内部 Vrefh 功能, Vrefh 引脚是 P4.0/AIN0 引脚

1: 禁止 ADC 内部 Vrefh 功能, P4.0/AIN0/Vrefh 引脚来自外部 Vrefh 输入引脚

Bit<1:0>: VHS<1:0>-AD 内部参考电压选择位

VHS1	VHS0	内部 Vrefh 电压
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	VDD

ADC 内置 5 种参考电压, 由 VREFH 寄存器控制: 包括 1 个外部参考电压和 4 个内部参考源 (VDD、4V、3V、2V)。EVHENB = 1 时, ADC 参考电压由外部参考源提供 (P4.0), 必须输入一个电压作为 ADC 参考电压的高电平, 且不能低于 2V。EVHENB = 0 时, ADC 参考电压由内部参考源提供, 并由 VHS[1:0] 选择控制。VHS[1:0] = 11 时, ADC 参考源选择 VDD; VHS[1:0] =

10 时，ADC 参考源选择 4V；VHS[1:0] = 01 时，ADC 参考源选择 3V；VHS[1:0] = 00 时，ADC 参考源选择 2V。外部参考源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。

3.1.8 0B1H-ADM (ADC 模式寄存器)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADENB	ADS	EOC	GCHS	--	CHS2	CHS1	CHS0

Bit<7>:ADENB-ADC 使能控制位（睡眠模式下，禁止 ADC 省电）

0:禁止

1:使能

Bit<6>:ADS-ADC 启动控制位（ADC 处理完成，ADS 位自动清零）

0:停止

1:开始

Bit<5>:EOC-ADC 状态控制位

0:转换进行中

1:转换结束，ADS 复位

Bit<4>:GCHS-ADC 通道选择位

0:禁止 AIN 通道

1:使能 AIN 通道

Bit<2:0>: CHS<2:0>-ADC 输入通道选择位

CHS<2:0>	通道编号	对应引脚
000	AIN0	P40
001	AIN1	P41
010	AIN2	P42
011	AIN3	P43
100	AIN4	P44
101	AIN5	内部 1/4VDD 通道

AIN5 是内部 1/4VDD 输入通道，外部没有输入引脚。AIN5 可以作为电池系统的电池检测。为了选择合适的内部 Vref 电平并进行比较，系统配置了高性能/廉价的电池检测通道。

3.1.9 0B2H-ADB (ADC 数据缓存器)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

Bit<7:0>:ADB<7:0>-ADC 转换数据高 8 位（只读）

3.1.10 0B3H-ADR (ADC 精度选择寄存器)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	ADCKS1	--	ADCKS0	ADB3	ADB2	ADB1	ADB0

Bit<6, 4>:ADCKS<1:0>-ADC 时钟源选择位

ADCKS1	ADCKS0	ADC 时钟源
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

Bit<3:0>:ADB<3:0>-ADC 转换数据低 4 位

3.1.11 0B4H-ADT (ADC 零漂寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTS1	ADTS0	--	ADT4	ADT3	ADT2	ADT1	ADT0

3.1.12 0B8H-P0M (P0M 方向控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06M	P05M	P04M	P03M	P02M	P01M	P00M

Bit<6:0>:P0M<6:0>-P0 口方向控制位

0:输入模式

1:输出模式

注:P04M 只能作为输入口, P04M=0;

3.1.13 0BFH-PEDGE (P00 模式控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P00G1	P00G0	--	--	--

Bit<4:3>:P00G<1:0>-P00 外部中断触发控制位

00: 保留

01: 上升沿触发

10: 下降沿触发

11: 双沿触发

3.1.14 0C4H-P4M (P4M 模式控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44M	P43M	P42M	P41M	P40M

Bit<4:0>:P4M<4:0>-P4 口方向控制位

0:输入模式

1:输出模式

3.1.15 0C5H-P5M (P5M 模式控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54M	P53M	--	--	--

Bit<4:3>:P5M<4:3>-P5 口方向控制位

0:输入模式)

1:输出模式

3.1.16 0C8H-INTRQ (中断请求寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIRQ	TC1IRQ	TC0IRQ	--	--	--	P01IRQ	P00IRQ

Bit<7>:ADCIRQ-ADC 中断标志位

Bit<6>:TC1IEN-TC1 中断标志位

Bit<5>:TC0IEN-TC0 中断标志位

Bit<1>:P01IEN-P01 外部中断标志位

Bit<0>:P00IEN-P00 外部中断标志位

0:无中断请求

1:有中断请求

3.1.17 0C9H-INTEN (中断使能寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIEN	TC1IEN	TC0IEN	--	--	--	P01IEN	P00IEN

Bit<7>:ADCIEN-ADC 中断控制位

Bit<6>:TC1IEN-TC1 中断控制位

Bit<5>:TC0IEN-TC0 中断控制位

Bit<1>:P01IEN-P01 外部中断控制位

Bit<0>:P00IEN-P00 外部中断控制位

0:禁止

1:使能

3.1.18 0CAH-OSCM (振动模式寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	CPUM1	CPUM0	CLKMD	STPHX	--

Bit<4:3>:CPUM<1:0>-CPU 工作模式控制位

CPUS1	CPUS0	工作模式
0	0	普通模式
0	1	睡眠模式
1	0	绿色模式
1	1	保留

Bit<2>:CLKMD-系统时钟模式控制位

0:普通(双时钟)模式,高速时钟作为系统时钟

1:低速模式,低速时钟作为系统时钟

Bit<1>:STPHX-高速振荡器控制位

0:运行

1:停止,内部低速 RC 振荡器仍然运行

3.1.19 0CCH-WDTR (看门狗清零寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0

看门狗清零的方法是对看门狗计数器清零寄存器 WDTR 写入清零控制字 5AH。

3.1.20 OCDH-TCOR (TC0 自动装载数据缓存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCOR7	TCOR6	TCOR5	TCOR4	TCOR3	TCOR2	TCOR1	TCOR0

TCOR 初始值计算公式:TCOR 初始值= N- (TC0 中断间隔时间*输入时钟)

N 是 TC0 最大溢出值。TC0 的溢出时间和有效值见下表:

TCOCKS	TCOX8	PWM0	ALOAD0	TCOOUT	N	TCOR 有效值
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH

	$(F_{osc}/1 \sim F_{osc}/128)$	1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

例:TC0 中断间隔时间设置为 10ms,时钟源选 Fcpu(TC0KS=0,TC0X8 = 0),无 PWM 输出(PWM0=0),高速时钟为外部 4MHz, Fcpu=Fosc/4, TCORATE=010 (Fcpu/64)。

$$\begin{aligned}
 TCOR &= N - (TC0 \text{ 中断间隔时间} * \text{输入时钟}) \\
 &= 256 - (10\text{ms} * 4\text{MHz} / 4 / 64) \\
 &= 256 - (0.01 * 4000000 / 4 / 64) \\
 &= 100 = 64\text{H}
 \end{aligned}$$

3.1.21 OCEH-PCL (程序计数器低位)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

3.1.22 OCFH-PCH (程序计数器高位)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	--	--	PC9	PC8

程序计数器 PC 是一个 10 位二进制程序地址寄存器,分高 2 位和低 8 位。专门用来存放下一条需要执行指令的内存地址。通常,程序计数器会随程序中指令的执行自动增加。若程序执行 CALL 和 JMP 指令时,PC 指向特定的地址。

3.1.23 OD0H-P0 (P0 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06	P05	P04	P03	P02	P01	P00

Bit<6:0>:P0<6:0>-P0 数据寄存器

注:P04 使能外部复位时,P04 保持为“1”

3.1.24 OD4H-P4 (P4 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44	P43	P42	P41	P40

Bit<4:0>:P4<4:0>-P4 数据寄存器

3.1.25 OD5H-P5 (P5 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54	P53	--	--	--

Bit<4:3>:P5<4:3>-P5 数据寄存器

3.1.26 OD8H-TOM (TC 加速和 TC0 唤醒功能寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TC0X8	TC0GN	--

Bit<3>:TC1X8-TC1 内部时钟选择控制位

0:TC1 内部时钟来自 Fcpu, TC1RATE = Fcpu/2~Fcpu/256

1:TC1 内部时钟来自 Fosc, TC1RATE = Fosc/1~Fosc/128

注:TC1CKS=1 时,TC1X8 和 TC1RATE 可以忽略不计

Bit<2>:TC0X8-TC0 内部时钟选择控制位

0:TC0 内部时钟来自 Fcpu, TCORATE = Fcpu/2~Fcpu/256

1:TC0 内部时钟来自 Fosc, TCORATE = Fosc/1~Fosc/128

Bit<1>:TC0GN-TC0 绿色模式唤醒功能控制位

0:禁止 TC0 的唤醒功能

1:允许 TC0 的唤醒功能

注:TC1CKS=1 时, TC0X8 和 TCORATE 可以忽略不计

3.1.27 ODAH-TCOM (TC0 模式寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0ENB	TC0rate2	TC0rate1	TC0rate0	TC0CKS	ALOADO	TC0OUT	PWM0OUT

Bit<7>:TC0ENB-TC0 启动控制位

0:关闭

1:打开

Bit<6:4>:TC0rate<2:0>:TC0 分频选择位

TC0rate<2:0>	TC0X8=0	TC0X8=1
000	Fcpu/256	Fcpu/128
001	Fcpu/128	Fcpu/64
010	Fcpu/64	Fcpu/32
011	Fcpu/32	Fcpu/16
100	Fcpu/16	Fcpu/8
101	Fcpu/8	Fcpu/4
110	Fcpu/4	Fcpu/2
111	Fcpu/2	Fcpu/1

Bit<3>:TC0CKS-TC0 时钟信号控制位

0:内部时钟 (Fcpu 或 Fosc)

1:外部时钟, 由 P0.0/INT0 输入

Bit<2>:ALOADO-自动装载控制位 仅当 PWM0OUT=0 时有效

0:禁止 TC0 自动重装

1:允许 TC0 自动重装

Bit<1>:TC0OUT-TC0 溢出信号输出控制位 仅当 PWM0OUT=0 时有效

0:禁止, P5.4 作为输入/输出口

1:允许, P5.4 输出 TC0OUT 信号

Bit<0>:PWM0OUT-PWM 输出控制

0:禁止 PWM 输出

1:使能 PWM 输出, PWM 输出占空比由 TOOUT 和 ALOADO 控制

注:若 TC0CKS=1, 则 TC0 用作外部事件计数器, 此时不需要考虑 TCORATE 的设置, P0.0 口无中断信号 (P00IRQ=0)

3.1.28 ODBH-TCOC (TC0 计数寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0

TCOC 初始值计算公式:TCOC 初始值= N - (TC0 中断间隔时间*输入时钟)

N 为 TC0 二进制计数范围。各模式下参数的设定如下表所示:

TC0CKS	TC0X8	PWM0	ALOAD0	TC0OUT	N	TC0C 有效值
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.1.29 ODCH-TC1M (TC1 模式寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1ENB	TC1rate2	TC1rate1	TC1rate0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT

Bit<7>:TC1ENB-TC1 启动控制位

0:关闭

1:打开

Bit<6:4>:TC1rate<2:0>:TC0 分频选择位

TC1rate<2:0>	TC1X8=0	TC1X8=1
000	Fcpu/256	Fcpu/128
001	Fcpu/128	Fcpu/64
010	Fcpu/64	Fcpu/32
011	Fcpu/32	Fcpu/16
100	Fcpu/16	Fcpu/8
101	Fcpu/8	Fcpu/4
110	Fcpu/4	Fcpu/2
111	Fcpu/2	Fcpu/1

Bit<3>:TC1CKS-TC1 时钟信号控制位

0:内部时钟 (Fcpu 或 Fosc, 由 TC1X8 位控制)

1:外部时钟, 由 P0.1/INT1 输入, 使能时间计数器功能。TC1Rate<2:0>位处于无效状态

Bit<2>:ALOAD1-自动装载控制位 仅当 PWM1OUT=0 时有效

0:禁止 TC1 自动重装

1:允许 TC1 自动重装

Bit<1>:TC1OUT-TC1 超时输出信号控制。仅当 PWM1OUT=0 时有效

0:禁止, P5.3 为 GPIO 引脚

1:使能, P5.3 输出 TC1/2 Buzzer 信号

Bit<0>:PWM1OUT-PWM 输出控制

0:禁止 PWM 输出, P5.3 为 GPIO 引脚

1:使能 PWM 输出, P5.3 输出 PWM 信号, PWM 输出占空比由 TC1OUT 和 ALOAD1 控制

注:若 TC1CKS=1, 则 TC1 用作外部事件计数器, 此时不需要考虑 TC1RATE 的设置, P0.1 口无中断信号 (P0.1IRQ=0)

3.1.30 ODDH-TC1C (TC1 计数寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0

TC1C 初始值的计算公式: $TC1C \text{ 初始值} = N - (TC1 \text{ 中断间隔时间} * \text{输入时钟})$

N 为 TC1 二进制计数范围。各模式下参数的设定如下表所示:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1C 有效值
0	0 ($F_{cpu}/2^{\sim}$ $F_{cpu}/256$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 ($F_{osc}/1^{\sim}$ $F_{osc}/128$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.1.31 ODEH-TC1R (TC1 自动装载数据缓存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0

TC1R 初始值计算公式: $TC1R \text{ 初始值} = N - (TC1 \text{ 中断间隔时间} * \text{输入时钟})$

N 是 TC1 最大溢出值。TC1 的溢出时间和有效值见下表:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1R 有效值
0	0 ($F_{cpu}/2^{\sim}$ $F_{cpu}/256$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 ($F_{osc}/1^{\sim}$ $F_{osc}/128$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

例: TC0 中断间隔时间设置为 10ms, 时钟源选 F_{cpu} ($TC0KS=0, TC0X8=0$), 无 PWM 输出 ($PWM0=0$), 高速时钟为外部 4MHz, $F_{cpu}=F_{osc}/4$, $TCORATE=010$ ($F_{cpu}/64$)。

$$\begin{aligned}
 TCOR &= N - (TC0 \text{ 中断间隔时间} * \text{输入时钟}) \\
 &= 256 - (10\text{ms} * 4\text{MHz} / 4 / 64) \\
 &= 256 - (0.01 * 4000000 / 4 / 64) \\
 &= 100 = 64H
 \end{aligned}$$

3.1.32 0DFH-STKP (堆栈指针)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	--	--	--	--	STKPB2	STKPB1	STKPB0

Bit<7>:GIE-全局中断控制位

- 0:禁止
- 1:使能

Bit<2:0>:STKPB<2:0>:堆栈指针

例:系统复位,堆栈指针寄存器内容为默认值,但建议在程序初始部分重新设定:

```
MOV    A, #00000111B
BOMOV STKP, A
```

3.1.33 0E0H-P0UR (P0 上拉电阻控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06R	P05R	--	P03R	P02R	P01R	P00R

Bit<6:0>:POUP<6:0>-P0 上拉使能控制位

- 0:禁止
- 1:使能

注:P0.4 是单向输入引脚,无上拉电阻,因此 P0UR.4 始终为“1”

3.1.34 0E4H-P4UR (P4 上拉电阻控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44R	P43R	P42R	P41R	P40R

Bit<4:0>:P4UR<4:0>-P4 上拉使能控制位

- 0:禁止
- 1:使能

3.1.35 0E5H-P5UR (P5 上拉电阻控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54R	P53R	--	--	--

Bit<4:3>:P5DR<4:3>-P5 上拉使能控制位

- 0:禁止
- 1:使能

3.1.36 0E7H-@YZ (间接寻址寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0

通过指针寄存器 (Y/Z) 访问 RAM 数据

例:用@YZ 实现间接寻址

```
BOMOV    Y, #0        ;Y 清零以寻址 RAM bank 0。
BOMOV    Z, #12H     ;设定寄存器地址。
BOMOV    A, @YZ
```

3.1.37 0F8H~0FFH-STK0~STK3 (堆栈寄存器)

3.2 GPIO 功能模块

YF2712J 有 3 组双向 I/O 端口，共 14 个输入，13 个输出，大部分 I/O 可以复用为其它功能；13 个可编程上拉 I/O 引脚：P00~P03, P05~P06, P40~P44, P53~P54；

端口输入特性表格如下：

端口	翻转特性
P00~P01, P05~P06	0. 8VDD/0. 3VDD
P02~P03	0. 7VDD/0. 4VDD
P04	0. 7VDD/0. 2VDD
P40~P44	0. 8VDD/0. 3VDD
P53~P54	0. 8VDD/0. 3VDD

3.2.1 GPIO 寄存器说明

0B8H (P0M 方向控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06M	P05M	-	P03M	P02M	P01M	P00M

Bit<6:0>:P0M<6:0>-P0 口方向控制位

0:输入模式

1:输出模式

注:P04M 只能作为输入口，P04M=0；

0C4H (P4M 方向控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44M	P43M	P42M	P41M	P40M

Bit<4:0>:P4M<4:0>-P4 口方向控制位

0:输入模式 (默认)

1:输出模式

0C5H (P5M 方向控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54M	P53M	-	-	-

Bit<4:3>:P5M<4:3>-P5 口方向控制位

0:输入模式 (默认)

1:输出模式

0D0H (P0 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06	P05	-	P03	P02	P01	P00

Bit<6:0>:P0<6:0>-P0 数据寄存器

注:P04 使能外部复位时，P04 保持为“1”；

0D4H (P4 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44	P43	P42	P41	P40

Bit<4:0>:P4<4:0>-P4 数据寄存器

0D5H (P5 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54	P53	-	-	-

Bit<4:3>:P5<4:3>-P5 数据寄存器

0E0H (POUR 上拉控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06R	P05R	-	P03R	P02R	P01R	P00R

Bit<6:0>:POUP<6:0>-P0 上拉使能控制位

0:禁止 (默认)

1:使能

注:P04 使能外部复位时,无上拉电阻,P04 保持为“1”;

0E4H (P4UR 上拉控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44R	P43R	P42R	P41R	P40R

Bit<4:0>:P4UR<4:0>-P4 上拉使能控制位

0:禁止 (默认)

1:使能

0E5H (P5UR 上拉控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54R	P53R	-	-	-

Bit<4:3>:P5DR<4:3>-P5 上拉使能控制位

0:禁止 (默认)

1:使能

3.3 TCC 定时器功能模块

3.3.1 看门狗定时器

看门狗定时器 WDT 是一个 4 位二进制计数器，用于监控程序的正常执行。如果由于干扰，程序进入了未知状态，看门狗定时器溢出，系统复位。看门狗的工作模式由编译选项控制，其时钟源由内部低速 RC 振荡器（16KHz @3V，32KHz @5V）提供。

看门狗溢出时间 = 8192 / 内部低速振荡器周期 (sec)

VDD	内部低速 RC	看门狗溢出时间
3V	16KHz	512ms
5V	32KHz	256ms

看门狗定时器的 3 种工作模式由编译选项“WatchDog”控制：

Disable: 禁止看门狗定时器功能。

Enable: 使能看门狗定时器功能，在普通模式和低速模式下有效；在睡眠模式和绿色模式下看门狗停止工作。

Always_On: 使能看门狗定时器功能，在睡眠模式和绿色模式下，看门狗仍会正常工作。

在高干扰环境下，强烈建议将看门狗设置为“Always_On”以确保系统在出错状态和重启时正常复位。

看门狗清零的方法是对看门狗计数器清零寄存器 WDTR 写入清零控制字 5AH。

0CCH (WDTR 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0

例如：对看门狗定时器的操作，在主程序开头对看门狗清零。

```
MOV      A, #5AH      ; 看门狗定时器清零。
BOMOV   WDTR, A
```

看门狗定时器应用注意事项如下：

- (1) 对看门狗清零之前，检查 I/O 口的状态和 RAM 的内容可增强程序的可靠性；
- (2) 不能在中断中对看门狗清零，否则无法侦测到主程序跑飞的状况；
- (3) 程序中应该只在主程序中有一次清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

3.3.2 定时/计数器 TC0

3.3.2.1 概述

8 位二进制定时/计数器具有基本定时器、事件计数器、PWM 和 Buzzer 功能。基本定时器功能可以支持标志显示 (TC0IRQ) 和中断操作 (中断向量)。由 TC0M、TC0C、TC0R 寄存器控制 TC0 的中断间隔时间。事件计数器可以将 TC0 时钟源由系统时钟更改为外部时钟信号 (如连续的脉冲、R/C 振荡信号等)。TC0 作为计数器时记录外部时钟数目以进行测量应用。TC0 还内置周期/占空比可编程控制的 PWM 功能, PWM 的周期和分辨率由 TC0M 和 TC0R 寄存器控制。TC0 还内置 Buzzer 功能, 以输出 TC0/2 信号。TC0 支持自动重装功能。TC0 溢出时, TC0R 的值自动装入 TC0C。TC0 内置绿色模式唤醒功能, 由 TC0GN 控制。

TC0 的主要用途如下:

- (1) 8 位可编程定时器: 根据选择的时钟信号, 产生周期中断;
- (2) 中断功能: TC0 定时器支持中断, 当 TC0 溢出时, TC0IRQ 置 1, 系统执行中断;
- (3) 外部事件计数器: 对外部事件计数;
- (4) PWM 输出: 由 TC0rate, TC0R 寄存器和 TC0M 寄存器的 ALOAD0 和 TC0OUT 位控制占空比/周期;
- (5) Buzzer 输出: Buzzer 输出信号为 TC0 间隔时间的 1/2 周期;
- (6) 绿色模式功能: TC0 溢出时, TC0 内置绿色模式唤醒功能, 由 TC0GN 控制。

3.3.2.2 TC0 操作

TC0 定时器由 TC0ENB 控制。当 TC0ENB=0 时, TC0 停止工作; 当 TC0ENB=1 时, TC0 开始计数。使能 TC0 之前, 先要设定好 TC0 的功能模式, 如基本定时器、TC0 中断等。TC0C 溢出 (从 0FFH 到 00H) 时, TC0IRQ 置 1 以显示溢出状态并由程序清零。在不同的功能模式下, TC0C 不同的值对应不同的操作, 若改变 TC0C 的值影响到操作, 会导致功能出错。TC0 内置双重缓存器以避免此种状况的发生。在 TC0C 计数的过程中不断的刷新 TC0C, 保证将最新的值存入 TC0R (重装缓存器) 中, 当 TC0 溢出后, TC0R 的值由自动存入 TC0C。进入下一个周期后, TC0 按新的配置工作。使能 TC0 时, 自动使能 TC0 的自动重装功能。如果使能 TC0 中断功能 (TC0IEN=1), 在 TC0 溢出时系统执行中断服务程序, 在中断时必须由程序清 TC0IRQ。TC0 可以在普通模式、低速模式和绿色模式下工作。但在绿色模式下, TC0 虽继续工作, 设置 TC0IRQ 和 PWM 输出、Buzzer 功能, 由 TC0GN 控制将系统唤醒。

TC0 根据不同的时钟源选择不同的应用模式, TC0 的时钟源由 Fcpu (指令周期)、Fhosc (高速振荡时钟) 和外部引脚输入 (P0.0) 提供, 由 TC0CKS 和 TC0X8 控制。TC0X8 选择时钟源来自 Fcpu 或者 Fhosc, 当 TC0X8=0 时, TC0 时钟源来自 Fcpu, 可以由 TC0Rate[2:0] 选择不同的分频。当 TC0X8=1 时, TC0 时钟源来自 Fhosc, 可以由 TC0Rate[2:0] 选择不同的分频。TC0CKS 决定时钟源由外部引脚输入或者由 TC0X8 控制, TC0CKS=0 时, TC0 的时钟源由 TC0X8 控制, TC0CKS=1 时, TC0 时钟源由外部输入引脚提供, 此时使能外部事件计数功能。TC0X8=1 时, TC0Rate[2:0] 处于无效状态。

TC0CK S1	TC0Rate[2:0]	TC0 时钟	TC0 间隔时间			
			Fhosc=16MHz Fcpu=Fhosc/4		Fhosc=4MHz Fcpu=Fhosc/4	
			max. (ms)	Unit (us)	max. (ms)	Unit (us)
0	000b	Fcpu/256	16.384	64	65.536	256
0	001b	Fcpu/128	8.192	32	32.768	128
0	010b	Fcpu/64	4.096	16	16.384	64
0	011b	Fcpu/32	2.048	8	8.192	32
0	100b	Fcpu/16	1.024	4	4.096	16
0	101b	Fcpu/8	0.512	2	2.048	8
0	110b	Fcpu/4	0.256	1	1.024	4
0	111b	Fcpu/2	0.128	0.5	0.512	2
1	000b	Fhosc/128	2.048	8	8.192	32
1	001b	Fhosc/64	1.024	4	4.096	16
1	010b	Fhosc/32	0.512	2	2.048	8
1	011b	Fhosc/16	0.256	1	1.024	4
1	100b	Fhosc/8	0.128	0.5	0.512	2
1	101b	Fhosc/4	0.064	0.25	0.256	1
1	110b	Fhosc/2	0.032	0.125	0.128	0.5
1	111b	Fhosc/1	0.016	0.0625	0.064	0.25

3.3.2.3 TC0M 模式寄存器

模式寄存器 TC0M 控制 TC0 的工作模式，包括 TC0 分频、时钟源和 PWM 功能等。这些设置必须在使能 TC0 定时器之前完成。

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0ENB	TC0rate2	TC0rate1	TC0rate0	TC0CKS	ALOAD0	TC0OUT	PWM0OUT

Bit<7>:TC0ENB-TC0 启动控制位

0:关闭

1:打开

Bit<6:4>:TC0rate<2:0>:TC0 分频选择位

TC0rate<2:0>	TC0X8=0	TC0X8=1
000	Fcpu/256	Fcpu/128
001	Fcpu/128	Fcpu/64
010	Fcpu/64	Fcpu/32
011	Fcpu/32	Fcpu/16
100	Fcpu/16	Fcpu/8
101	Fcpu/8	Fcpu/4
110	Fcpu/4	Fcpu/2
111	Fcpu/2	Fcpu/1

Bit<3>:**TCOCKS**-TC0 时钟信号控制位

- 0: 内部时钟 (Fcpu 或 Fosc)
- 1: 外部时钟, 由 P0.0/INT0 输入

Bit<2>:**ALOADO**-自动装载控制位 仅当 PWM0OUT=0 时有效

- 0: 禁止 TC0 自动重装
- 1: 允许 TC0 自动重装

Bit<1>:**TC0OUT**-TC0 溢出信号输出控制位 仅当 PWM0OUT=0 时有效

- 0: 禁止, P5.4 作为输入/输出口
- 1: 允许, P5.4 输出 TC0OUT 信号

Bit<0>:**PWM0OUT**-PWM 输出控制

- 0: 禁止 PWM 输出
- 1: 使能 PWM 输出, PWM 输出占空比由 T0OUT 和 ALOADO 控制

注:若 TCOCKS=1, 则 TC0 用作外部事件计数器, 此时不需要考虑 TCORATE 的设置, P0.0 口无中断信号 (P00IRQ=0)

3.3.2.4 TCOX8/TCOEN 标志

TC0 时钟源现在和绿色模式唤醒功能有 T0M 寄存器控制。

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TCOX8	TCOEN	--

Bit<2>:**TCOX8**-TC0 内部时钟选择控制位

- 0: TC0 内部时钟来自 Fcpu, $TCORATE = Fcpu/2^{\sim}Fcpu/256$
- 1: TC0 内部时钟来自 Fosc, $TCORATE = Fosc/1^{\sim}Fosc/128$

Bit<1>:**TCOEN**-TC0 绿色模式唤醒功能控制位

- 0: 禁止 TC0 的唤醒功能
- 1: 允许 TC0 的唤醒功能

注:TC1CKS=1 时, TCOX8 和 TCORATE 可以忽略不计

3.3.2.5 TCOC 计数寄存器

8 位计数器 TCOC 溢出时, TC0IRQ 置 1 并由程序清零, 用来控制 TC0 的中断间隔时间。首先须写入正确的值到 TCOC 和 TCOR 寄存器, 并使能 TC0 定时器以保证第一个周期正确。TC0 溢出后, TCOR 的值自动装入 TCOC。

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCOC7	TCOC6	TCOC5	TCOC4	TCOC3	TCOC2	TCOC1	TCOC0

TCOC 初始值计算公式: TCOC 初始值 = N - (TC0 中断间隔时间 * 输入时钟)

N 为 TC0 二进制计数范围。各模式下参数的设定如下表所示:

TCOCKS	TCOX8	PWM0	ALOAD0	TCOOUT	N	TCOC 有效值
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.3.2.6 TCOR 自动装载寄存器

TC0 内置自动重装功能，TCOR 寄存器存储重装值。TCOC 溢出时，TCOR 的值自动装入 TCOC 中。TC0 定时器工作在计时模式时，要通过修改 TCOR 寄存器来修改 TC0 的间隔时间，而不是通过修改 TCOC 寄存器。在 TC0 定时器溢出后，新的 TCOC 值会被更新，TCOR 会将新的值装载到 TCOC 寄存器中。但在初次设置 TCOM 时，必须要在开启 TC0 定时器前把 TCOC 以及 TCOR 设置成相同的值。

TC0 为双重缓存器结构。若程序对 TCOR 进行了修改，那么修改后的 TCOR 值首先被暂存在 TCOR 的第一个缓存器中，TC0 溢出后，TCOR 的新值就会被存入 TCOR 缓存器中，从而避免 TC0 中断时间出错以及 PWM 误动作。

注：在 PWM 模式下，系统自动开启重装功能，ALOAD0 用于控制溢出范围。

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCOR7	TCOR6	TCOR5	TCOR4	TCOR3	TCOR2	TCOR1	TCOR0

TCOR 初始值计算公式：TCOR 初始值 = N - (TC0 中断间隔时间 * 输入时钟)

N 是 TC0 最大溢出值。TC0 的溢出时间和有效值见下表：

TCOCKS	TCOX8	PWM0	ALOAD0	TCOOUT	N	TCOR 有效值
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

例:TC0 中断间隔时间设置为 10ms,时钟源选 Fcpu(TC0KS=0,TC0X8 = 0),无 PWM 输出(PWM0=0),高速时钟为外部 4MHz, Fcpu=Fosc/4, TCORATE=010 (Fcpu/64)。

$$\begin{aligned}
 TCOR &= N - (TC0 \text{ 中断间隔时间} * \text{输入时钟}) \\
 &= 256 - (10ms * 4MHz / 4 / 64) \\
 &= 256 - (0.01 * 4000000 / 4 / 64) \\
 &= 100 = 64H
 \end{aligned}$$

3.3.2.7 TC0 事件计数器

TC0 作为外部事件计数器时,其时钟源由外部输入引脚(P0.0)提供。当 TC0CKS=1 时,TC0 的时钟源由外部输入引脚(P0.0)提供,下降沿触发,下降沿触发时,TC0C 开始计数。TC0C 溢出(从 FFH 到 00H)时,TC0 触发事件计数器溢出。使能外部事件计数功能,同时外部输入引脚的唤醒功能被禁止以避免外部事件的触发信号将系统唤醒而耗电。此时,P0.0 的外部中断功能也被禁止,即 P00IRQ=0。外部事件计数器通常用来测量外部连续信号的比率,如连续的脉冲信号,R/C 振荡信号等,外部信号的相位与 MCU 时钟的相位并不同步,通过 TC0 事件计数器的测量和计算以达到不同的应用。

3.3.2.8 TC0 时钟频率输出

Buzzer 输出(TC0OUT)为定时/计数器 TC0 频率输出功能,通过设置 TC0 时钟频率,时钟信号输出到 P5.4,此时自动禁止 P5.4 的普通 I/O 功能。TC0 间隔时间 2 分频后作为 TC0OUT 频率。通过 TC0 时钟可以获得不同的频率。

TC0 溢出后,Buzzer 输出时,TC0IRQ 有效,且当 TC0IEN=1 时,使能 TC0 中断功能。但强烈建议小心同时使用 Buzzer 和 TC0 定时器,以确保两种功能都能正常工作。

Buzzer 输出引脚与 GPIO 引脚共用,TC0OUT=1 时,该引脚自动设为 Buzzer 输出引脚。如清 TC0OUT 位以禁止 Buzzer 输出后,该引脚自动返回到最后一个 GPIO 模式。

例如:设置 TC0 输出到 TC0OUT(P5.4)。外部高速时钟 Fosc 为 4MHz,指令周期 Fcpu=Fosc/4,TC0OUT 频率为 0.5KHz。TC0OUT 信号为 2 分频,TC0 的时钟频率为 1KHz。TC0 的时钟源为外部振荡时钟,TC0rate=Fcpu/8,

TCORATE2~TCORATE0=101, TC0C=TCOR=131。

MOV A, #01010000B

BOMOV TCOM, A ; TC0 速率 = Fcpu/8。

MOV A, #131 ;自动装载参考值设置。

BOMOV TC0C, A

BOMOV TCOR, A

BOBSET FTC0OUT ;TC0 的输出信号由 P5.4 输出,禁止 P5.4 的普通 I/O 功能。

BOBSET FALOADO ;使能 TC0 自动重装功能。

BOBSET FTCOENB ;开启 TC0 定时器。

注:蜂鸣器的输出有效时,“PWM0OUT”必须被置为“0”。

3.3.2.9 脉冲宽度调制 PWM

可编程控制占空比/周期的 PWM 可以提供不同的 PWM 信号。使能 TC0 定时器且 PWM0OUT=1 时，由 PWM 输出引脚输出 PWM 信号。PWM 首先输出高电平，然后输出低电平。TCORATE、ALOAD0 和 TCOOUT 位控制 PWM 的周期，

TCOR 控制 PWM 的占空比（脉冲高电平的长度）。使能 TC0 定时器时，设置 TCOC 的初始值为 0。当 TCOC=TCOR 时，

PWM 输出低电平；TC0 溢出时（TCOC 的值从 0FFH 到 00H），整个 PWM 周期完成，并进入下一个周期。在 PWM 输出的过程由程序更改 PWM 的周期，则在下一个周期开始输出新的占空比的 PWM 信号。

PWM 内置 4 种可编程控制的分辨率（1/256、1/64、1/32、1/16），在 PWM0OUT = 1 时由 ALOAD0 和 TCOOUT 位控制。

PWM0	ALOAD0	TCOOUT	PWM 分辨率	TCOR 有效值	TCOR 值（二进制）
1	0	0	256	00H~0FFH	00000000B~11111111B
1	0	1	64	00H~3FH	xx000000B~xx111111B
1	1	0	32	00H~1FH	xxx00000B~xxx11111B
1	1	1	16	00H~0FH	xxxx0000B~xxxx1111B

PWM 的高电平宽度（PWM 占空比）由 TCOR 控制。TCOC = TCOR 时，PWM 输出低电平。PWM 输出过程中，TC0 溢出时，TCOIRQ 有效，TCOIEN=1 时，即使能 TC0 中断时，PWM 模式下的 TC0 中断间隔时间与 PWM 的周期相等，即 TC0 中断区域有 4 种不同的分辨率和 ALOAD0、TCOOUT 值。但强烈建议小心同时使用 PWM 和 TC0 定时器功能，保证两种功能都能正常工作。

PWM 的输出引脚与 GPIO 共用，PWM0OUT=1 时，自动输出 PWM 信号；PWM0OUT=0，即禁止 PWM 时，该引脚自动返回到上一个 GPIO 模式。这样有利于处理 ON/OFF 操作的载波信号，而不控制 TCOENB 位。

3.3.3 定时/计数器 TC1

3.3.3.1 概述

8 位二进制定时/计数器具有基本定时器、事件计数器、PWM 和 Buzzer 功能。基本定时器功能可以支持标志显示 (TC1IRQ) 和中断操作 (中断向量)。由 TC1M、TC1C、TC1R 寄存器控制 TC1 的中断间隔时间。事件计数器可以将 TC1 时钟源由系统时钟更改为外部时钟信号 (如连续的脉冲、R/C 振荡信号等)。TC1 作为计数器时记录外部时钟数目以进行测量应用。TC1 还内置周期/占空比可编程控制的 PWM 功能, PWM 的周期和分辨率由 TC1M 和 TC1R 寄存器控制。TC1 还内置 Buzzer 功能, 以输出 TC1/2 信号。TC1 支持自动重装功能。TC1 溢出时, TC1R 的值自动装入 TC1C。

TC1 的主要用途如下:

- (1) 8 位可编程定时器: 根据选择的时钟信号, 产生周期中断;
- (2) 中断功能: TC1 定时器支持中断, 当 TC1 溢出时, TC1IRQ 置 1, 系统执行中断;
- (3) 外部事件计数器: 对外部事件计数;
- (4) PWM 输出: 由 TC1rate, TC1R 寄存器和 TC1M 寄存器的 ALOAD1 和 TC1OUT 位控制占空比/周期;
- (5) Buzzer 输出: Buzzer 输出信号为 TC1 间隔时间的 1/2 周期;
- (6) 绿色模式功能: TC1 正常工作, 但不能将系统唤醒。

3.3.3.2 TC1 操作

TC1 定时器由 TC1ENB 控制。当 TC1ENB=0 时, TC1 停止工作; 当 TC1ENB=1 时, TC1 开始计数。使能 TC1 之前, 先要设定好 TC1 的功能模式, 如基本定时器、TC1 中断等。TC1C 溢出 (从 0FFH 到 00H) 时, TC1IRQ 置 1 以显示溢出状态并由程序清零。在不同的功能模式下, TC1C 不同的值对应不同的操作, 若改变 TC1C 的值影响到操作, 会导致功能出错。TC1 内置双重缓存器以避免此种状况的发生。在 TC1C 计数的过程中不断的刷新 TC1C, 保证将最新的值存入 TC1R (重装缓存器) 中, 当 TC1 溢出后, 新的 TC1R 值将自动装载到 TC1C。进入下一个周期后, TC1 进行新的工作状态。定时/计数器模式时, 使能 TC1 时, 自动使能自动重装功能。如果使能 TC1 中断功能 (TC1IEN=1), 在 TC1 溢出时系统执行中断服务程序, 在中断时必须由程序清 TC1IRQ。TC1 可以在普通模式、低速模式和绿色模式下工作。但在绿色模式下, TC1 虽继续工作, 但不能唤醒系统。

TC1 根据不同的时钟源选择不同的应用模式, TC1 的时钟源由 Fcpu (指令周期)、Fhosc (高速振荡时钟) 提供和外部输入引脚 P0.1 提供, 由 TC1CKS 和 TC1X8 位控制。TC1X8 选择时钟源来自 Fcpu 或者 Fhosc, 当 TC1X8=0 时, TC1 时钟源来自 Fcpu, 可以由 TC1Rate[2:0] 选择不同的分频, 包括 $F_{cpu}/2^{\sim}F_{cpu}/256$ 。当 TC1X8=1 时, TC1 时钟源来自 Fhosc, 可以由 TC1Rate[2:0] 选择不同的分频, 包括 $F_{cpu}/1^{\sim}F_{cpu}/128$ 。TC1CKS 选择时钟源来自外部输入引脚或由 TC1X8 位控制, TC1CKS=0 时, TC1 的时钟源由 TC1X8 位控制; TC1CKS=1 时, TC1 的时钟源由外部输入引脚提供, 即使能事件计数器功能, 此时 TC1Rate[2:0] 处于无效状态。

TC1X8	TC1Rate[2:0]	TC1 时钟	TC1 间隔时间			
			Fhosc=16MHz Fcpu=Fhosc/4		Fhosc=4MHz Fcpu=Fhosc/4	
			max. (ms)	Unit (us)	max. (ms)	Unit (us)
0	000b	Fcpu/256	16.384	64	65.536	256
0	001b	Fcpu/128	8.192	32	32.768	128
0	010b	Fcpu/64	4.096	16	16.384	64
0	011b	Fcpu/32	2.048	8	8.192	32
0	100b	Fcpu/16	1.024	4	4.096	16
0	101b	Fcpu/8	0.512	2	2.048	8
0	110b	Fcpu/4	0.256	1	1.024	4
0	111b	Fcpu/2	0.128	0.5	0.512	2
1	000b	Fhosc/128	2.048	8	8.192	32
1	001b	Fhosc/64	1.024	4	4.096	16
1	010b	Fhosc/32	0.512	2	2.048	8
1	011b	Fhosc/16	0.256	1	1.024	4
1	100b	Fhosc/8	0.128	0.5	0.512	2
1	101b	Fhosc/4	0.064	0.25	0.256	1
1	110b	Fhosc/2	0.032	0.125	0.128	0.5
1	111b	Fhosc/1	0.016	0.0625	0.064	0.25

3.3.3.3 TC1M 模式寄存器

模式寄存器 TC1M 控制 TC1 的工作模式，包括 TC1 分频、时钟源和 PWM 功能等。这些设置必须在使能 TC1 定时器之前完成。

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1ENB	TC1rate2	TC1rate1	TC1rate0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT

Bit<7>:TC1ENB-TC1 启动控制位

0:关闭

1:打开

Bit<6:4>:TC1rate<2:0>:TC1 分频选择位

TC1rate<2:0>	TC1X8=0	TC1X8=1
000	Fcpu/256	Fcpu/128
001	Fcpu/128	Fcpu/64
010	Fcpu/64	Fcpu/32
011	Fcpu/32	Fcpu/16
100	Fcpu/16	Fcpu/8
101	Fcpu/8	Fcpu/4
110	Fcpu/4	Fcpu/2
111	Fcpu/2	Fcpu/1

Bit<3>: **TC1CKS**-TC1 时钟信号控制位

- 0: 内部时钟 (Fcpu 或 Fosc)
- 1: 外部时钟, 由 P0.1/INT1 输入

Bit<2>: **ALOAD1**-自动装载控制位 仅当 PWM1OUT=0 时有效

- 0: 禁止 TC1 自动重装
- 1: 允许 TC1 自动重装

Bit<1>: **TC1OUT**-TC0 溢出信号输出控制位 仅当 PWM1OUT=0 时有效

- 0: 禁止, P5.3 作为输入/输出口
- 1: 允许, P5.3 输出 TC0OUT 信号

Bit<0>: **PWM1OUT**-PWM 输出控制

- 0: 禁止 PWM 输出
- 1: 使能 PWM 输出, PWM 输出占空比由 T0OUT 和 ALOAD0 控制

注: 若 TC1CKS=1, 则 TC1 用作外部事件计数器, 此时不需要考虑 TC1RATE 的设置, P0.1 口无中断信号 (P01IRQ=0)

3.3.3.4 TC1X8/TC1GN 标志

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TC0X8	TC0GN	--

Bit<3>: **TC1X8**-TC1 内部时钟选择控制位

- 0: TC1 内部时钟来自 Fcpu, $TC1RATE = Fcpu/2^{\sim}Fcpu/256$
- 1: TC1 内部时钟来自 Fosc, $TC1RATE = Fosc/1^{\sim}Fosc/128$

注: TC1CKS=1 时, TC1X8 和 TC1RATE 可以忽略不计

3.3.3.5 TC1C 计数寄存器

8 位计数器 TC1C 溢出时, TC1IRQ 置 1 并由程序清零, 用来控制 TC1 的中断间隔时间。首先须写入正确的值到 TC1C 和 TC1R 寄存器, 并使能 TC1 定时器以保证第一个周期正确。TC1 溢出后, TC1R 的值自动装入 TC1C。

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0

TC1C 初始值的计算公式: $TC1C \text{ 初始值} = N - (TC1 \text{ 中断间隔时间} * \text{输入时钟})$

N 为 TC1 二进制计数范围。各模式下参数的设定如下表所示:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1C 有效值
0	0 (Fcpu/2 [~] Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1 [~] Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.3.3.6 TC1R 自动装载寄存器

TC1 内置自动重装功能，TC1R 寄存器存储重装数据。当 TC1C 溢出时，TC1R 的值自动装入 TC1C 中。TC1 定时器工作在计时模式时，要通过修改 TC1R 寄存器来修改 TC1 的间隔时间，而不是通过修改 TC1C 寄存器。在 TC1 定时器溢出后，新的 TC1C 值会被更新，TC1R 会将新的值装载到 TC1C 寄存器中。但在初次设置 TC1M 时，必须要在开启 TC1 定时器前把 TC1C 以及 TC1R 设置成相同的值。

TC1 为双重缓存器结构。若程序对 TC1R 进行了修改，那么修改后的 TC1R 值首先被暂存在 TC1R 的第一个缓存器中，TC1 溢出后，TC1R 的新值就会被存入 TC1R 缓存器中，从而避免 TC1 中断时间出错以及 PWM 误动作。

注：在 PWM 模式下，系统自动开启重装功能，ALOAD1 用于控制溢出范围。

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0

TC1R 初始值计算公式： $TC1R \text{ 初始值} = N - (TC1 \text{ 中断间隔时间} * \text{输入时钟})$

N 是 TC1 最大溢出值。TC1 的溢出时间和有效值见下表：

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1R 有效值
0	0 ($F_{cpu}/2^{\sim}$ $F_{cpu}/256$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 ($F_{osc}/1^{\sim}$ $F_{osc}/128$)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

例：TC0 中断间隔时间设置为 10ms，时钟源选 Fcpu (TC0KS=0, TC0X8 = 0)，无 PWM 输出 (PWM0=0)，高速时钟为外部 4MHz， $F_{cpu}=F_{osc}/4$ ，TCORATE=010 ($F_{cpu}/64$)。

$$\begin{aligned}
 TCOR &= N - (TC0 \text{ 中断间隔时间} * \text{输入时钟}) \\
 &= 256 - (10ms * 4MHz / 4 / 64) \\
 &= 256 - (0.01 * 4000000 / 4 / 64) \\
 &= 100 = 64H
 \end{aligned}$$

3.3.3.7 TC1 事件计数器

TC1 作为外部事件计数器时，其时钟源由外部输入引脚（P0.1）提供。当 TC1CKS=1 时，TC1 的时钟源由外部输入引脚（P0.1）提供，下降沿触发，下降沿触发时，TC1C 开始计数。TC1C 溢出（从 FFH 到 00H）时，TC1 触发事件计数器溢出。使能外部事件计数功能，同时外部输入引脚的唤醒功能被禁止以避免外部事件的触发信号将系统唤醒而耗电。此时，P0.1 的外部中断功能也被禁止，即 P01IRQ=0。外部事件计数器通常用来测量外部连续信号的比率，如连续的脉冲信号，R/C 振荡信号等，外部信号的相位与 MCU 时钟的相位并不同步，通过 TC1 事件计数器的测量和计算以达到不同的应用。

3.3.3.8 TC1 时钟频率输出

Buzzer 输出（TC1OUT）为定时/计数器 TC1 频率输出功能，通过设置 TC1 时钟频率，时钟信号输出到 P5.3，此时自动禁止 P5.3 的普通 I/O 功能。TC1 间隔时间 2 分频后作为 TC1OUT 频率。通过 TC1 时钟可以获得不同的频率。

TC1 溢出后，Buzzer 输出时，TC1IRQ 有效，且当 TC1IEN=1 时，使能 TC1 中断功能。但强烈建议小心同时使用 Buzzer 和 TC1 定时器，以确保两种功能都能正常工作。

Buzzer 输出引脚与 GPIO 引脚共用，TC1OUT=1 时，该引脚自动设为 Buzzer 输出引脚。如清 TC1OUT 位以禁止 Buzzer 输出后，该引脚自动返回到最后一个 GPIO 模式。

例如：设置 TC1 输出到 TC1OUT（P5.3）。外部高速时钟 Fhosc 为 4MHz，指令周期 Fcpu=Fhosc/4，TC1OUT 频率为 0.5KHz。TC0OUT 信号为 2 分频，TC1 的时钟频率为 1KHz。TC1 的时钟源为外部振荡时钟，TC1rate=Fcpu/8，

TC1RATE2~TC1RATE0=101, TC1C=TC1R=131。

MOV A, #01010000B

BOMOV TC1M, A ;TC1 速率 = Fcpu/8。

MOV A, #131 ;自动装载参考值设置。

BOMOV TC1C, A

BOMOV TC1R, A

BOBSET FTC1OUT ;TC1 的输出信号由 P5.3 输出，禁止 P5.3 的普通 I/O 功能。

BOBSET FALIADO ;使能 TC1 自动重装功能。

BOBSET FTC1ENB ;开启 TC1 定时器。

注：蜂鸣器的输出有效时，“PWM1OUT”必须被置为“0”。

3.3.3.9 脉冲宽度调制 PWM

可编程控制占空比/周期的 PWM 可以提供不同的 PWM 信号。使能 TC1 定时器且 PWM1OUT=1 时，由 PWM 输出引脚（P5.3）输出 PWM 信号。PWM 首先输出高电平，然后输出低电平。TC1RATE、ALOAD1 和 TC1OUT 位控制 PWM 的周期，TC1R 控制 PWM 的占空比（脉冲高电平的长度）。使能 TC1 定时器时，设置 TC1C 的初始值为 0。当 TC1C=TC1R 时，PWM 输出低电平；TC1 溢出时（TC1C 的值从 0FFH 到 00H），整个 PWM 周期完成，并进入下一个

周期。在 PWM 输出的过程由程序更改 PWM 的周期，则在下一个周期开始输出新的占空比的 PWM 信号。

PWM 内置 4 种可编程控制的分辨率（1/256、1/64、1/32、1/16），在 PWM1OUT = 1 时由 ALOAD1 和 TC1OUT 位控制。

PWM1	ALOAD1	TC1OUT	PWM 分辨率	TC1R 有效值	TC1R 值（二进制）
1	0	0	256	00H~0FFH	00000000B~11111111B
1	0	1	64	00H~3FH	xx000000B~xx111111B
1	1	0	32	00H~1FH	xxx00000B~xxx11111B
1	1	1	16	00H~0FH	xxxx0000B~xxxx1111B

PWM 的高电平宽度（PWM 占空比）由 TC1R 控制。TC1C = TC1R 时，PWM 输出低电平。PWM 输出过程中，TC1 溢出时，TC1IRQ 有效，TC1IEN=1 时，即使能 TC1 中断时，PWM 模式下的 TC1 中断间隔时间与 PWM 的周期相等，即 TC1 中断区域有 4 种不同的分辨率和 ALOAD1、TC1OUT 值。但强烈建议小心同时使用 PWM 和 TC1 定时器功能，保证两种功能都能正常工作。

PWM 的输出引脚与 GPIO 共用，PWM1OUT=1 时，自动输出 PWM 信号；PWM1OUT=0，即禁止 PWM 时，该引脚自动返回到上一个 GPIO 模式。这样有利于处理 ON/OFF 操作的载波信号，而不控制 TC1ENB 位。

3.4 ADC 数模转换功能模块

3.4.1 概述

模拟数字转换 (ADC) 是一个 SAR 结构, 内置 6 个模拟通道, 高达 4096 阶的分辨率, 能将一个模拟信号转换成相应的 12 位数字信号。通过 CHS[2:0] 现在模拟信号输入引脚 (AIN 引脚), 内部 $1/4 \times V_{dd}$ 电压源, GCHS 位使能全部 ADC 通道, 模拟信号输入至 SAR ADC。ADC 的分辨率为 12 位; 可以通过 ADCKS[1:0] 位选择 ADC 的转换速率以决定 ADC 的转换时间。ADC 参考电压的高电平包括 2 种, 内部参考源, 包括 V_{dd}、4V、3V、2V (EBHENB=0), 外部参考源, 由 P4.0 提供 (EVHENB=1)。ADC 内置 P4CON 寄存器来设置模拟输入引脚, 必须由程序将 ADC 输入引脚设为不带上拉电阻的输入引脚。设置好 ADENB 和 ADS 位后, ADC 开始转换, 转换结束时, ADC 电路将 EOC 和 ADCIRQ 置 1, 并将转换结果存入 ADB 和 ADR 寄存器中。若 ADCIEN=1, ADC 请求中断, AD 转换完成后, ADCIRQ=1 时, 程序计数器跳转中断向量地址 (ORG 0008H) 执行中断服务程序。

注意:

- (1) 设置 ADC 输入引脚为不带上拉电阻的输入模式;
- (2) 进入睡眠模式前禁止 ADC (ADENB=0) 以省电;
- (3) 睡眠模式下设置 P4CON 寄存器的相关位以避免额外的功耗;
- (4) 使能 ADC 后 (ADENB=1) 延时 100us 以等待 ADC 电路稳定。

3.4.2 ADC 模式数据寄存器

ADC 模式寄存器 ADM 设置 ADC 的相关配置: 包括 ADC 启动, ADC 通道选择, ADC 的参考源选择和 ADC 处理状态显示等。必须在 AD 开始转换前将这些配置设置完毕。

0B1H (ADM 控制寄存器)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADENB	ADS	EOC	GCHS	--	CHS2	CHS1	CHS0

Bit<7>:ADENB-ADC 使能控制位 (睡眠模式下, 禁止 ADC 省电)

0:禁止

1:使能

Bit<6>:ADS-ADC 启动控制位 (ADC 处理完成, ADS 位自动清零)

0:停止

1:开始

Bit<5>:EOC-ADC 状态控制位

0:转换进行中

1:转换结束, ADS 复位

Bit<4>:GCHS-ADC 通道选择位

0:禁止 AIN 通道

1:使能 AIN 通道

Bit<2:0>: CHS<2:0>-ADC 输入通道选择位

CHS<2:0>	通道编号	对应引脚
000	AIN0	P40
001	AIN1	P41
010	AIN2	P42
011	AIN3	P43
100	AIN4	P44
101	AIN5	内部 $1/4V_{DD}$ 通道

AIN5 是内部 1/4VDD 输入通道，外部没有输入引脚。AIN5 可以作为电池系统的电池检测。为了选择合适的内部 Vref 电平并进行比较，系统配置了高性能/廉价的电池检测通道。

0B2H (ADB 数据寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

Bit<7:0>:ADB<7:0>-ADC 转换数据高 8 位

ADC 数据缓存器共 12 位，用来存储 AD 转换结果，8 位只读寄存器 ADB 存放结果的高字节 (bit4~bit11)，ADR (ADR[3:0]) 存放低字节 (bit0~bit3)。ADC 数据缓存器是只读寄存器，系统复位后处于未知状态。

0B3H (ADR 控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0

Bit<6, 4>:ADCKS<1:0>-ADC 时钟源选择位

ADCKS1	ADCKS0	ADC 时钟源
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

Bit<3:0>:ADB<3:0>-ADC 转换数据低 4 位

3.4.3 ADC 引脚配置寄存器

ADC 输入引脚与 P4 口共用，ADC 输入通道的选择由 ADCHS[2:0]控制，ADCCHS[2:0]=000 时选择 AIN0，ADCCHS[2:0]=001 时选择 AIN1 等，同一时间设置 P4 口的一个引脚作为 ADC 的输入引脚，该引脚必须设置为输入引脚，禁止内部上拉，并首先由程序使能 P4CON 寄存器。通过 ADCHS[2:0]选择好 ADC 输入通道后，GCHS 置 1 以使能 ADC 功能。

注:ADC 输入引脚为 GPIO 引脚时必须设为输入模式。必须禁止 ADC 输入引脚的内部上拉电阻。ADC 输入通道的 P4CON 位必须置 1。

EVHENB = 1 时，P4.0/AIN0 为 ADC 外部参考源的输入引脚，此时，P4.0 必须设为输入模式，并禁止其上拉电阻。

注:ADC 外部参考源输入引脚为 GPIO 引脚时必须设为输入模式。必须禁止 ADC 外部参考源输入引脚的内部上拉电阻。

ADC 输入引脚与普通 I/O 引脚共用。当输入一个模拟信号到 CMOS 结构端口时，尤其当模拟信号为 1/2 VDD 时，可能产生额外的漏电流。当 P4 输入多个模拟信号时，也会产生额外的漏电流。睡眠模式下，上述漏电流会严重影响到系统的整体功耗。P4CON 为 P4 口的配置寄存器，将 P4CON[4:0]置 1，其对应的 P4 引脚将被设为纯模拟信号输入引脚，从而避免上述漏电流的产生。

0AEH (P4CON 端口配置控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0

Bit<4:0>:P4CON<4:0>-P4 ADC 端口配置控制位

0:P4 端口可以作为模拟输入 (ADC 输入) 引脚或者 GPIO 引脚

1:P4 端口只能作为模拟输入引脚，不能作为 GPIO 引脚

3.4.4 ADC 参考电压寄存器

ADC 内置 5 种参考电压，由 VREFH 寄存器控制：包括 1 个外部参考电压和 4 个内部参考源（VDD、4V、3V、2V）。EVHENB = 1 时，ADC 参考电压由外部参考源提供（P4.0），必须输入一个电压作为 ADC 参考电压的高电平，且不能低于 2V。EVHENB = 0 时，ADC 参考电压由内部参考源提供，并由 VHS[1:0] 选择控制。VHS[1:0] = 11 时，ADC 参考源选择 VDD；VHS[1:0] = 10 时，ADC 参考源选择 4V；VHS[1:0] = 01 时，ADC 参考源选择 3V；VHS[1:0] = 00 时，ADC 参考源选择 2V。外部参考源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。

0AFH（VREFH 基准电压控制寄存器）

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EVHENB	-	-	-	-	-	VHS1	VHS0

Bit<7>:EVHENB-ADC 内部参考电压控制位

0:允许 ADC 内部 Vrefh 功能，Vrefh 引脚是 P4.0/AIN0 引脚

1:禁止 ADC 内部 Vrefh 功能，P4.0/AIN0/Vrefh 引脚来自外部 Vrefh 输入引脚

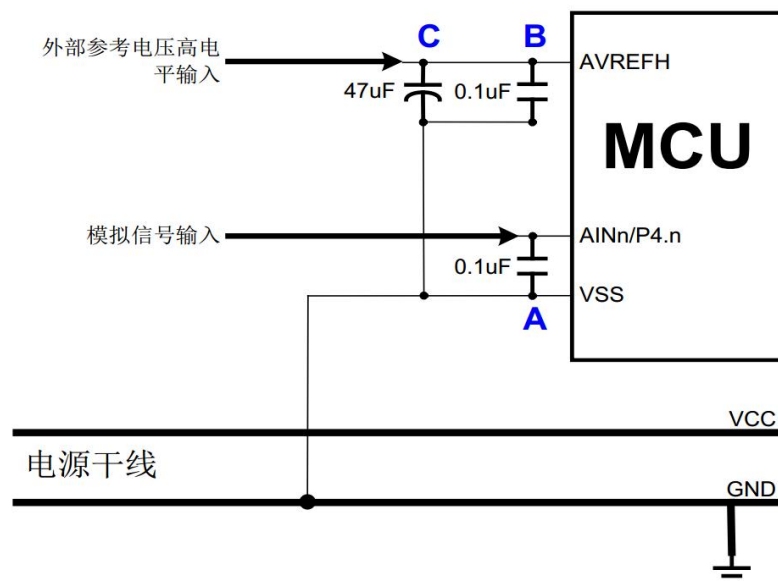
Bit<1:0>:VHS<1:0>-AD 内部参考电压选择位

VHS1	VHS0	内部 Vrefh 电压
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	VDD

3.4.5 ADC 数模转换应用电路说明

模拟信号从 ADC 输入引脚输入。在 ADC 输入引脚和芯片的 GND 引脚之间必须连接一个 0.1uF 的电容器，且要尽可能的靠近 ADC 输入引脚。不能将电容器的 GND 直接连接到电源干线上的 GND，必须通过芯片的 GND 引脚。该电容器可以减少电源干扰对模拟信号的影响。

ADC 参考电压高电平由外部参考源提供，外部参考源连接到 AVREFH 引脚。在 AVREFH 引脚和芯片的 GND 引脚之间连接电容器，首先在连接一个 47uF 的电解电容器，再连接一个 0.1uF 的电容器，且要尽可能的靠近 AVREFH 引脚。不能将电容器的 GND 直接连接到电源干线上的 GND，必须通过芯片的 GND 引脚。如下图



3.4.6 ADC 模数转换设置说明

1. 复位 ADC

CLR ADM ; 清 ADM 寄存器。

2. 设置 ADC 时钟 Rate 和 ADC 分辨率

MOV A, #0nmn0000b ; nn DCKS[1:0]代表 ADC 时钟 Rate。

BOMOV ADR, A ; m 代表 ADC 分辨率。

3. 设置 ADC 参考电压高电平源

BOBSET FEVHENB ; 外部参考源。

Or

MOV A, #000000nnb ; 内部 VDD。

; “nn” 选择内部参考源的电平。

; 11 = VDD, 10 = 4V, 01 = 3V, 00 = 2V。

4. 设置 ADC 输入通道

MOV A, #value1 ; 设置 P4CON 选择 ADC 输入通道。

BOMOV P4CON, A

MOV A, #value2 ; 设置 ADC 输入通道为输入模式。

BOMOV P4M, A

MOV A, #value3 ; 禁止 ADC 输入通道的内部上拉电阻。

BOMOV P4UR, A

5. 使能 ADC

BOBSET FADCENB

6. 执行 ADC 100us 启动时间延迟循环

CALL 100usDLY ; 100us 延迟循环

7. 选择 ADC 输入通道

MOV A, #value ; 设置 ADCHS[2:0]选择 ADC 输入通道

OR ADM, A

8. 使能 ADC 输入通道

BOBSET FGCHS

9. 使能 ADC 中断功能

BOBCLR FADCIRQ ; 清 ADC 中断请求。

BOBSET FADCIEN ; 使能 ADC 中断功能。

10. 开始 AD 转换

BOBSET FADS

注:

1、使能 ADENB 后(不是使能 ADS), 系统必须延迟 100us 等待启动 ADC, 然后设置 ADS 开始 AD 转换, 否则 ADC 的结果出错系统正常运行时, 设置 ADENB 一次, 延时一次。

2、睡眠模式和绿色模式下禁止 ADC 以省电。

3.5 LVD 电压检测功能模块

电压检测 (LVD) 是 YF2712J 内置的掉电复位保护装置, 当 VDD 跌落并低于 LVD 检测电压值时, LVD 被触发, 系统复位。不同的单片机有不同的 LVD 检测电平, LVD 检测电平值仅为一个电压点, 并不能覆盖所有死区范围。因此采用 LVD 依赖于系统要求和环境状况。如果电源跌落剧烈, 远低于 LVD 触发点, LVD 能够起到保护作用, 让系统正常复位; 如果电源电压跌落不是很剧烈, 仅仅是接近 LVD 触发点而造成的系统错误, 则 LVD 就不能起到保护作用让系统复位。

LVD 设计为三层结构 (2.0V/2.4V/3.6V), 由 LVD 编译选项控制。对于上电复位和掉电复位, 2.0V LVD 始终处于使能状态; 2.4V LVD 具有 LVD 复位功能, 并能通过标志位显示 VDD 状态; 3.6V LVD 具有标记功能, 可显示 VDD 的工作状态。LVD 标志功能只是一个低电压检测装置, 标志位 LVD24 和 LVD36 给出 VDD 的电压情况。对于低电压检测应用, 只需查看 LVD24 和 LVD36 的状态即可检测电池状况。

3.5.1 LVD 电压检测寄存器说明

086H (PFLAG 控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NT0	NPD	LVD36	LVD24	-	C	DC	Z

Bit<5>:LVD36-3.6VLVD 工作电压标志位, LVD 编译选项为 LVD_H 时有效

0:系统工作电压 VDD 超过 3.6V, 低电压检测器没有工作

1:系统工作电压 VDD 低于 3.6V, 说明此时低电压检测器已处于监控状态

Bit<4>:LVD36-2.4VLVD 工作电压标志位, LVD 编译选项为 LVD_H 时有效

0:系统工作电压 VDD 超过 2.4V, 低电压检测器没有工作

1:系统工作电压 VDD 低于 2.4V, 说明此时低电压检测器已处于监控状态

LVD	LVD 编译选项		
	LVD_L	LVD_M	LVD_H
2.0V 复位	有效	有效	有效
2.4V 标志	-	有效	-
2.4V 复位	-	-	有效
3.6V 标志	-	-	有效

LVD_L: 如果 $VDD < 2.0V$, 系统复位;

LVD24 和 LVD36 标志位无意义。

LVD_M: 如果 $VDD < 2.0V$, 系统复位;

LVD24:如果 $VDD > 2.4V$, $LVD24 = 0$; 如果 $VDD \leq 2.4V$, $LVD24 = 1$;

LVD36 标志位无意义。

LVD_H: 如果 $VDD < 2.4V$, 系统复位;

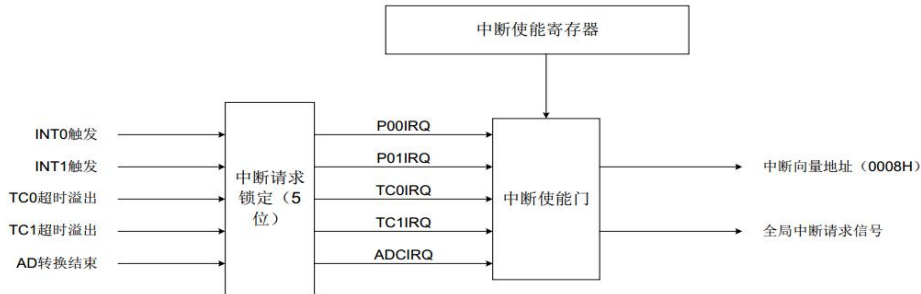
LVD36:如果 $VDD > 3.6V$, $LVD36 = 0$; 如果 $VDD \leq 3.6V$, $LVD36 = 1$;

注: (1) LVD 复位结束后, LVD24 和 LVD36 都将被清零;

(2)LVD2.4V 和 LVD3.6V 检测电平值仅作为设计参考, 不能用作芯片工作电压值的精确检测。

3.6 中断功能模块

YF2712J 具有 5 个中断源, 3 个内部中断 (TC0/TC1/ADC) 和 2 个外部中断 (INT0/INT1)。外部中断可以将系统从睡眠模式中唤醒进入高速模式, 在返回到高速模式前, 中断请求被锁定。一旦程序进入中断, 寄存器 STKP 的位 GIE 被硬件自动清零以避免响应其它中断。系统退出中断后, 硬件自动将 GIE 置“1”, 以响应下一个中断。中断请求存放在寄存器 INTRQ 中。



3.6.1 中断请求使能寄存器

中断请求控制寄存器 INTEN 包括所有中断的使能控制位。INTEN 的有效位被置为“1”则系统进入该中断服务程序, 程序计数器入栈, 程序转至 0008H 即中断程序。程序运行到指令 RETI 时, 中断结束, 系统退出中断服务。

0C9H (INTEN 控制寄存器)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIEN	TC1IEN	TC0IEN	-	-	-	P01IEN	P00IEN

Bit<7>:ADCIEN-ADC 中断控制位

0:禁止 (默认)

1:使能

Bit<6>:TC1IEN-TC1 中断控制位

0:禁止 (默认)

1:使能

Bit<5>:TC0IEN-TC0 中断控制位

0:禁止 (默认)

1:使能

Bit<1>:P01IEN-P01 外部中断控制位

0:禁止 (默认)

1:使能

Bit<0>:P00IEN-P00 外部中断控制位

0:禁止 (默认)

1:使能

3.6.2 中断请求寄存器

中断请求寄存器 INTRQ 中存放各中断请求标志。一旦有中断请求发生，则 INTRQ 中对应位将被置“1”，该请求被响应后，程序应将该标志位清零。根据 INTRQ 的状态，程序判断是否有中断发生，并执行相应的中断服务。

0C8H (INTRQ 控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIRQ	TC1IRQ	TC0IRQ	-	-	-	P01IRQ	P00IRQ

Bit<7>:ADCIRQ-ADC 中断标志位

0:禁止 (默认)

1:使能

Bit<6>:TC1IEN-TC1 中断标志位

0:禁止 (默认)

1:使能

Bit<5>:TC0IEN-TC0 中断标志位

0:禁止 (默认)

1:使能

Bit<1>:P01IEN-P01 外部中断标志位

0:禁止 (默认)

1:使能

Bit<0>:P00IEN-P00 外部中断标志位

0:禁止 (默认)

1:使能

3.6.3 GIE 全局中断

只有当全局中断控制位 GIE 置“1”的时候程序才能响应中断请求。一旦有中断发生，程序计数器 (PC) 指向中断向量地址 (ORG8)，堆栈层数加 1。

0DFH (STKP 控制寄存器)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	-	-	-	-	STKPB2	STKPB1	STKPB0

Bit<7>:GIE-全局中断控制位

0:禁止全局中断 (默认)

1:使能全局中断

注：在所有中断中，GIE 都必须处于使能状态；

3.6.4 INTO (P0.0) 中断

INT0 被触发，则无论 P00IEN 处于何种状态，P00IRQ 都会被置“1”。如果 P00IRQ=1 且 P00IEN=1，系统响应该中断；如果 P00IRQ=1 而 P00IEN=0，系统并不会执行中断服务。在处理多中断时尤其需要注意。

如果中断的触发方向和唤醒功能的触发方向是一样的，则在系统由 P0.0 从睡眠模式和绿色模式唤醒时，INT0 的中断请求（INT0IRQ）就会被锁定。系统会在唤醒后马上进入中断向量地址执行中断服务程序。

注：INT0 的中断请求被 P0.0 的唤醒触发功能锁定，P0.0 的中断触发边沿由 PEDGE 控制。

0BFH（PEDGE 控制寄存器）

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P00G1	P00G0	-	-	-

Bit<4:3>:P00G<1:0>-P00 外部中断触发控制位

- 00: 保留
- 01: 上升沿触发
- 10: 下降沿触发
- 11: 双沿触发

3.6.5 INT1（P0.1）中断

INT1 被触发，则无论 P01IEN 处于何种状态，P01IRQ 都会被置“1”。如果 P01IRQ = 1 且 P01IEN = 1，系统响应该中断；如果 P01IRQ = 1 而 P01IEN = 0，系统并不会执行中断服务。在处理多中断时尤其需要注意。

如果中断的触发方向和唤醒功能的触发方向是一样的，则在系统由 P0.1 从睡眠模式和绿色模式唤醒时，INT1 的中断请求（INT1IRQ）就会被锁定。系统会在唤醒后马上进入中断向量地址执行中断服务程序。

注：INT1 的中断请求被 P0.1 的唤醒触发功能锁定，P0.1 的中断由下降沿触发。

3.6.6 TC0 中断

TC0C 溢出时，无论 TC0IEN 处于何种状态，TC0IRQ 都会置“1”。若 TC0IEN 和 TC0IRQ 都置“1”，系统就会响应 TC0 的中断；若 TC0IEN=0，则无论 TC0IRQ 是否置“1”，系统都不会响应 TC0 中断。尤其需要注意多种中断下的情形。

TC0 中断请求设置：

```

BOBCLR FTCOEN      ; 禁止 TC0 中断。
BOBCLR FTCOENB
MOV    A, #20H
BOMOV  TCOM, A      ; TC0 时钟=Fcpu / 64。
MOV    A, # 64H     ; TC0C 初始值=64H。
BOMOV  TCOC, A      ; TC0 间隔= 10 ms。
BOBCLR FTCOIRQ     ; 清 TC0 中断请求标志。
BOBSET FTCOEN      ; 使能 TC0 中断。
BOBSET FTCOENB
BOBSET FGIE        ; 使能 GIE。

```

3.6.7 TC1 中断

TC1C 溢出时，无论 TC1IEN 处于何种状态，TC1IRQ 都会置“1”。若 TC1IEN 和 TC1IRQ 都置“1”，系统就会响应 TC1 的中断；若 TC1IEN = 0，则无论 TC1IRQ 是否置“1”，系统都不会响应 TC1 中断。尤其需要注意多种中断下的情形。

TC1 中断请求设置：

```

BOBCLR FTC1IEN     ; 禁止 TC1 中断。
BOBCLR FTC1ENB

```

```

MOV    A, #20H
BOMOV TC1M, A    ; TC1 时钟=Fcpu / 64。
MOV    A, # 64H  ; TC1C 初始值=64H。
BOMOV TC1C, A    ; TC1 间隔= 10 ms。
BOBCLR FTC1IRQ   ; 清 TC1 中断请求标志。
BOBSET FTC1IEN   ; 使能 TC1 中断。
BOBSET FTC1ENB   ; 开启 TC1 定时器。
BOBSET FGIE      ; 使能 GIE。

```

3.6.8 ADC 中断

当 ADC 转换完成后,无论 ADCIEN 是否使能,ADCIQR 都会置“1”。若 ADCIEN 和 ADCIQR 都置“1”,那么系统就会响应 ADC 中断。若 ADCIEN = 0,不管 ADCIRQ 是否置“1”,系统都不会进入 ADC 中断。用户应注意多种中断下的处理。

ADC 中断设置:

```

BOBCLR FADCIEN   ;禁止 ADC 中断。
MOV    A, #10110000B
BOMOV ADM, A     ;允许 P4.0 ADC 输入, 使能 ADC 功能
MOV    A, #00000000B ;设置 AD 转换速率 = Fcpu/16。
BOMOV ADR, A
BOBCLR FADCIRQ   ;清除 ADC 中断请求标志。
BOBSET FADCIEN   ;使能 ADC 中断。
BOBSET FGIE      ;使能 GIE
BOBSET FADS      ;开始 AD 转换

```

3.6.9 中断处理说明

中断请求控制寄存器 INTEN 包括所有中断的使能控制位。INTEN 的有效位被置为“1”则系统进入该中断服务程序,程序计数器入栈,程序转至 0008H 即中断程序。程序运行到指令 RETI 时,中断结束,系统退出中断服务。

有中断请求发生并被响应后,程序转至 0008H 执行中断子程序。响应中断之前,必须保存 ACC、PFLAG 的内容。芯片提供 PUSH 和 POP 指令进行入栈保存和出栈恢复,从而避免中断结束后可能的程序运行错误。

在同一时刻,系统中可能出现多个中断请求。此时,用户必须根据系统的要求对各中断进行优先权的设置。中断请求标志 IRQ 由中断事件触发,当 IRQ 处于有效值“1”时,系统并不一定会响应该中断。各中断触发事件如下表所示:

中断	有效触发
P00IRQ	由 PEDGE 控制
P01IRQ	下降沿触发
TC0IRQ	TC0C 溢出
TC1IRQ	TC1C 溢出
ADCIRQ	AD 转换完成

多个中断同时发生时,需要注意的是:首先,必须预先设定好各中断的优先级。其次,利用 IEN 和 IRQ 控制系统是否响应该中断。在程序中,必须对中断控制位和中断请求标志进行检测。

例：多中断条件下检测中断请求：

```
ORG 8H ;  
JMP INT_SERVICE
```

INT_SERVICE:

```
... ; 保存 ACC 和 PFLAG。
```

INTP00CHK: ; 检查是否有 P00 中断请求。

```
BOBTS1 FP00IEN ; 检查是否使能 P00 中断。  
JMP INTP01CHK ; 跳到下一个中断。  
BOBTS0 FP00IRQ ; 检查是否有 P00 中断请求。  
JMP INTP00 ; 进入 INTO 中断。
```

INTP01CHK: ; 检查是否有 P01 中断请求。

```
BOBTS1 FP01IEN ; 检查是否使能 P01 中断。  
JMP INTTC0CHK ; 跳到下一个中断。  
BOBTS0 FP01IRQ ; 检查是否有 P01 中断请求。  
JMP INTP01 ; 进入 INT1 中断。
```

INTTC0CHK: ; 检查是否有 TC0 中断请求。

```
BOBTS1 FTC0IEN ; 检查是否使能 TC0 中断。  
JMP INTTC1CHK ; 跳到下一个中断。  
BOBTS0 FTC0IRQ ; 检查是否有 TC0 中断请求。  
JMP INTTC0 ; 进入 TC0 中断。
```

INTTC1CHK: ; 检查是否有 TC1 中断请求。

```
BOBTS1 FTC1IEN ; 检查是否使能 TC1 中断。  
JMP INTADCHK ; 跳到下一个中断。  
BOBTS0 FTC1IRQ ; 检查是否有 TC1 中断请求。  
JMP INTTC1 ; 进入 TC1 中断。
```

INTADCHK: ; 检查是否有 ADC 中断请求。

```
BOBTS1 FADCIEN ; 检查是否使能 ADC 中断。  
JMP INT_EXIT ;  
BOBTS0 FADCIRQ ; 检查是否有 ADC 中断请求。  
JMP INTADC ; 进入 ADC 中断。
```

INT_EXIT:

```
... ; 恢复 ACC 和 PFLAG。  
RETI ; 退出中断。
```

3.7 复位功能模块

3.7.1 复位功能概述

YF2712J 系统提供几种复位方式：

- 1、上电复位；
- 2、看门狗复位；
- 3、掉电复位；
- 4、RESET 脚输入低电平复位（外部复位）；

上述任一种复位发生时，所有的系统寄存器恢复默认状态，程序停止运行，同时程序计数器 PC 清零。复位结束后，系统从向量 0000H 处重新开始运行。PFLAG 寄存器的 NTO 和 NPD 两个标志位能够给出系统复位状态的信息。用户可以编程控制 NTO 和 NPD，从而控制系统的运行路径。

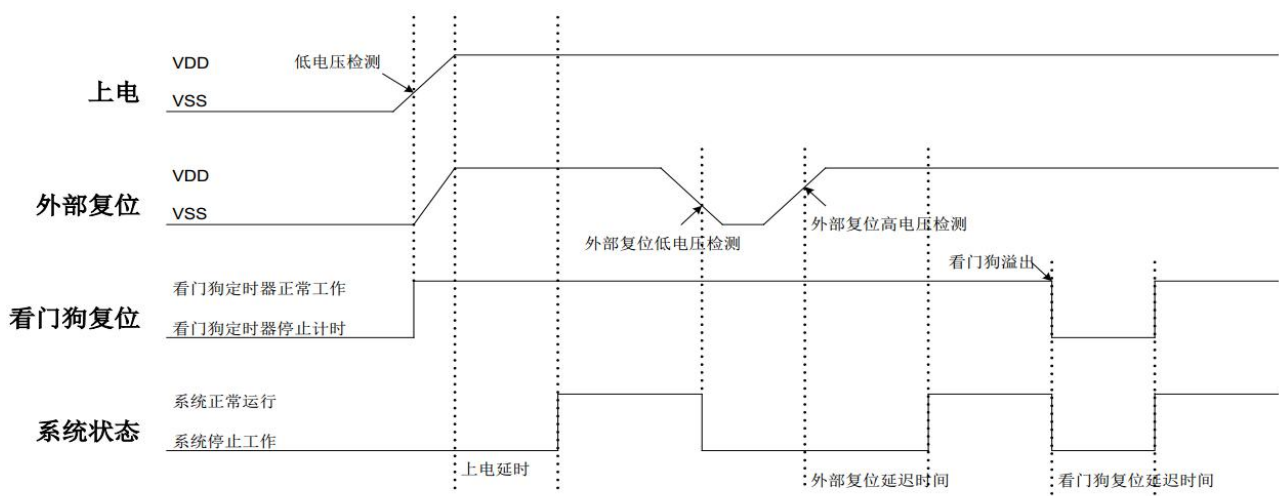
0B6H（PFLAG 控制寄存器）

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NTO	NPD	LVD36	LVD24		C	DC	Z

Bit<7:6>:NPO/NPD-复位状态标识位

NTO	NPD	复位情况	说明
0	0	看门狗复位	看门狗溢出
0	1	保留	-
1	0	上电及 LVD 复位	电源电压低于 LVD 检测值
1	1	外部复位	外部复位引脚检测到低电平

任何一种复位情况都需要一定的响应时间，系统提供完善的复位流程以保证复位动作的顺利进行。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的起振时间最短，晶体振荡器的起振时间则较长。在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。



3.7.2 上电复位

上电复位与 LVR 操作密切相关。系统上电的过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。

上电:系统检测到电源电压上升并等待其稳定;

外部复位(仅限于外部复位引脚使能状态):系统检测外部复位引脚状态。如果不为高电平,系统保持复位状态直到外部复位引脚释放;

系统初始化:所有的系统寄存器被置为初始值;

振荡器开始工作:振荡器开始提供系统时钟;

执行程序:上电结束,程序开始运行。

3.7.3 WDT 看门狗复位

看门狗复位是系统的一种保护设置。在正常状态下,由程序将看门狗定时器清零。若出错,系统处于未知状态,看门狗定时器溢出,此时系统复位。看门狗复位后,系统重启进入正常状态。

看门狗定时器状态:系统检测看门狗定时器是否溢出,若溢出,则系统复位;

系统初始化:所有的系统寄存器被置为默认状态;

振荡器开始工作:振荡器开始提供系统时钟;

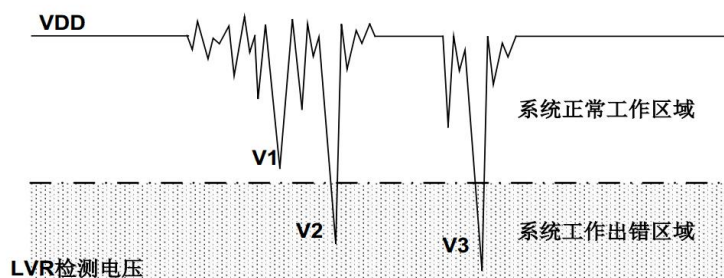
执行程序:上电结束,程序开始运行。

看门狗定时器应用注意事项:

- 1、对看门狗清零之前,检查 I/O 口的状态和 RAM 的内容可增强程序的可靠性;
- 2、不能在中断中对看门狗清零,否则无法侦测到主程序跑飞的状况;
- 3、程序中应该只在主程序中有一次清看门狗的动作,这种架构能够最大限度的发挥看门狗的保护功能。

3.7.4 掉电复位

掉电复位针对外部因素引起的系统电压跌落情形(例如,干扰或外部负载的变化),掉电复位可能会引起系统工作状态不正常或程序执行错误。



电压跌落可能会进入系统死区。系统死区意味着电源不能满足系统的最小工作电压要求。上图是一个典型的掉电复位示意图。图中,VDD 受到严重的干扰,电压值降的非常低。虚线以上区域系统正常工作,在虚线以下的区域内,系统进入未知的工作状态,这个区域称作死区。当 VDD 跌至 V1 时,系统仍处于正常状态;当 VDD 跌至 V2 和 V3 时,系统进入死区,则容易导致出错。以下情况系统可能进入死区:

DC 运用中:

DC 运用中一般都采用电池供电,当电池电压过低或单片机驱动负载时,系统电压可能跌落并进入死区。这时,电源不会进一步下降到 LVD 检测电压,因此系统维持在死区。

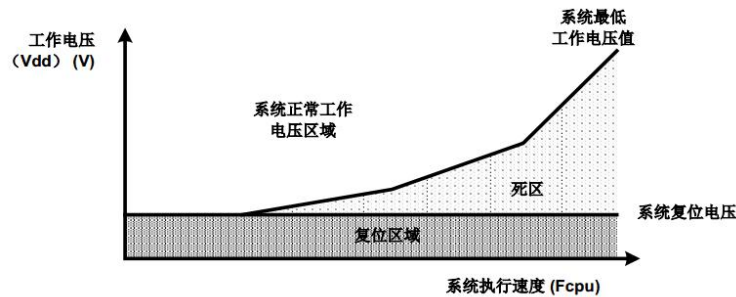
AC 运用中:

系统采用 AC 供电时, DC 电压值受 AC 电源中的噪声影响。当外部负载过高, 如驱动马达时, 负载动作产生的干扰也影响到 DC 电源。VDD 若由于受到干扰而跌落至最低工作电压以下时, 则系统将有可能进入不稳定工作状态。

在 AC 运用中, 系统上、下电时间都较长。其中, 上电时序保护使得系统正常上电, 但下电过程却和 DC 运用中情形类似, AC 电源关断后, VDD 电压在缓慢下降的过程中易进入死区。

3.7.5 工作频率与 LVR 低压复位关系

为了改善系统掉电复位的性能, 首先必须明确系统具有的最低工作电压值。系统最低工作电压与系统执行速度有关, 不同的执行速度下最低工作电压值也不同。



系统工作电压与执行速度关系图

如上图所示, 系统正常工作电压区域一般高于系统复位电压, 同时复位电压由低电压检测 (LVR) 电平决定。当系统执行速度提高时, 系统最低工作电压也相应提高, 但由于系统复位电压是固定的, 因此在系统最低工作电压与系统复位电压之间就会出现一个电压区域, 系统不能正常工作, 也不会复位, 这个区域即为死区。

为避免出现死区电压, 再选择工作频率的时候, 要选择相应的 LVR 复位电压点。如下表:

工作频率	LVR 复位电压点
Fcpu=8M	LVR=3.6V
Fcpu=4M	LVR=3.6V
Fcpu=2M 及以下	不做要求

注:

Fcpu=8M 时, YF2712J 必须 5V 供电, 复位电压必须选择 3.6V。而且只能在不高于 45 度的环境下使用。

Fcpu=4M 时, YF2712J 推荐选择复位电压 3.6V, 可全温度范围使用。最低也要选 2.4V, 但是不可在高温环境下工作。

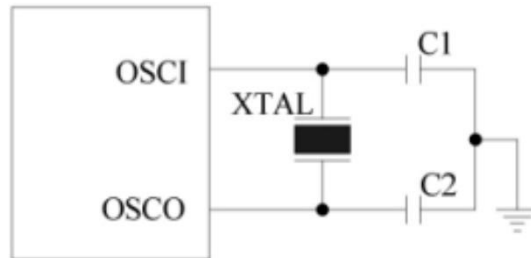
3.8 系统时钟功能模块

YF2712J 内部集成了双时钟系统，高速时钟和低速时钟。高速时钟包括内部高速时钟和外部高速时钟，由编译选项选择。低速时钟由内部低速振荡器提供，由 OSCM 寄存器的 CLKMD 位控制。双时钟都可以作为系统时钟源。

振荡器类型	说明
IHRC 内部高速振荡器	高达 32MHz
XT 外部晶振振荡器	4MHz/12MHz/32KHz
ILRC 内部低速振荡器	16KHz (3V) / 32KHz (5V)

3.8.1 外部晶体振荡器/陶瓷谐振器 (XT)

在大多数应用中，引脚 OSC0 和 OSC1 上可接晶体或陶瓷谐振器来产生振荡，电路图如下，不论是 HXT 还是 LXT 模式都适用，表中为 C1、C2 的推荐值。由于各个谐振器特性不同，用户应参照其规格选择 C1、C2 的合适值。



晶体/振荡器电路

晶体振荡器或陶瓷振荡器的电容选择参考：

振荡器模式	频率模式	频率	C1 (pF)	C2 (pF)
晶体振荡器	LXT (32.768K)	32.768 KHz	40	40
	HXT (1M~16M)	1 MHz	30	30
		2 MHz	30	30
		4 MHz	20	20
		8 MHz	30	30

注：以上数据仅供参考，一切以实物测试为准。

3.8.2 内部 RC 振荡器模式 (IRC)

内部 RC 振荡模式频率值如 32MHz, 16MHz, 8MHz, 4MHz。通过设置 OPTION 的配置位，可选择 IRC 工作频率，下面是它们的对应关系：

Firc	IRC 频率
32 M	IRC 频率选为 32MHz
16 M	IRC 频率选为 16MHz
8 M	IRC 频率选为 8MHz
4 M	IRC 频率选为 4MHz

YF2712J 提供了多种分频选择，可以在 OPTION 中选择，适用于更多的场合。如下表：

Clocks	Clocks 分频
8clock	分频为 8clock
16clock	分频为 16clock
32clock	分频为 32clock
64clock	分频为 64clock
128clock	分频为 128clock

3.8.3 OSCM 寄存器

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	CPUM1	CPUM0	CLKMD	STPHX	--

Bit<4:3>:CPUM<1:0>-CPU 工作模式控制位

CPUS1	CPUS0	工作模式
0	0	普通模式
0	1	睡眠模式
1	0	绿色模式
1	1	保留

Bit<2>:CLKMD-系统时钟模式控制位

- 0: 普通（双时钟）模式，高速时钟作为系统时钟
- 1: 低速模式，低速时钟作为系统时钟

Bit<1>:STPHX-高速振荡器控制位

- 0: 运行
- 1: 停止，内部低速 RC 振荡器仍然运行

STPHX 位为内部高速 RC 振荡器和外部高速振荡器的控制位。当 STPHX=0，内部高速 RC 振荡器和外部高速振荡器正常运行；当 STPHX=1，外部高速振荡器和内部高速 RC 振荡器停止运行。不同的高速时钟选项决定不同的 STPHX 功能。

IHRC_16M: STPHX=1，禁止内部高速 RC 振荡器；

RC, 4M, 12M, 32K: STPHX=1，禁止外部振荡器。

3.9 系统工作模式

3.9.1 概述

YF2712J 可以在 4 种工作模式下以不同的时钟频率工作，这些模式可以控制振荡器的工作、程序的执行以及模拟电路的功能损耗。

1. 普通模式：系统高速工作模式；
2. 低速模式：系统低速工作模式；
3. 省电模式：系统省电模式（睡眠模式）；
4. 绿色模式：系统理想模式。

工作模式	普通模式	低速模式	绿色模式	睡眠模式
EHOSC	运行	STPHX 控制	STPHX 控制	停止
IHRC	运行	STPHX 控制	STPHX 控制	停止
ILRC	运行	运行	运行	停止
CPU 指令	执行	执行	停止	停止
TC0	TC0ENB 控制	TC0ENB 控制	TC0ENB 控制 仅 PWM/Buzzer 有效	无效
TC1	TC1ENB 控制	TC1ENB 控制	TC1ENB 控制 仅 PWM/Buzzer 有效	无效
看门狗定时器	WDT 选项控制	WDT 选项控制	WDT 选项控制	WDT 选项控制
内部中断	全部有效	全部有效	TC0	全部无效
外部中断	全部有效	全部有效	全部有效	全部无效
唤醒功能	-	-	P0, TC0, 复位	P0, 复位

EHOSC：外部高速时钟（XIN/XOUT）。

IHRC：内部高速时钟（RC 振荡器）。

ILRC：内部低速时钟（RC 振荡器）。

3.9.2 普通模式

普通模式是系统高速时钟正常工作模式，系统时钟源由高速振荡器提供。程序被执行。上电复位或任意一种复位触发后，系统进入普通模式执行程序。当系统从睡眠模式被唤醒后进入普通模式。普通模式下，高速振荡器正常工作，功耗最大。

程序被执行，所有的功能都可控制。

系统速率为高速。

高速振荡器和内部低速 RC 振荡器都正常工作。

通过 OSCM 寄存器，系统可以从普通模式切换到其它任何一种工作模式。

系统从睡眠模式唤醒后进入普通模式。

低速模式可以切换到普通模式。

从普通模式切换到绿色模式，唤醒后返回到普通模式。

3.9.3 低速模式

低速模式为系统低速时钟正常工作模式。系统时钟源由内部低速 RC 振荡器提供。低速模式由 OSCM 寄存器的 CLKMD 位控制。当 CLKMD=0 时，系统为普通模式；当 CLKMD=1 时，系统进入低速模式。切换进入低速模式后，不能自动禁止高速振荡器，必须通过 SPTHX 位来禁止以减少功耗。低速模式下，系统速率被固定为 $F_{osc}/4$ (F_{osc} 为内部低速 RC 振荡器频率)。

程序被执行，所有的功能都可控制。

系统速率位低速 ($F_{osc}/4$)。

内部低速 RC 振荡器正常工作，高速振荡器由 SPTHX=1 控制。低速模式下，强烈建议停止高速振荡器。

通过 OSCM 寄存器，低速模式可以切换进入其它的工作模式。

从低速模式切换到睡眠模式，唤醒后返回到普通模式。

普通模式可以切换进入低速模式。

从低速模式切换到绿色模式，唤醒后返回到低速模式。

3.9.4 睡眠模式

睡眠模式是系统的理想状态，不执行程序，振荡器也停止工作。整个芯片的功耗低于 1uA。睡眠模式可以由 P0 的电平变换触发唤醒。从任何工作模式进入睡眠模式，被唤醒后都返回到普通模式。由 OSCM 寄存器的 CPUM0 位控制是否进入睡眠模式，当 CPUM0=1，系统进入睡眠模式。当系统从睡眠模式被唤醒后，CPUM0 被自动禁止 (0 状态)。

程序停止执行，所有的功能被禁止。

所有的振荡器，包括外部高速振荡器、内部高速振荡器和内部低速振荡器都停止工作。

功耗低于 1uA。

系统从睡眠模式被唤醒后进入普通模式。

睡眠模式的唤醒源为 P0 电平变换触发。

注：普通模式下，设置 SPTHX=1 禁止高速时钟振荡器，这样，无系统时钟在执行，此时系统进入睡眠模式，可以由 P0 电平变换触发唤醒。

3.9.5 绿色模式

绿色模式是另外一种理想状态。在睡眠模式下，所有的功能和硬件设备都被禁止，但在绿色模式下，系统时钟保持工作，绿色模式下的功耗大于睡眠模式下的功耗。绿色模式下，不执行程序，但具有唤醒功能的定时器仍正常工作，定时器的时钟源为仍在工作的系统时钟。绿色模式下，有 2 种方式可以将系统唤醒：1、P0 电平变换触发；2、具有唤醒功能的定时器溢出，这样，用户可以给定时器设定固定的周期，系统就在溢出时被唤醒。由 OSCM 寄存器 CPUM1 位决定是否进入绿色模式，当 CPUM1=1，系统进入绿色模式。当系统从绿色模式下被唤醒后，自动禁止 CPUM1 (0 状态)。

程序停止执行，所有的功能被禁止。

具有唤醒功能的定时器正常工作。

作为系统时钟源的振荡器正常工作，其它的振荡器工作状态取决于系统工作模式的配置。

由普通模式切换到绿色模式，被唤醒后返回到普通模式。

由低速模式切换到绿色模式，被唤醒后返回到低速模式。

绿色模式下的唤醒方式为 P0 电平变换触发唤醒和指定的定时器溢出。

绿色模式下 PWM 和 Buzzer 功能仍然有效，但是定时器溢出时不能唤醒系统。

注：提供宏“GreenMode”来控制绿色模式的工作状态，必要时使用宏“GreeMode”进绿色模式。该宏共有 3 条指令。但在使用 BRANCH 指令（如 BTS0、BTS1、BOBTS0、BOBTS1、INCS、INCMS、DECS、DECMS、CMPRS、JMP）时必须注意宏的长度，否则程序会出错。

3.9.6 控制宏

宏名称	长度	说明
SleepMode	1-word	系统进入睡眠模式。
GreenMode	3-word	系统进入绿色模式。
SlowMode	2-word	系统进入低速模式并停止高速振荡器。
Slow2Normal	5-word	系统从低速模式返回到普通模式。该宏包括工作模式的切换，使能高速振荡器，高速振荡器唤醒延迟时间。

例：从普通/低速模式切换进入绿色模式，并使能 T0 唤醒功能。

;设置定时器 T0 的唤醒功能

BOBCLR FTOIEN ; 禁止 T0 中断。

BOBCLR FTOENB ; 禁止 T0 定时器。

MOV A, #20H

BOMOV TOM, A ; 设置 T0 时钟= Fcpu / 64。

MOV A, #64H

BOMOV TOC, A ; 设置 TOC 的初始值=64H (设置 T0 间隔值 = 10 ms)。

BOBCLR FTOIEN ; 禁止 T0 中断。

BOBCLR FTOIRQ ; 清 T0 中断请求。

BOBSET FTOENB ; 使能 T0 定时器。

;进入绿色模式。

GreenMode ; 直接宣告“GreenMode”宏。

3.9.7 系统唤醒

睡眠模式和绿色模式下，系统并不执行程序。唤醒触发信号可以将系统唤醒进入普通模式或低速模式。唤醒触发信号包括：外部触发信号（P0 的电平变换）和内部触发（T0 定时器溢出）。从睡眠模式唤醒后只能进入普通模式，且将其唤醒的触发只能是外部触发信号（P0 电平变化）；如果是将系统由绿色模式唤醒返回到上一个工作模式（普通模式或低速模式），唤醒触发信号可以是外部触发信号（P0 电平变换）和内部触发信号（T0 溢出）。

系统进入睡眠模式后，高速时钟振荡器停止运行。把系统从睡眠模式唤醒时，单片机需要等待一段时间以等待振荡电路稳定工作，等待的这一段就称为唤醒时间。唤醒时间结束后，系统进入普通模式。

注：从绿色模式下唤醒系统不需要唤醒时间，因为系统时钟在绿色模式下仍然正常工作。

外部高速振荡器的唤醒时间的计算如下：

$$\text{唤醒时间} = 1/F_{\text{Hosc}} * 2048 \text{ (sec)} + \text{高速时钟启动时间}$$

例：睡眠模式下，系统被唤醒进入普通模式。唤醒时间的计算如下：

$$\text{唤醒时间} = 1/F_{\text{Hosc}} * 2048 = 0.512 \text{ ms (4 MHz)}$$

$$\text{总的唤醒时间} = 0.512 \text{ ms} + \text{振荡器启动时间}$$

内部高速 RC 振荡器的唤醒时间的计算如下：

$$\text{唤醒时间} = 1/F_{\text{Hosc}} * 8 \text{ (sec)} + \text{高速时钟启动时间}$$

例：睡眠模式下，系统被唤醒进入普通模式。唤醒时间的计算如下：

$$\text{唤醒时间} = 1/F_{\text{osc}} * 8 = 0.5 \text{ us (F}_{\text{Hosc}}=16\text{MHz)}$$

注：高速时钟的启动时间与 VDD 和振荡器类型有关。

4 CODE OPTION 寄存器

CODE OPTION	选项	功能描述
代码加密	加密	烧录模式数据加密
	不加密	烧录模式数据不加密
看门狗	使能-Enable	看门狗使能
	使能-Always On	看门狗长打开
	禁止-Disable	禁止看门狗
振荡模式	HIRC	内部高速振荡器
	HIRC-RTC	双时钟
	LXT	外接低速晶振
	HXT	外接高速晶振
HIRC 频率	4M	内部 4M 频率
	8M	内部 8M 频率
	16M	内部 16M 频率
	32M	内部 32M 频率
Clocks 分频	8 Clocks	时钟分频选择 8 Clocks
	16 Clocks	时钟分频选择 16 Clocks
	32 Clocks	时钟分频选择 32 Clocks
	64 Clocks	时钟分频选择 64 Clocks
	128 Clocks	时钟分频选择 128 Clocks
系统滤波	使能	系统滤波使能
	禁止	系统滤波禁止
复位端口上拉	使能	复位端口上拉使能
	禁止	复位端口上拉禁止
P04 端口	as GPI	P04 端口作为通用输入口
	as RST	P04 端口作为复位脚
RST 脚 复位延时	无延时	RST 脚复位无延时
	延时 128CLK	RST 脚复位延时 128CLK
P04 端口 开漏输出	使能	P04 端口开漏输出使能
	禁止	P04 端口开漏输出禁止
LVD	LVR-L	LVR:2.0V
	LVR-M	LVR:2.0V LVD:2.4V
	LVR-H	LVR:2.4V LVD:3.6V
	LVD-S	LVR:3.6V

5 芯片电气特性

5.1 芯片极限参数

工作温度(°C):	(√) E:-40-85;
存储温度(°C):	(√) -65~+150
极限电压(V)	(√) 其它 <u>-0.3~6</u>
极限输入电压 (V)	(√) 其它 <u>GND-0.3~VDD+1.0;</u>
极限输出电压 (V)	(√) 其它 <u>GND-0.3~VDD+1.0;</u>

5.2 芯片直流参数

(T=25°C, VDD=5±5%V, GND=0V)

符号	参数说明	条件	最小	典型	最大	单位
IRC1	IRC1 (校正后)	OPTION 选择 4MHz	-	4	-	MHz
IRC2	IRC2 (校正后)	OPTION 选择 8MHz	-	8	-	MHz
IRC3	IRC3 (校正后)	OPTION 选择 16MHz	-	16	-	MHz
IRC4	IRC4 (校正后)	OPTION 选择 32MHz	-	32	-	MHz
IOH1	输出高电平驱动 (除 P04)	Ioh=4.4V	16	17	18	mA
IOL1	I0 输出低电平驱动	Iol=0.6V	19	20	21	mA
IPH	上拉电流	上拉使能, 输入接地	55	58	60	μA
I _{sb1}	绿色模式功耗 1	从低速模式进入绿色模式	1.8	2	3	μA
I _{sb2}	绿色模式功耗 2	普通模式 (16M) 进入绿色模式	280	290	300	μA
I _{sb3}	绿色模式功耗 3	普通模式 (4M) 进入绿色模式	110	120	130	μA
I _{lb1}	低速模式功耗	从低速模式进入绿色模式	4	5	6	μA
ILb1	睡眠模式功耗 1	常规睡眠	-	-	1	μA
ILb2	睡眠模式功耗 2	开 LVD 睡眠	4	5	6	μA
Iop1	工作电流 1 (VDD=5V)	IRC=32MHz 8clock	-	1.6	-	mA
Iop2	工作电流 2 (VDD=5V)	IRC=16MHz 8clock	-	0.9	-	mA
Iop3	工作电流 3 (VDD=5V)	IRC=16MHz 16clock	-	0.6	-	mA
Iop4	工作电流 4 (VDD=5V)	IRC=8MHz 8clock	-	0.5	-	mA
Iop5	工作电流 5 (VDD=5V)	IRC=4MHz 8clock	-	0.3	-	mA
Iop6	工作电流 6 (VDD=5V)	XT=12MHz 8clock	-	1	-	mA
Iop7	工作电流 7 (VDD=5V)	XT=4MHz 8clock	-	0.6	-	mA
Iop8	工作电流 8 (VDD=5V)	XT=32.768KHz 2clock	-	13	-	μA
LVR	低电压复位电压	选择 LVR 复位点	V _{lvr} -0.2	V _{lvr}	V _{lvr} +0.2	V
LVD	低压检测电压	选择检测点为 V _{lvd}	V _{lvd} -0.2	V _{lvd}	V _{lvd} +0.2	V

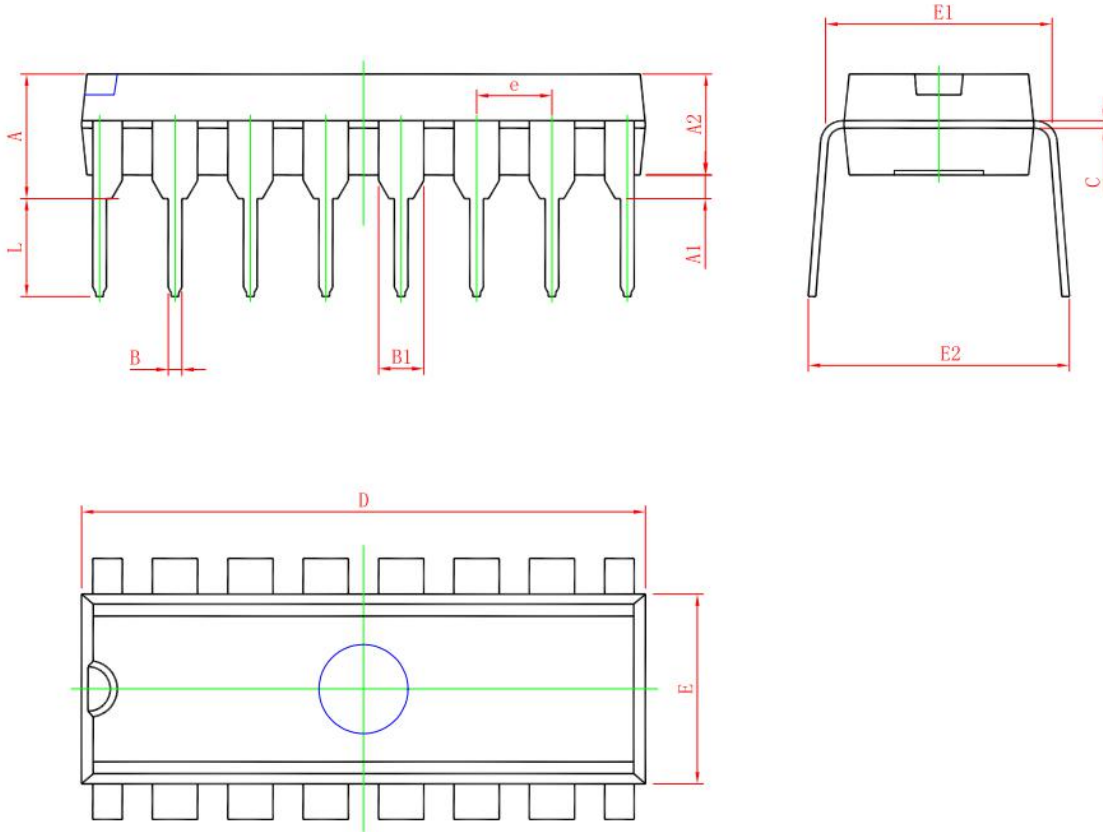
6 封装类型

OTP MCU	封装类型	引脚数	封装尺寸
YF2712JPB/PD	DIP16/SOP16	16	§ 7.1
YF2712JM/N	DIP14/SOP14	14	§ 7.2
YF2712JB/D	DIP8/SOP8	8	§ 7.3

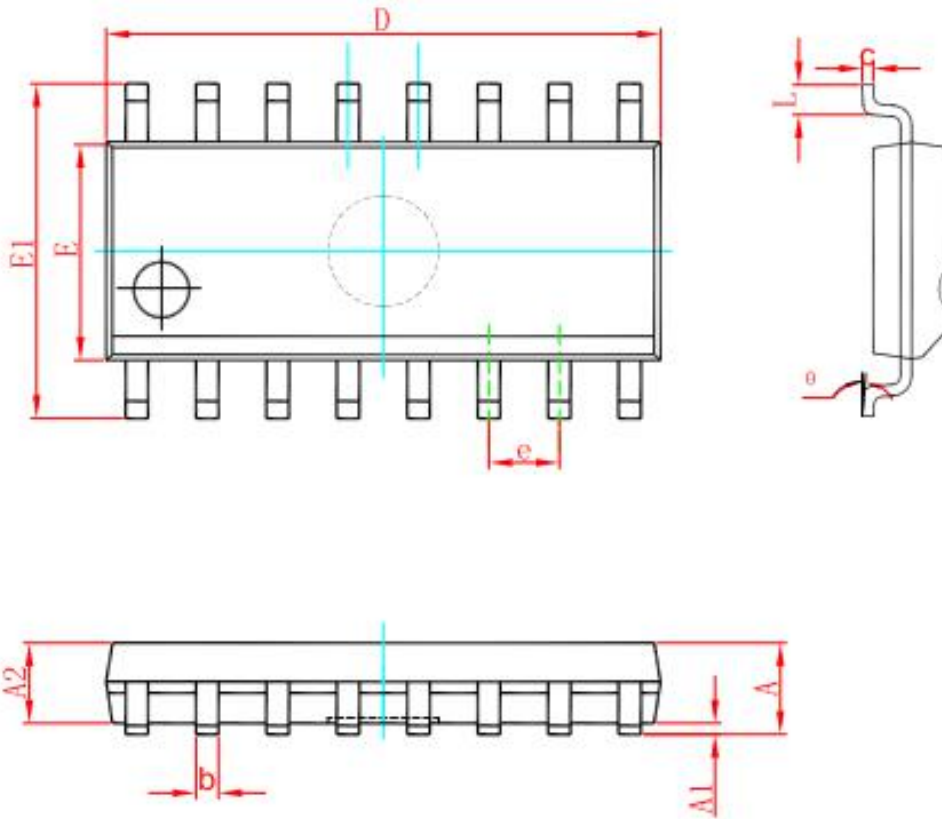
7 封装尺寸

7.1 16PIN 封装尺寸

DIP16



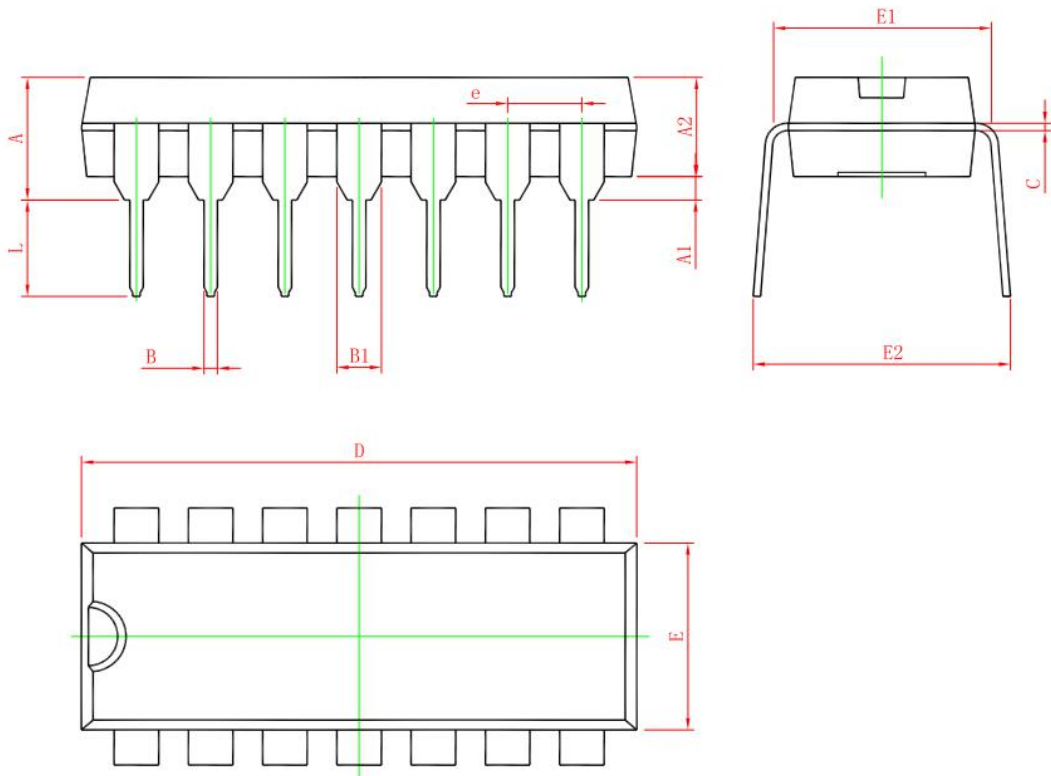
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	3.710	4.310	0.146	0.170
A1	0.510		0.020	
A2	3.200	3.600	0.126	0.142
B	0.380	0.570	0.015	0.022
B1	1.524(BSC)		0.060(BSC)	
C	0.204	0.360	0.008	0.014
D	18.800	19.200	0.740	0.756
E	6.200	6.600	0.244	0.260
E1	7.320	7.974	0.288	0.314
e	2.540(BSC)		0.100(BSC)	
L	3.000	3.600	0.118	0.142
E2	8.400	9.000	0.331	0.354



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.007	0.010
D	9.800	10.200	0.386	0.402
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

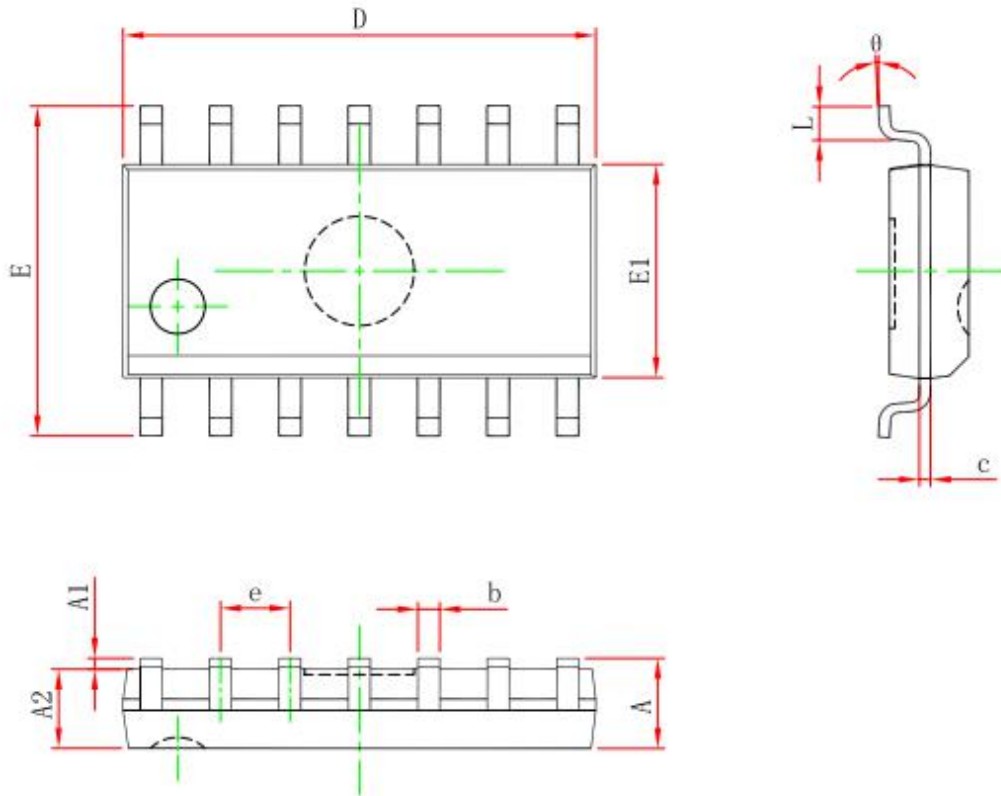
7.3 14PIN 封装尺寸

DIP14



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	3.710	4.310	0.146	0.170
A1	0.510		0.020	
A2	3.200	3.600	0.126	0.142
B	0.380	0.570	0.015	0.022
B1	1.524(BSC)		0.060(BSC)	
C	0.204	0.360	0.008	0.014
D	18.800	19.200	0.740	0.756
E	6.200	6.600	0.244	0.260
E1	7.320	7.974	0.288	0.314
e	2.540(BSC)		0.100(BSC)	
L	3.000	3.600	0.118	0.142
E2	8.400	9.000	0.331	0.354

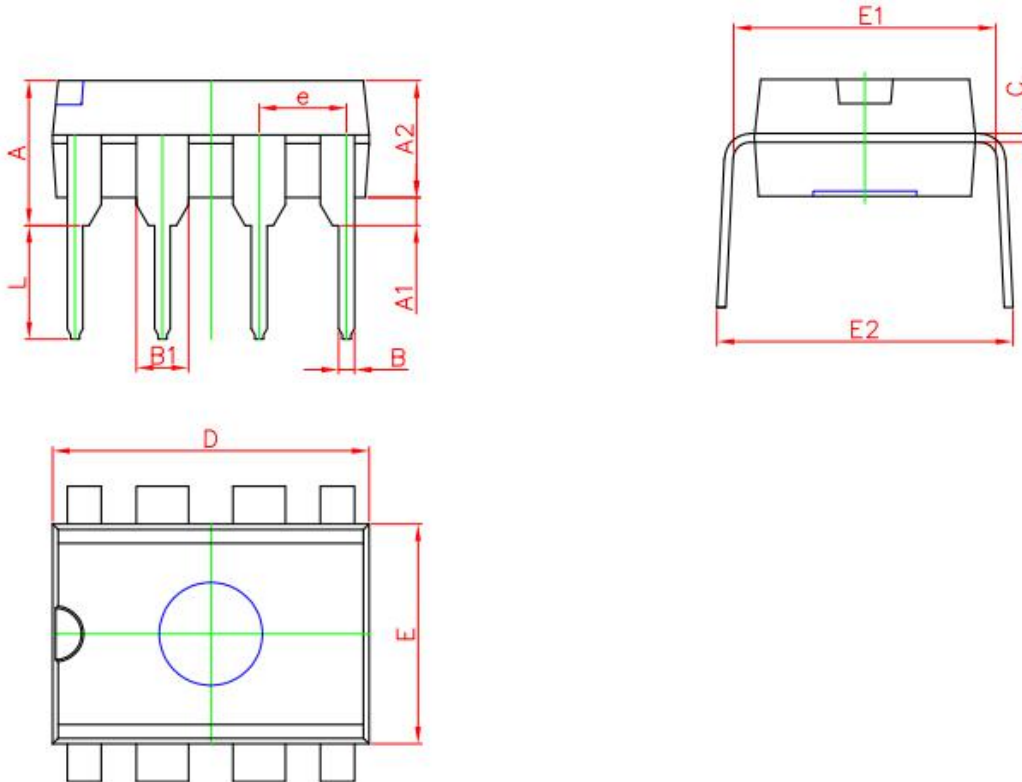
SOP14



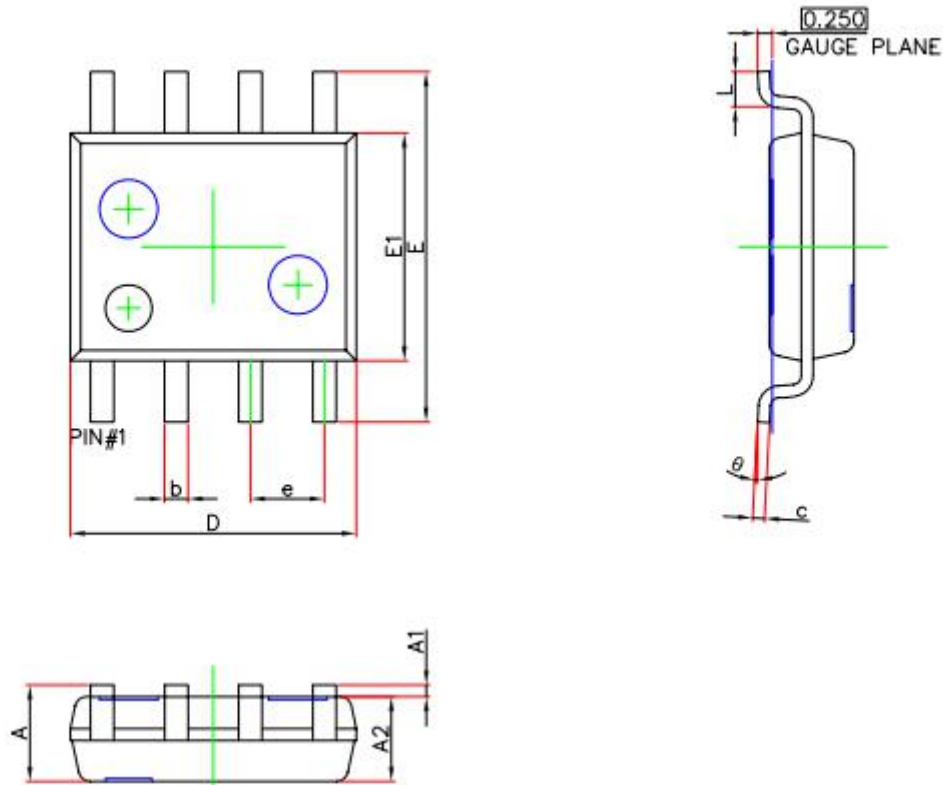
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	--	1.750	--	0.069
A1	0.100	0.250	0.004	0.010
A2	1.250	--	0.049	--
b	0.310	0.510	0.012	0.020
c	0.100	0.250	0.004	0.010
D	8.450	8.850	0.333	0.348
E	5.800	6.200	0.228	0.244
E1	3.800	4.000	0.150	0.157
e	1.270(BSC)		0.050(BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

7.3 8PIN 封装尺寸

DIP8



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	3.710	4.310	0.146	0.170
A1	0.510		0.020	
A2	3.200	3.600	0.126	0.142
B	0.380	0.570	0.015	0.022
B1	1.524(BSC)		0.060(BSC)	
C	0.204	0.360	0.008	0.014
D	9.000	9.400	0.354	0.370
E	6.200	6.600	0.244	0.260
E1	7.320	7.950	0.288	0.313
e	2.540(BSC)		0.100(BSC)	
L	3.000	3.600	0.118	0.142
E2	8.400	9.000	0.331	0.354



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	1.450	1.750	0.057	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.007	0.010
D	4.700	5.100	0.185	0.201
E	5.800	6.200	0.228	0.244
E1	3.800	4.000	0.150	0.157
e	1.270(BSC)		0.050(BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°