

目 录

1. 产品概述	3
2. 主要特性	3
3. 管脚配置	6
4. 系统结构	9
5. 中央处理器	10
5.1 指令集	10
5.2 寻址方式	14
5.3 CPU寄存器	16
6. 存储系统(ROM/RAM).....	18
6.1 RAM	18
6.2 ROM	19
6.3 OPTION	20
7. 特殊功能寄存器(SFR).....	21
7.1 SFR地址	21
7.2 扩展特殊功能寄存器地址	21
7.3 SFR汇总	22
8. 时钟系统	26
9. 工作模式	28
9.1 STOP模式	29
9.2 IDLE模式	29
10. 中断系统	30
10.1 中断结构	31
10.2 中断优先级	32
10.3 中断向量列表	32
10.4 中断 SFR	33
11. 实时时钟系统(RTC)	38
11.1 RTC功能	38
11.2 RTC SFR	39
11.3 RTC应用说明	42
11.4 RTC配置流程	42
12. 定时器(TIMER).....	44
12.1 TIMER0	44
12.1.1 定时模式	46
12.1.2 计数模式	46
12.2 TIMER1	47
12.3 TIMER2	49
12.4 PWM	51
12.4.1 PWM SFR	51
12.4.2 PWM配置流程	53
13. 通用输入输出端口(GPIO).....	54
13.1 GPIO SFR	54
13.2 IO配置说明	57

14. 端口资源映射.....	59
15. 电容触摸(CDC).....	62
16. 模数转换器(ADC)	63
16.1 ADC SFR	63
16.2 ADC配置流程	67
17. 低电压检测(LVD)	68
18. 同步串行总线(Mini I2C)	69
18.1 I2C SFR	70
18.2 I2C功能描述	73
18.3 I2C通讯过程	75
18.3.1 主机发送模式软件配置流程	75
18.3.2 主机接收模式软件配置流程	76
18.3.3 从机接收模式软件配置流程	77
18.3.4 从机发送模式软件配置流程	78
18.4 I2C时钟设置	79
19. 通用异步收发器(UART)	80
19.1 UART SFR	80
19.2 UART工作模式	83
19.3 UART波特率.....	85
20. 在线编程(IAP).....	86
20.1 IAP SFR.....	86
20.2 IAP操作流程.....	89
21. 看门狗定时器(WDT)	90
22. 复位系统(RESET).....	92
23. 程序下载和仿真	93
24. 电气特性	94
24.1 电气特性极限参数.....	94
24.2 直流特性.....	94
24.3 振荡器特性	95
24.4 ADC特性	96
25. 封装尺寸图	97
25.1 SSOP28封装	97
25.2 QFN24封装	98
26. 历史记录	99

触摸 A/D 型 8-Bit MCU

文件编号: PT-DS24005

1. 产品概述

PT5F2307 是一款 51 内核的触控 A/D 型 8 位 MCU，内置 16K*8bit FLASH、内部 256*8bit SRAM、外部 512*8bit SRAM、触控检测、12 位高精度 ADC、RTC、PWM 等功能，抗干扰能力强，适用于滑条遥控器、智能门锁、消费类电子产品等电子应用领域。

2. 主要特性

■ CPU

- CPU: 1T 8051 内核，指令兼容 MCS-51 指令集
- 程序存储器 FLASH COD_ROM: 16K*8bit，可反复擦写
- FLASH 可划分为程序空间和数据空间，实现掉电数据存储，省略 EEPROM
- 内部数据存储器 SRAM: 256*8bit
- 外部数据存储器 SRAM: 512*8bit
- 内置 4Byte UID 用于无线通讯对码
- 支持 ISP 和 IAP

■ 中断

- 16 个中断源: T0、T1、T2、RTC、INT0、INT1、ADC、PWM、UART*4、TOUCH、LVD、LPVT、I2C
- 2 级中断优先级可设，16 个中断向量入口

■ 工作电压

- 8MHz @1.8-5.5V
- 16MHz @2.7-5.5V

■ 时钟系统

- 内部 RC 高频振荡器 32MHz 精度: $\pm 1.5\%$
- 内部 RC 低频振荡器 131KHz 精度: $\pm 1.5\%$
- 外部晶振 32.768KHz
- 内嵌外部晶振时钟监控模块

■ I/O 口

- 26 个双向 I/O 端口，带 SMIT 输入，支持推挽、开漏、上拉、下拉模式设置
I/O 拉电流 8 级可设(默认 14mA): 2mA/5mA/8mA/11mA/14mA/20mA/30mA/100mA
I/O 灌电流 4 级可设(默认 60mA): 5mA/15mA/60mA/120mA
- I/O 输入电压门限 2 档可选择
 $V_{IH_0.7*VDD} / V_{IL_0.3*VDD}$
 $V_{IH_0.5*VDD} / V_{IL_0.2*VDD}$
- I/O 输出电压可软件配置为 3/5VDD 用以适配非等电位模块之间的控制

- I/O 输出电压可软件配置为 1/2VDD 用以软件模拟 LCD 驱动

■ TOUCH

- 25 路触摸通道输入，其中 P30,P31 为高穿透力触摸通道
- 支持单键及滑条触摸模式
- 支持触摸引脚与 LED 驱动引脚复用，节省芯片引脚资源
- 支持跳频功能，高抗干扰能力
- 内建 2.3V/2.0V LDO 供电可选

■ 定时器

- TIMER0
 - 自动装载 8 位定时器，支持预分频
 - 支持计数器功能
- TIMER1
 - 自动装载 16 位定时器，支持预分频
- TIMER2
 - 自动装载 16 位定时器，支持预分频

■ PWM

- 支持 6 路 PWM
- 其中 PWM0~PWM2 时钟来源 TIMER1，PWM3~PWM5 可选择时钟来源 TIMER1 或 TIMER2
- PWM 中断可设置上升沿或下降沿触发

■ RTC

- 内置 RTC 模块可计时、分、秒、星期、天数，支持闹钟功能
- 支持毫秒、半秒、闹钟中断

■ UART

- 2 路异步 UART 通讯接口，最大波特率 115200
- 支持 1 字节接收缓存

■ Mini I²C

- 1 路精简 I²C 接口，支持主从模式
- 支持标准/快速模式，最高支持 400Kbit/S

■ 资源映射

- INTx(x=0~1)可映射至所有 I/O 端口
- PWM0、PWM1 可映射至所有非 PWM 输出的 I/O 端口
- T0 可映射至 P05、P06、P07 端口
- UART1_TX 可映射至 P32 端口，UART1_RX 可映射至 P02 端口
- I2C_SDA 可映射至 P31 端口，I2C_SCL 可映射至 P30 端口

■ ADC

- 12 位 SAR ADC
- 23 路外部输入通道(ADC0~ADC22)，2 路内部特殊通道（内部 VDD/4、内部 GND）

- 3 种参考电压可选：VDD、外部参考电压 VREF、内部基准电压（1.28V、2.048V、3.072V、4.096V）
- 具备初始失调校准功能

■ 保护系统及工作模式

- 5 种系统复位方式：
 - 上电复位(POR)
 - 低压复位(LVR)
 - 看门狗(WDT)溢出复位
 - 软件复位(WRST)
 - 外部复位(RST)
- 支持 3 种工作模式
 - Normal 模式：正常工作模式
 - STOP 模式：低功耗模式，CPU 停止工作，外设停止工作
唤醒方式：外部中断、RTC 中断、触摸中断、输入 IO 电平变化、WDT 溢出、LPVT
 - IDLE 模式：仅 CPU 停止工作，其它外设可以工作
唤醒方式：所有中断、输入 IO 电平变化、WDT 溢出
- 内嵌 LVR 功能，复位阈值可选：关闭、1.8V、2.0V、2.2V、2.4V、2.7V、2.9V、3.6V
- 内嵌 LVD 功能，检测阈值可选：2.0V、2.2V、2.5V、2.7V、2.9V、3.2V、4.0V、4.3V
- 内嵌 WDT，支持预分频功能

■ 抗干扰能力

- HBM ESD: >4KV

■ 封装形式

- SSOP28: PT5F2307P28
- QFN24: PT5F2307F24

3. 管脚配置

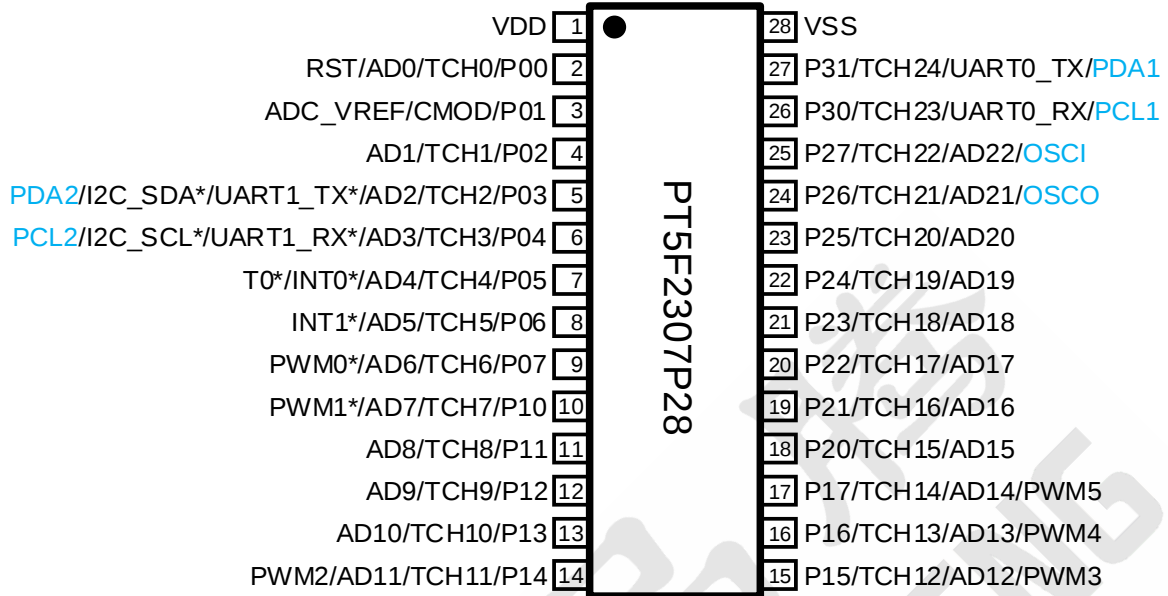


图 1 SSOP28 管脚示意图

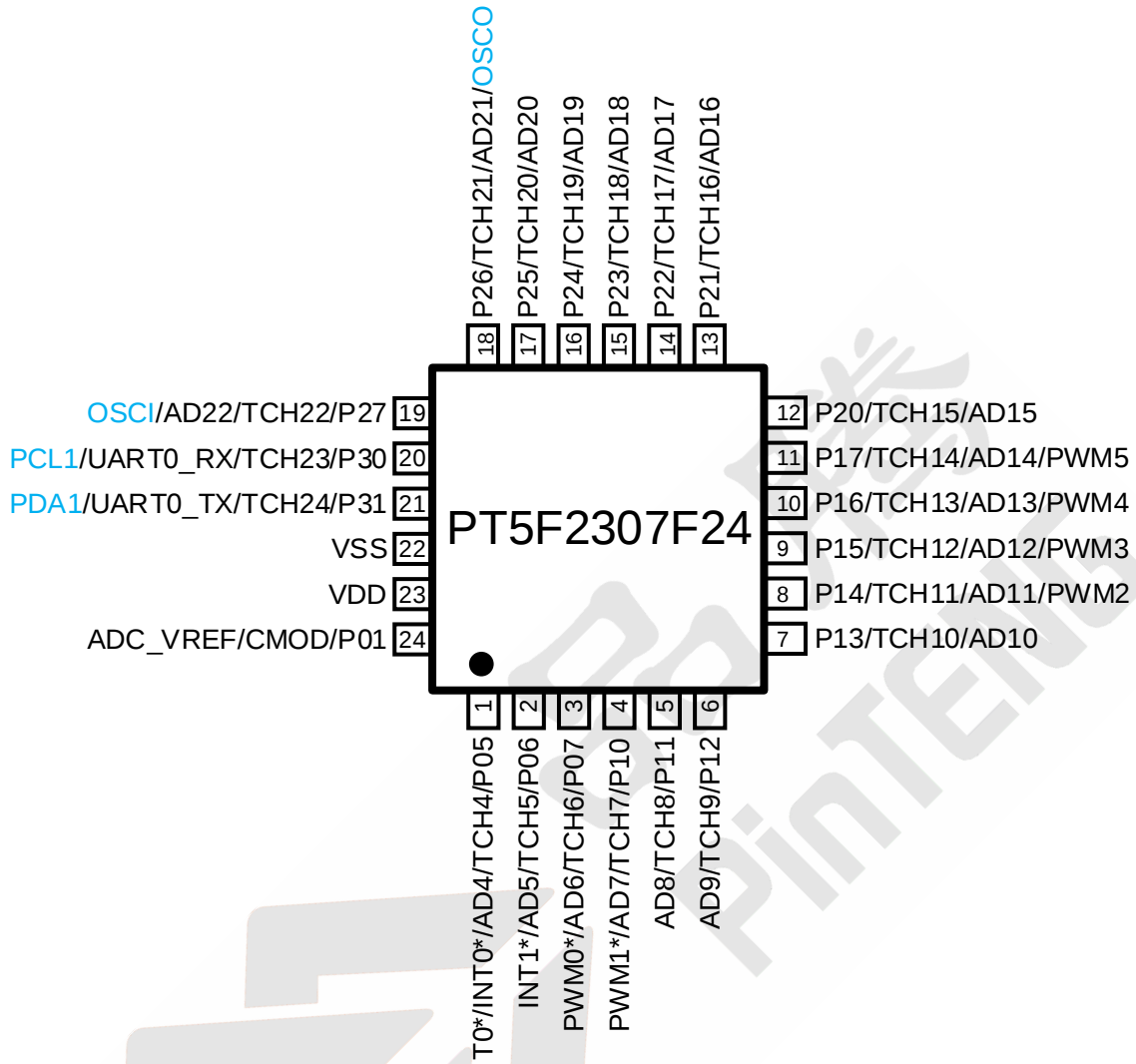


图 2 QFN24 管脚示意图

表 1 管脚信号说明

管脚名称	IO类型	管脚说明
VDD	P	电源
VSS	P	地
PCL	I	烧录时钟线
PDA	I/O	烧录数据线
ADC_VREF	I	ADC参考电压输入
OSCI/OSCO	I/O	外部32.768KHz晶振端口
Pxx	I/O	输入/输出GPIO
TCHx	I	触摸通道
ADx	I	ADC输入通道
INTx	I	外部中断输入脚
PWMx	O	PWM输出脚
Tx	I	外部时钟输入脚
UARTx_TX	O	串口数据输出
UARTx_RX	I	串口数据输入
I2C_SDA	I/O	I2C数据线
I2C_SCL	I/O	I2C时钟线

注：PIN 类型

I： 仅有 CMOS 输入

O： CMOS/NMOS 输出

P： 电源/地

* ： 资源映射可配置管脚

4. 系统结构

基于 1T 速度的 8051 CPU 内核架构，内置 16K*8bit ROM，256*8bit 内部 SRAM，512*8bit 外部 SRAM，内部集成了 TOUCH、ADC、TIMER、PWM、RTC、LVR、WDT、UART、Mini I2C 等外设。

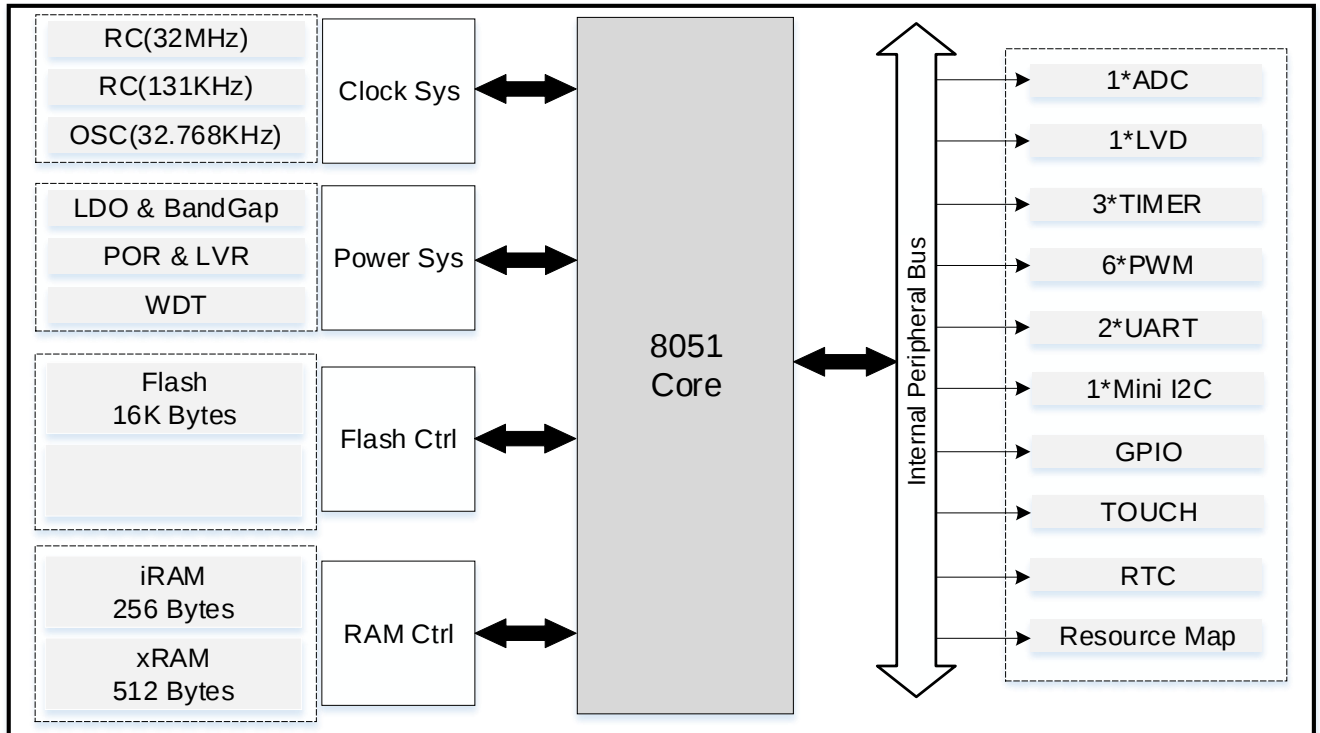


图 3 系统框图

5. 中央处理器

所有指令兼容标准的 8051 指令。

5.1 指令集

表 2 指令集符号说明

符号	说明
Rn	工作寄存器 R0-R7
direct	内部 RAM 的低 128B 地址或特殊功能寄存器
@Ri	由寄存器 R0 或 R1 间接寻址的内部或外部 RAM 地址
#data	包含在指令中的 8 位常数（立即数），为指令的第 2 字节
#data16	包含在指令中的 16 位常数（立即数），为指令的第 2 和 3 字节
bit	128 个位寻址区地址或特殊功能寄存器中可位寻址标志位
A	累加器
addr16	LCALL 或 LJMP 的目的地址，64KB ROM 空间的任意地址跳转
addr11	ACALL 或 AJMP 的目的地址，PC 高 5 位地址不变时的 2KB ROM 地址跳转
rel	SJMP 和条件跳转指令的 8 位偏移量，范围是下条指令第一字节+127/-128

表 3 指令意义((X1))ROM→X2 说明

符号	说明
X1、X2	寻址方式，包括 direct(字节)、Rn\ Ri(寄存器)、DPTR(数据指针)
(X1)	X1 地址中的数据
((X1))	X1 地址中内容是另一个数据的地址
→	存至某地址

表 4 指令集

类 别	序 号	指令格式	指令意义	操作码	指令字节	指令周期	标志位
算 术 运 算	1	ADD A,Rn	$(A) + (Rn) \rightarrow A$	0x28~0x2F	1	1	OV AC Cy
	2	ADD A,direct	$(A) + (\text{direct}) \rightarrow A$	0x25	2	2	
	3	ADD A,@Ri	$(A) + ((Ri)) \rightarrow A$	0x26~0x27	1	2	
	4	ADD A,#data	$(A) + \#data \rightarrow A$	0x24	2	2	
	5	ADDC A,Rn	$(A) + (Rn) + (Cy) \rightarrow A$	0x38~0x3F	1	1	
	6	ADDC A,direct	$(A) + (\text{direct}) + (Cy) \rightarrow A$	0x35	2	2	
	7	ADDC A,@Ri	$(A) + ((Ri)) + (Cy) \rightarrow A$	0x36~0x37	1	2	
	8	ADDC A,#data	$(A) + \#data + (Cy) \rightarrow A$	0x34	2	2	
	9	SUBB A,Rn	$(A) - (Rn) - (Cy) \rightarrow A$	0x98~0x9F	1	1	
	10	SUBB A,direct	$(A) - (\text{direct}) - (Cy) \rightarrow A$	0x95	2	2	
	11	SUBB A,@Ri	$(A) - ((Ri)) - (Cy) \rightarrow A$	0x96~0x97	1	2	
	12	SUBB A,#data	$(A) - \#data - (Cy) \rightarrow A$	0x94	2	2	
	13	INC A	$(A) + 1 \rightarrow A$	0x04	1	1	-
	14	INC Rn	$(Rn) + 1 \rightarrow Rn$	0x08~0x0F	1	1	-
	15	INC direct	$(\text{direct}) + 1 \rightarrow \text{direct}$	0x05	2	2	-
	16	INC @Ri	$((Ri)) + 1 \rightarrow (Ri)$	0x06~0x07	1	2	-
	17	INC DPTR	$(DPTR) + 1 \rightarrow DPTR$	0xA3	1	1	-
	18	DEC A	$(A) - 1 \rightarrow A$	0x14	1	1	-
	19	DEC Rn	$(Rn) - 1 \rightarrow Rn$	0x18~0x1F	1	1	-
	20	DEC direct	$(\text{direct}) - 1 \rightarrow \text{direct}$	0x15	2	2	-
	21	DEC @Ri	$((Ri)) - 1 \rightarrow (Ri)$	0x16~0x17	1	2	-
	22	MUL AB	$(A) \cdot (B)$ 的低字节 $\rightarrow A$, $(A) \cdot (B)$ 的高字节 $\rightarrow B$	0xA4	1	4	OV
	23	DIV AB	$(A)/(B)$ 的商 $\rightarrow A$, $(A)/(B)$ 的余数 $\rightarrow B$	0x84	1	4	Cy
	24	DA A	对 A 进行十进制调整	0xD4	1	1	Cy
逻 辑 运 算	25	ANL A,Rn	$(A) \& (Rn) \rightarrow A$	0x58~0x5F	1	1	-
	26	ANL A,direct	$(A) \& (\text{direct}) \rightarrow A$	0x55	2	2	-
	27	ANL A,@Ri	$(A) \& ((Ri)) \rightarrow A$	0x56~0x57	1	2	-
	28	ANL A,#data	$(A) \& \#data \rightarrow A$	0x54	2	2	-
	29	ANL direct,A	$(\text{direct}) \& (A) \rightarrow \text{direct}$	0x52	2	2	-
	30	ANL direct,#data	$(\text{direct}) \& \#data \rightarrow \text{direct}$	0x53	3	3	-
	31	ORL A,Rn	$(A) (Rn) \rightarrow A$	0x48~0x4F	1	1	-
	32	ORL A,direct	$(A) (\text{direct}) \rightarrow A$	0x45	2	2	-
	33	ORL A,@Ri	$(A) ((Ri)) \rightarrow A$	0x46~0x47	1	2	-
	34	ORL A,#data	$(A) \#data \rightarrow A$	0x44	2	2	-
	35	ORL direct,A	$(\text{direct}) (A) \rightarrow \text{direct}$	0x42	2	2	-
	36	ORL direct,#data	$(\text{direct}) \#data \rightarrow \text{direct}$	0x43	3	3	-
	37	XRL A,Rn	$(A) \wedge (Rn) \rightarrow A$	0x68~0x6F	1	1	-
	38	XRL A,direct	$(A) \wedge (\text{direct}) \rightarrow A$	0x65	2	2	-
	39	XRL A,@Ri	$(A) \wedge ((Ri)) \rightarrow A$	0x66~0x67	1	2	-
	40	XRL A,#data	$(A) \wedge \#data \rightarrow A$	0x64	2	2	-
	41	XRL direct,A	$(\text{direct}) \wedge (A) \rightarrow \text{direct}$	0x62	2	2	-

类别	序号	指令格式	指令意义	操作码	指令字节	指令周期	标志位
	42	XRL direct,#data	(direct) ^ #data→direct	0x63	3	3	-
	43	CLR A	0→A	0xE4	1	1	-
	44	CPL A	~(A)→A	0xF4	1	1	-
	45	RL A	A 循环左移 1 位	0x23	1	1	-
	46	RLC A	A 带进位循环左移 1 位	0x33	1	1	Cy
	47	RR A	A 循环右移 1 位	0x03	1	1	-
	48	RRC A	A 带进位循环右移 1 位	0x13	1	1	Cy
	49	SWAP A	A 半字节交换	0xC4	1	1	-
数据 传 送	50	MOV A,Rn	(Rn)→A	0xE8~0xEF	1	1	-
	51	MOV A,direct	(direct)→A	0xE5	2	2	-
	52	MOV A,@Ri	((Ri))→A	0xE6~0xE7	1	2	-
	53	MOV A,#data	data→A	0x74	2	2	-
	54	MOV Rn,A	(A)→Rn	0xF8~0xFF	1	1	-
	55	MOV Rn,direct	(direct)→Rn	0xA8~0xAF	2	2	-
	56	MOV Rn,#data	data→Rn	0x78~0x7F	2	2	-
	57	MOV direct,A	(A)→direct	0xF5	2	2	-
	58	MOV direct,Rn	(Rn)→direct	0x88~0x8F	2	2	-
	59	MOV direct1,direct2	(direct2)→direct1	0x85	3	3	-
	60	MOV direct,@Ri	((Ri))→direct	0x86~0x87	2	2	-
	61	MOV direct,#data	data→direct	0x75	3	3	-
	62	MOV @Ri,A	(A)→(Ri)	0xF6~0xF7	1	1	-
	63	MOV @Ri,direct	(direct)→(Ri)	0xA6~0xA7	2	2	-
	64	MOV @Ri,#data	data→(Ri)	0x76~0x77	2	2	-
	65	MOV DPTR,#data16	data16→DPTR	0x90	3	3	-
	66	MOVC A,@A+DPTR	((A) + (DPTR)) ROM→A	0x93	1	3	-
	67	MOVC A,@A+PC	((A) + (PC))ROM→A	0x83	1	3	-
	68	MOVX A,@Ri	((Ri))→A	0xE2~0xE3	1	3	-
	69	MOVX A,@DPTR	((DPTR))→A	0xE0	1	3	-
	70	MOVX @Ri,A	(A)→(Ri)	0xF2~0xF3	1	3	-
	71	MOVX @DPTR,A	(A)→(DPTR)	0xF0	1	3	-
	72	PUSH direct	(SP) + 1→SP, (direct)→(SP)	0xC0	2	2	-
	73	POP direct	((SP))→direct, (SP) - 1→SP	0xD0	2	2	-
	74	XCH A,Rn	(A)↔(Rn)	0xC8~0xCF	1	1	-
	75	XCH A,direct	(A)↔(direct)	0xC5	2	2	-
	76	XCH A,@Ri	(A)↔((Ri))	0xC6~0xC7	1	2	-
	77	XCHD A,@Ri	(A)低 4 位↔((Ri))低 4 位	0xD6~0xD7	1	2	-
程 序 分 支	78	ACALL addr11	(PC) + 2→PC, (SP) + 1→SP, (PC)L→(SP), (SP) + 1→SP, (PC)H→(SP), addr11→ PC10~0	xxx10001b	2	2	-
	79	LCALL addr16	(PC) + 3→PC, (SP) + 1→SP, (PC)L→(SP), (SP) + 1→SP, (PC)H→(SP), addr16 →PC	0x12	3	3	-

类别	序号	指令格式	指令意义	操作码	指令字节	指令周期	标志位
	80	RET	子程序返回(SP)→(PC)H, (SP-1)→(PC)L	0x22	1	4	-
	81	RETI	中断返回(SP)→(PC)H, (SP-1)→(PC)L	0x32	1	4	-
	82	AJMP addr11	2KB 程序跳转: addr11→PC[10:0]	xxx00001b	2	2	-
	83	LJMP addr16	64KB 程序跳转: addr16→PC[15:0]	0x02	3	3	-
	84	SJMP rel	相对跳转 rel 地址	0x80	2	3	-
	85	JMP @A+DPTR	(A) + (DPTR)→PC	0x73	1	2	-
	86	JZ rel	如果 A=0, 相对跳转 rel 地址	0x60	2	3	-
	87	JNZ rel	如果 A≠0, 相对跳转 rel 地址	0x70	2	3	-
	88	JC rel	如果 Cy=1, 相对跳转 rel 地址	0x40	2	3	-
	89	JNC rel	如果 Cy=0, 相对跳转 rel 地址	0x50	2	3	-
	90	JB bit,rel	如果 bit=1, 相对跳转 rel 地址	0x20	3	4	-
	91	JNB bit,rel	如果 bit=0, 相对跳转 rel 地址	0x30	3	4	-
	92	JBC bit,rel	如果 bit=1, 则 0→bit, 再相对跳转 rel 地址	0x10	3	4	-
	93	CJNE A,direct,rel	(A) - (direct) ≠0, 相对跳转 rel 地址 (A) < (direct) → Cy=1 (A) ≥ (direct) → Cy=0	0xB5	3	4	Cy
	94	CJNE A,#data,rel	(A) - data ≠0, 相对跳转 rel 地址 (A) < data → Cy=1 (A) ≥ data → Cy=0	0xB4	3	4	Cy
	95	CJNE Rn,#data,rel	(Rn) - data ≠0, 相对跳转 rel 地址 (Rn) < data → Cy=1 (Rn) ≥ data → Cy=0	0xB8~0xBF	3	4	Cy
	96	CJNE @Ri,#data,rel	((Ri)) - data ≠0, 相对跳转 rel 地址 ((Ri)) < data → Cy=1 ((Ri)) ≥ data → Cy=0	B6~B7	3	5	Cy
	97	DJNZ Rn,rel	(Rn)-1→Rn, 如果(Rn)≠0, 相对跳转 rel 地址	D8~DF	2	3	-
	98	DJNZ direct,rel	(direct)-1→direct, 如果(direct)≠0, 相对跳转 rel 地址	D5	3	4	-
	99	NOP	空指令不做任何操作	0	1	1	-
布尔操作	100	CLR C	0→Cy	0xC3	1	1	-
	101	CLR bit	0→bit	0xC2	2	2	-
	102	SETB C	1→Cy	0xD3	1	1	-
	103	SETB bit	1→bit	0xD2	2	2	-
	104	CPL C	!Cy→Cy	0xB3	1	1	-
	105	CPL bit	!bit→bit	0xB2	2	2	-
	106	ANL C,bit	Cy & bit→Cy	0x82	2	2	Cy
	107	ANL C,/bit	Cy & !bit→Cy	0xB0	2	2	Cy
	108	ORL C,bit	Cy bit→Cy	0x72	2	2	-
	109	ORL C,/bit	Cy !bit→Cy	0xA0	2	2	-
	110	MOV C,bit	bit→Cy	0xA2	2	2	-
	111	MOV bit,C	Cy→bit	0x92	2	2	-

5.2 寻址方式

说明：指令集符号说明见 5.1

1. 立即寻址

在指令操作数中直接给出参加运算的操作数，通常用“#”作为前缀。指令举例如下：

```
MOV A, #20H
MOV SP, #81H
MOV DPTR, #1234H
```

2. 直接寻址

指令中直接给出操作数的地址，使用于内部数据存储器 RAM 低 128 单元和 80H-FFH 区的 SFR 区。指令举例如下：

```
MOV 65H, A      ;将 A 的内容送入内部 RAM 的 65H 单元地址中
MOV A, direct   ;将直接地址单元的内容送入 A 中
MOV direct, direct ;将直接地址单元的内容送直接地址单元
MOV P1, #85H    ;将立即数 85H 送入中断允许寄存器 IE
```

3. 间接寻址

间接寻址方式是指寄存器中存放的是操作数的地址，即操作数是通过寄存器间接得到的，因此称为寄存器间接寻址。间接寻址采用 R0 或 R1 前添加“@”符号来表示。

```
MOV A, @R0      ;把内部 R0 内容指定地址里面的数据放到累加器 A 中
MOVX A, @R0     ;把外部 R0 内容指定地址里面的数据放到累加器 A 中
MOVX A, @DPTR   ;把外部 DPTR 内容指定地址里面的数据放到累加器 A 中，通常用来访问片外数据存储器
```

4. 寄存器寻址

寄存器寻址时对选定的工作寄存器 R7~R0、累加器 A、通用寄存器 B、数据指针 DPTR 进行操作。其中寄存器 R7~R0 由指令码的低 3 位表示，R7~R0、ACC、B、DPTR 隐含在指令码中。寄存器工作区的选择由程序状态字寄存器 PSW 中的 RS1、RS0 来决定，指令操作数指定的寄存器均指当前工作区的寄存器。

```
MOV A, R2
INC R0
```

5. 相对寻址

把指令中给定的地址偏移量与本指令所在单元地址（PC 内容）相加得到真正有效的操作数所存放的地址。由于目的地址是相对于 PC 中的基地址而言，所以这种寻址方式称为相对寻址。偏移量为带符号的数，所能表示的范围为+127~-128，这种寻址方式主要用于跳转指令。

```
JMP @A+DPTR
JC $+100H      ;表示若进位 C 为 0，则程序计数器 PC 中的内容不改变，即不跳转；若进位 C 为 1，则以 PC 中的当前值为基地址，加上偏移量 100H 后所得到的结果作为该跳转指令的目的地址。
```

6. 变址寻址

在变址寻址方式中，指令操作数指定一个存放变址基址的变址寄存器。变址寻址时，偏移量与变址基值相加，其结果作为操作数的地址。变址寄存器有程序计数器 PC 和地址寄存器 DPTR。

MOVC A, @A+DPTR

JMP @A+DPTR

MOVC A, @A+PC

表示累加器 A 为偏移量寄存器，其内容与地址寄存器 DPTR 中的内容相加，其结果作为操作数的地址，取出该单元中的数送入累加器 A 中。

7. 位寻址

位寻址是位操作指令的操作数是位地址，其对内部 RAM 或可位寻址的特殊功能寄存器 SFR 内的某个位直接置位为 1 或复位为 0。内部数据存储器 RAM 地址 20H-2FH 为位寻址区，共 16 个单元，每个单元 8 位，位寻址区总计 128 位，这 128 位都有一个单独的位地址为：00H-7FH。

特殊功能寄存器地址低 8 位为 0 或 8 的也可位寻址，其位地址是 80H~FFH。

MOV C,20H

SETB 81H

5.3 CPU寄存器

CPU 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
E0h	ACC	ACC[7:0]								0000 0000
F0h	B	B[7:0]								0000 0000
81h	SP	SP[7:0]								0000 0111
82h	DP0L	DP0[7:0]								0000 0000
83h	DP0H	DP0[15:8]								0000 0000
84h	DP1L	DP1[7:0]								0000 0000
85h	DP1H	DP1[15:8]								0000 0000
D0h	PSW	CY	AC	F0	RS[1:0]		OV	F1	P	0000 0000
92h	DPS	-							DPS0	---- ---0

- **程序计数器 PC**

程序计数器 PC 不属于 SFR 寄存器。PC 有 16 位，是用来控制指令执行顺序的寄存器。单片机上电或者复位后，PC 值为 0000H，也就是说单片机程序从 0000H 地址开始执行程序。

- **累加器 ACC**

累加器 ACC 是 8051 内核单片机的最常用的寄存器之一，指令系统中采用 A 作为助记符。常用来存放参加计算或者逻辑运算的操作数及结果。

- **B 寄存器**

B 寄存器在乘除法运算中必须与累加器 A 配合使用：乘法指令 MUL A,B 把累加器 A 和寄存器 B 中的 8 位无符号数相乘，所得的 16 位乘积的低位字节放在 A 中，高位字节放在 B 中。除法指令 DIV A,B 是用 A 除以 B，整数商放在 A 中，余数放在 B 中。

寄存器 B 还可以作为通用的暂存寄存器使用。

- **堆栈指针 SP**

堆栈指针是一个 8 位的专用寄存器，它指示了堆栈顶部在通用 RAM 中的位置。单片机复位后，SP 初始值为 07H，即堆栈会从 08H 开始向上增加。由于 08H~1FH 为工作寄存器组 R1~R3，建议用户在初始化程序时设置 SP 在 30H 以后的单元区间为宜。

- **数据指针 DP0、数据指针 DP1**

内核为 DP0 和 DP1 双数据指针结构，能加速数据存储移动。数据指针为 16 位的专用寄存器，它们既可以作为一个 16 位寄存器 DP0 和 DP1 来处理，也可以作为独立的 8 位寄存器 DP0L、DP0H 和 DP1L、DP1H 来处理。

● **状态寄存器(PSW, D0h):**

CPU 状态寄存器包含运算标志, 结果标志。

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	CY	AC	F0	RS[1:0]		OV	F1	P
Access	R/W	R/W	R/W	R/W		R/W	R/W	R
Default	0	0	0	00		0	0	0

Bit[7] CY: 进位标志位

- 1: 算术或逻辑运算中有进位或借位发生
- 0: 算术或逻辑运算中没有进位或借位发生

Bit[6] AC: 辅助进位标志位, 即半字节进位标志

- 1: 算术或逻辑运算中低 4 位有辅助进位或借位发生
- 0: 算术或逻辑运算中低 4 位没有辅助进位或借位发生

Bit[5] F0: F0 标志位, 用户自定义标志位

Bit[4:3] RS: R0~R7 寄存器页选择位

- 00: 页 0 (映射到 00h~07h)
- 01: 页 1 (映射到 08h~0Fh)
- 10: 页 2 (映射到 10h~17h)
- 11: 页 3 (映射到 18h~1Fh)

Bit[2] OV: 溢出标志位

- 1: 有溢出发生
- 0: 没有溢出发生

Bit[1] F1: F1 标志位, 用户自定义标志位

Bit[0] P: ACC 奇偶标志位

- 1: 累加器的二进制值有奇数个 1
- 0: 累加器的二进制值有偶数个 1

● **数据指针选择寄存器(DPS, 92h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-							DPS0
Access	-							R/W
Default	-							0

Bit[0] DPS0: 数据指针选择位

- 1: 数据指针 1
- 0: 数据指针 0

内核为 DP0 和 DP1 双数据指针结构, 能加速数据存储移动。数据指针为 16 位的专用寄存器, 它们既可以作为一个 16 位寄存器 DP0 和 DP1 来处理, 也可以作为独立的 8 位寄存器 DP0L、DP0H 和 DP1L、DP1H 来处理。通过寄存器 DPS 进行配置, 可选择数据指针 0 或 1, 所有读取或操作数据指针的相关指令将会选择最近一次配置选择的数据指针。

6. 存储系统(ROM/RAM)

6.1 RAM

共集成 768 字节的数据存储器 SRAM，由内部 256 字节和外部 512 字节组成。

- 内部 256 字节 SRAM:

- 内部 256 字节 SRAM 的低 128 字节（地址 00H~7FH）可直接寻址的方式访问，也可通过间接寻址的方式来访问，分为三部分：
 - 工作寄存器组 0~3，地址 00H~1FH，程序状态字寄存器 PSW 中的 RS0、RS1 组合决定了当前使用的工作寄存器，使用工作寄存器组 0~3 可加快运算的速度；
 - 位寻址区 20H~2FH，此区域用户可以使用普通 RAM 也可用作按位寻址 RAM。按位寻址时，位地址为 00H~7FH（此地址按位编地址，不同于通用 SRAM 按字节编地址），程序中可由指令区分；
 - 用户 RAM 区和堆栈 RAM 区 30H~7FH：复位过后 8 位堆栈指针 SP 指向堆栈区(默认值为 0x07)，用户一般会在初始化程序时设置初值，建议设置 30H 以后的单元区间。
- 内部 256 字节 SRAM 高 128 字节（地址 80H~FFH）只能用间接寻址的方式访问。
- 特殊功能寄存器 SFR 的地址也是 80H~FFH，它的地址与内部 256 字节 SRAM 高 128 字节地址相同，区别是 SFR 寄存器是直接寻址，而高 128 字节 SRAM 只能是间接寻址。

- 外部 512 字节 SRAM，只能用 MOVX 指令访问。

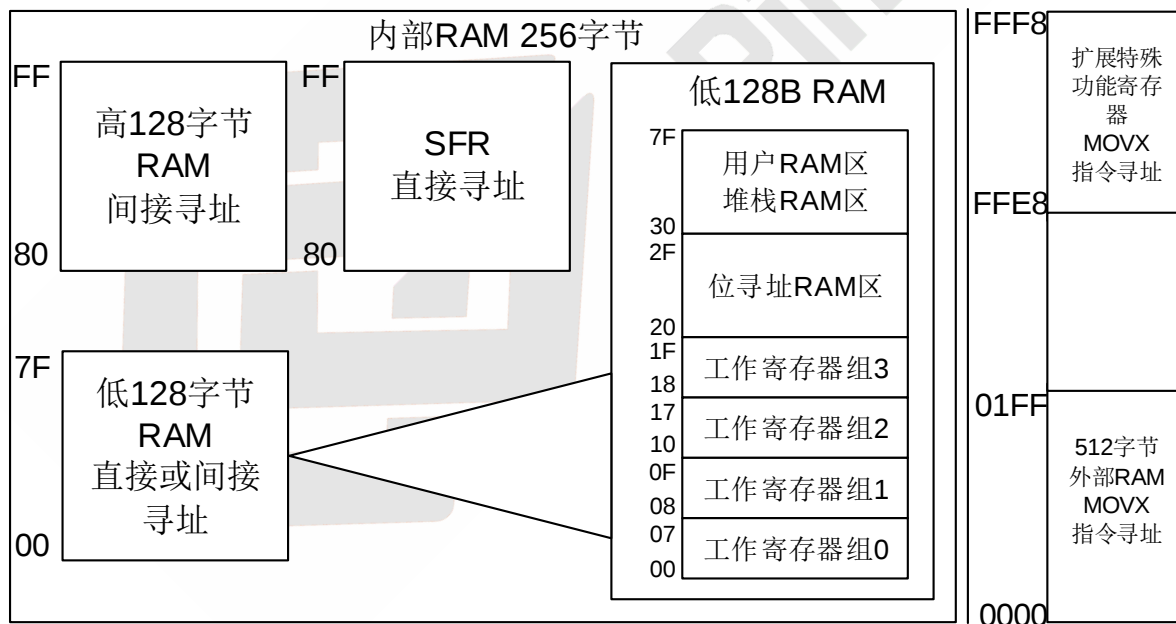


图 4 数据存储器

6.2 ROM

程序存储用于存放用户程序、数据等信息，单片机内部集成了 16K*8bit 的 CODE 区可擦写 10 万次，详情如下结构图。其中 003H-07BH 是单片机各种中断服务程序入口地址。

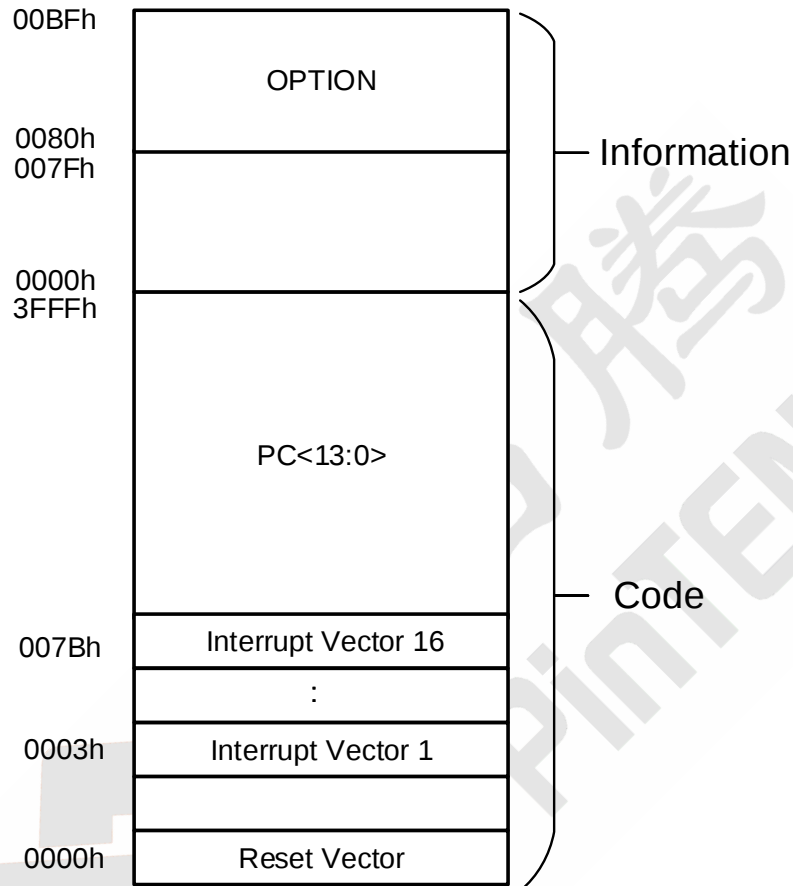


图 5 程序存储器 ROM

6.3 OPTION

表 5 配置选项 0(Information 区 0x80)

名称	位	默认	说明
LVRSEL[2:0]	[6:4]	100	低电压复位点选择 000: 关闭 001: 1.8V 010: 2.0V 011: 2.2V 100: 2.4V 101: 2.7V 110: 2.9V 111: 3.6V
SUT[1:0]	[3:2]	11	PWRT 计数周期选择位 11: PWRT = 16ms (default) 10: PWRT = 8ms 01: PWRT = 256ms 00: PWRT = 128ms
RESETP00_EN	[1]	0	P00 作为外部复位脚选择 1: P00 作为外部复位脚 0: P00 不作为外部复位(默认)
XTALS	[0]	0	32.768KHz 晶体振荡器选择 0: P26\P27 不复用为 XTAL 1: P26\P27 复用为 XTAL

表 6 配置选项 1(Information 区 0x81)

名称	位	默认	说明
OSCD[1:0]	[5:4]	10	CPU 时钟周期选择位 11: 2 个 RC32M 周期 10: 4 个 RC32M 周期(默认) 00: 8 个 RC32M 周期 01: 16 个 RC32M 周期

表 7 配置选项 2(Information 区 0x82)

名称	位	默认	说明
PROTECT	[7]	1	代码保护选择位 1: 代码不加密(默认) 0: 加密

7. 特殊功能寄存器(SFR)

7.1 SFR地址

地 址	可位寻址	不可位寻址						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
80h	SYSCON	SP	DP0L	DP0H	DP1L	DP1H	IOSET	PCON
88h	P0	P0OD	P0PH	P0PD	P0OE	P1OD	P1PH	P1PD
90h	P1	P1OE	DPS	P2PH	P2PD	P2OE	P3OD	P3PD
98h	P2	P3OE	T0CON	T0C	T0OVR	T1CON	T1OVR	T1OVRH
A0h	P3	T1CL	T1CH	T2CON	T2CL	T2CH	T2OVR	T2OVRH
A8h	IE0	IAPCON1	IAPTRI	IAPADRL	IAPADRH	IAPDAT	IAPCON2	P3PH
B0h	IE1	IF0	IF1	INTS	INTEN	IPH	IPL	-
B8h	THIF	THCGF0	THCFG1	THCFG2	THCFG3	THCON0	THCON1	THCON2
C0h	THCON3	THCON4	THOL	THOH	THNLINEL	THNLINELH	THPLINEL	THPLINELH
C8h	ADCON0	ADCON1	ADCON2	ADCON3	ADCCFG0	ADCCFG1	ADCCFG2	ADCOL
D0h	PSW	ADCOH	WDTSET	WDTCON	PWMIECFG	PWMCNT	P0WK	P2OD
D8h	RTCCON	RTCSS	RTCS	RTCM	RTCWH	RTCDL	RTCDH	RTCAS
E0h	ACC	RTCAM	RTCAH	RTCMSS	RTCCAL	UAT0CON	LVDCON0	LVDCON1
E8h	I2CCON0	I2CSPD	I2CSTATE	I2CDAT	I2CCON1	TLCFG0	TLCFG1	TLCFG2
F0h	B	TLCFG3	TLSEG0	TLSEG1	TLSEG2	TLSEG3	TLCNTL	TLCNTH
F8h	RTCIF	UAT1CON	UAT0SBUF	UAT1SBUF	UAT0SREL	UAT0SREH	UAT1SREL	UAT1SREH

7.2 扩展特殊功能寄存器地址

地址	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
FFE8h	MAP0	MAP1	MAP2	MAP3	MAP4	PWMCON0	PWMCON1	PWMCON2
FFF0h	PWMD0L	PWMD0H	PWMD1L	PWMD1H	PWMD2L	PWMD2H	PWMD3L	PWMD3H
FFF8h	PWMD4L	PWMD4H	PWMD5L	PWMD5H	-			

- 上述扩展特殊功能寄存器，只能用 **MOVX** 指令访问。

7.3 SFR汇总

表 8 寄存器列表 (SFR)

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
80h	SYSCON	WRST	CDCVS	XTALF	LRCCAL	OCDSM	LPVMS	CMODFS	XTALO	0000 0010
81h	SP	SP[7:0]								0000 0111
82h	DP0L	DP0[7:0]								0000 0000
83h	DP0H	DP0[15:8]								0000 0000
84h	DP1L	DP1[7:0]								0000 0000
85h	DP1H	DP1[15:8]								0000 0000
86h	IOSET	IOPHS	IOPDS	OLDIS[1:0]		PUDIS[2:0]			VTHS	0010 1000
87h	PCON	-						STOP	IDLE	---- --00
88h	P0	P07	P06	P05	P04	P03	P02	P01	P00	xxxx xxxx
89h	P0OD	P0OD7	P0OD6	P0OD5	P0OD4	P0OD3	P0OD2	P0OD1	P0OD0	0000 0000
8Ah	P0PH	P0PH7	P0PH6	P0PH5	P0PH4	P0PH3	P0PH2	P0PH1	P0PH0	1111 1111
8Bh	P0PD	P0PD7	P0PD6	P0PD5	P0PD4	P0PD3	P0PD2	P0PD1	P0PD0	0000 0000
8Ch	P0OE	P0OE7	P0OE6	P0OE5	P0OE4	P0OE3	P0OE2	P0OE1	P0OE0	1111 1111
8Dh	P1OD	P1OD7	P1OD6	P1OD5	P1OD4	P1OD3	P1OD2	P1OD1	P1OD0	0000 0000
8Eh	P1PH	P1PH7	P1PH6	P1PH5	P1PH4	P1PH3	P1PH2	P1PH1	P1PH0	1111 1111
8Fh	P1PD	P1PD7	P1PD6	P1PD5	P1PD4	P1PD3	P1PD2	P1PD1	P1PD0	0000 0000
90h	P1	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxx
91h	P1OE	P1OE7	P1OE6	P1OE5	P1OE4	P1OE3	P1OE2	P1OE1	P1OE0	1111 1111
92h	DPS	-							DPS0	---- --0
93h	P2PH	P2PH7	P2PH6	P2PH5	P2PH4	P2PH3	P2PH2	P2PH1	P2PH0	1111 1111
94h	P2PD	P2PD7	P2PD6	P2PD5	P2PD4	P2PD3	P2PD2	P2PD1	P2PD0	0000 0000
95h	P2OE	P2OE7	P2OE6	P2OE5	P2OE4	P2OE3	P2OE2	P2OE1	P2OE0	1111 1111
96h	P3OD	-					P3OD2	P3OD1	P3OD0	---- -000
97h	P3PD	-					P3PD2	P3PD1	P3PD0	---- -000
98h	P2	P27	P26	P25	P24	P23	P22	P21	P20	xxxx xxxx
99h	P3OE	-					P3OE2	P3OE1	P3OE0	---- -111
9Ah	T0CON	T0M	T0EG	-		T0FS[2:0]			T0EN	10-- 0000
9Bh	T0C	T0C[7:0]								0000 0000
9Ch	T0OVR	T0OVR[7:0]								0000 0000
9Dh	T1CON	-				T1FS[2:0]			T1EN	---- 0000
9Eh	T1OVR L	T1OVR L[7:0]								0000 0000
9Fh	T1OVR H	T1OVR H[7:0]								0000 0000
A0h	P3	-					P32	P31	P30	---- -xxx
A1h	T1CL	T1CL[7:0]								0000 0000
A2h	T1CH	T1CH[7:0]								0000 0000

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
A3h	T2CON	-				T2FS[2:0]			T2EN	---- 0000
A4h	T2CL	T2CL[7:0]								0000 0000
A5h	T2CH	T2CH[7:0]								0000 0000
A6h	T2OVR _L	T2OVR _L [7:0]								0000 0000
A7h	T2OVR _H	T2OVR _H [7:0]								0000 0000
A8h	IE0	EA	ADCIE	THIE	T2IE	T1IE	T0IE	INT1IE	INT0IE	0000 0000
A9h	IAPCON1	INFEN	IAPLOCK	-			IAPCMD[2:0]			00-- -000
AAh	IAPTRI	IAPTRI[7:0]								0000 0000
ABh	IAPAD _{R_L}	-		IAPADR[5:0]						--00 0000
ACH	IAPAD _{R_H}	IAPADR[13:6]								0000 0000
ADh	IAPDAT	IAPDAT[7:0]								0000 0000
A _E h	IAPCON2	PRTIME[3:0]				ERTIME[3:0]				0000 0000
A _F h	P3PH	-					P3PH2	P3PH1	P3PH0	---- -111
B0h	IE1	LPVTIE	UAT1TIE	UAT0TIE	UAT1RIE	UAT0RIE	PWMIE	LVDIE	RTCIE	0000 0000
B1h	IF0	I2CIF	ADCIF	-	T2IF	T1IF	T0IF	INT1IF	INT0IF	00-0 0000
B2h	IF1	LPVTIF	UAT1TIF	UAT0TIF	UAT1RIF	UAT0RIF	PWMIF	LVDIF	-	0000 000-
B3h	INTS	-				INT1S[1:0]		INT0S[1:0]		---- 0000
B4h	INTEN	-						EINT1	EINT0	---- --00
B5h	IPH	LPVTIP	UAT1TIP	UAT0TIP	UAT1RIP	UAT0RIP	PWMIP	LVDIP	RTCIP	0000 0000
B6h	IPL	I2CIP	ADCIP	THIP	T2IP	T1IP	T0IP	INT1IP	INT0IP	0000 0000
C8h	ADCON0	ADCVO	ADCCUR	ADCE	ADCS	ADCEN	ADCVREF[2:0]			0100 0000
C9h	ADCON1	ADCC[2:0]			-	ADCSPT[3:0]			000- 0000	
CAh	ADCON2	-			ADCCALD	ADCCAL[3:0]			---0 0000	
CBh	ADCON3	-			ADCADDR[4:0]					---0 0000
CCh	ADCCFG0	ADCCH7	ADCCH6	ADCCH5	ADCCH4	ADCCH3	ADCCH2	ADCCH1	ADCCH0	0000 0000
CDh	ADCCFG1	ADCCH15	ADCCH14	ADCCH13	ADCCH12	ADCCH11	ADCCH10	ADCCH9	ADCCH8	0000 0000
CEh	ADCCFG2	-	ADCCH22	ADCCH21	ADCCH20	ADCCH19	ADCCH18	ADCCH17	ADCCH16	-000 0000
CFh	ADCOL	ADCOL[7:0]								xxxx xxxx
D0h	PSW	CY	AC	F0	RS[1:0]		OV	F1	P	0000 0000
D1h	ADCOH	-				ADCOH[3:0]				---- xxxx
D2h	WDTSET	WDTSET[7:0]								1111 1111
D3h	WDTCON	LRCEN	WDTSEL	WDTEN	WDTF	WDTC	PS[2:0]			1010 0111
D4h	PWMIECFG	PWMIES	-				PWMIS2	PWMIS1	PWMIS0	0--- -000
D5h	PWMCNT	PWMCNT[7:0]								0000 0001
D6h	P0WK	P0WK7	P0WK6	P0WK5	P0WK4	P0WK3	P0WK2	P0WK1	P0WK0	0000 0000
D7h	P2OD	P2OD7	P2OD6	P2OD5	P2OD4	P2OD3	P2OD2	P2OD1	P2OD0	0000 0000
D8h	RTCCON	RTCEN	MSEN	HSEN	SCEN	MCEN	HCEN	RTCWE	CALS	0000 0000

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
D9h	RTCSS	RTCSS[7:0]								0000 0000
DAh	RTCS	-		RTCS[5:0]						--00 0000
DBh	RTCM	-		RTCM[5:0]						--00 0000
DCh	RTCWH	RTCW[2:0]			RTCH[4:0]					0000 0000
DDh	RTCDL	RTCDL[7:0]								0000 0000
DEh	RTCDH	RTCDH[7:0]								0000 0000
DFh	RTCAS	-		RTCAS[5:0]						--00 0000
E0h	ACC	ACC[7:0]								0000 0000
E1h	RTCAM	-		RTCAM[5:0]						--00 0000
E2h	RTCAH	-		RTCAH[5:0]						--00 0000
E3h	RTMSS	RTMSS[7:0]								0000 0000
E4h	RTCCAL	OPTRC131K[7:0]								1000 0000
E5h	UAT0CON	-	UAT0EN	-	UAT0SM	-	UAT0RB8	-	UAT0TB8	-0-0 -0-0
E6h	LVDCON0	LV DEN	LVDOUT	TRDY	-					001- ----
E7h	LVDCON1	-					LVDSEL[2:0]			---- -000
E8h	I2CCON0	I2CEN	I2CSTA	I2CSTO	I2CMOD	I2CIE	-		GC	0000 0--0
E9h	I2CSPD	I2CSPD[7:0]								1010 0000
EAh	I2CSTATE	STASTA	STOSTA	GCSTA	-		I2CESTA[2:0]			000- -000
EBh	I2CDAT	I2CDAT[7:0]								0000 0000
ECh	I2CCON1	I2CADR[6:0]							I2CWR	1010 0000
EDh	TLCFG0	TLEN7	TLEN6	TLEN5	TLEN4	TLEN3	TLEN2	TLEN1	TLEN0	0000 0000
EEh	TLCFG1	TLEN15	TLEN14	TLEN13	TLEN12	TLEN11	TLEN10	TLEN9	TLEN8	0000 0000
EFh	TLCFG2	TLEN23	TLEN22	TLEN21	TLEN20	TLEN19	TLEN18	TLEN17	TLEN16	0000 0000
F0h	B	B[7:0]								0000 0000
F1h	TLCFG3	-						TLEN25	TLEN24	---- --00
F2h	TLSEG0	TLDATA7	TLDATA6	TLDATA5	TLDATA4	TLDATA3	TLDATA2	TLDATA1	TLDATA0	0000 0000
F3h	TLSEG1	TLDATA15	TLDATA14	TLDATA13	TLDATA12	TLDATA11	TLDATA10	TLDATA9	TLDATA8	0000 0000
F4h	TLSEG2	TLDATA23	TLDATA22	TLDATA21	TLDATA20	TLDATA19	TLDATA18	TLDATA17	TLDATA16	0000 0000
F5h	TLSEG3	-						TLDATA25	TLDATA24	---- --00
F6h	TLCNTL	TLCNT[7:0]								0000 0000
F7h	TLCNTH	CDCLDO	LRCFS	CDCPAR	TLCNT[12:8]					0000 0000
F8h	RTCIF	-					RTCAF	RTCHF	RTCMF	---- -000
F9h	UAT1CON	-	UAT1EN	-	UAT1SM	-	UAT1RB8	-	UAT1TB8	-0-0 -0-0
FAh	UAT0SBUF	UAT0SBUF[7:0]								0000 0000
FBh	UAT1SBUF	UAT1SBUF[7:0]								0000 0000
FCh	UAT0SREL	UAT0SREL[7:0]								1101 1001
FDh	UAT0SREH	-				UAT0SREH[3:0]				---- 0011

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
FEh	UAT1SREL	UAT1SREL[7:0]								1101 1001
FFh	UAT1SREH	-				UAT1SREH[3:0]				---- 0011
FFE8h	MAP0	-		I2CSDAM	I2CSCLM	UAT1RXM	UAT1TXM	T0M[1:0]		--00 0000
FFE9h	MAP1	-			INT0M[4:0]					---0 0101
FFEAh	MAP2	-			INT1M[4:0]					---0 0110
FFEBh	MAP3	-			PWM0M[4:0]					---0 0111
FFECb	MAP4	-			PWM1M[4:0]					---0 0111
FFEDh	PWMCON0	-			PWM3C	PWM4C	PWM5C	-		---0 00--
FFEEh	PWMCON1	-		PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	--00 0000
FFEFh	PWMCON2	-		PWM5S	PWM4S	PWM3S	PWM2S	PWM1S	PWM0S	--00 0000
FFF0h	PWMD0L	PWMD0L[7:0]								XXXX XXXX
FFF1h	PWMD0H	PWMD0H[7:0]								XXXX XXXX
FFF2h	PWMD1L	PWMD1L[7:0]								XXXX XXXX
FFF3h	PWMD1H	PWMD1H[7:0]								XXXX XXXX
FFF4h	PWMD2L	PWMD2L[7:0]								XXXX XXXX
FFF5h	PWMD2H	PWMD2H[7:0]								XXXX XXXX
FFF6h	PWMD3L	PWMD3L[7:0]								XXXX XXXX
FFF7h	PWMD3H	PWMD3H[7:0]								XXXX XXXX
FFF8h	PWMD4L	PWMD4L[7:0]								XXXX XXXX
FFF9h	PWMD4H	PWMD4H[7:0]								XXXX XXXX
FFFAh	PWMD5L	PWMD5L[7:0]								XXXX XXXX
FFFBh	PWMD5H	PWMD5H[7:0]								XXXX XXXX

说明：

- ： 无效位，回读为‘0’

x ： 不定态

8. 时钟系统

本芯片有三个时钟源：RC32M、RC131K 及 XTAL，其分别产生：CLKHRC、CLKLRC、CLKXTAL、CLKCPU、CLKPER、CLK4M，系统时钟结构如下图所示：

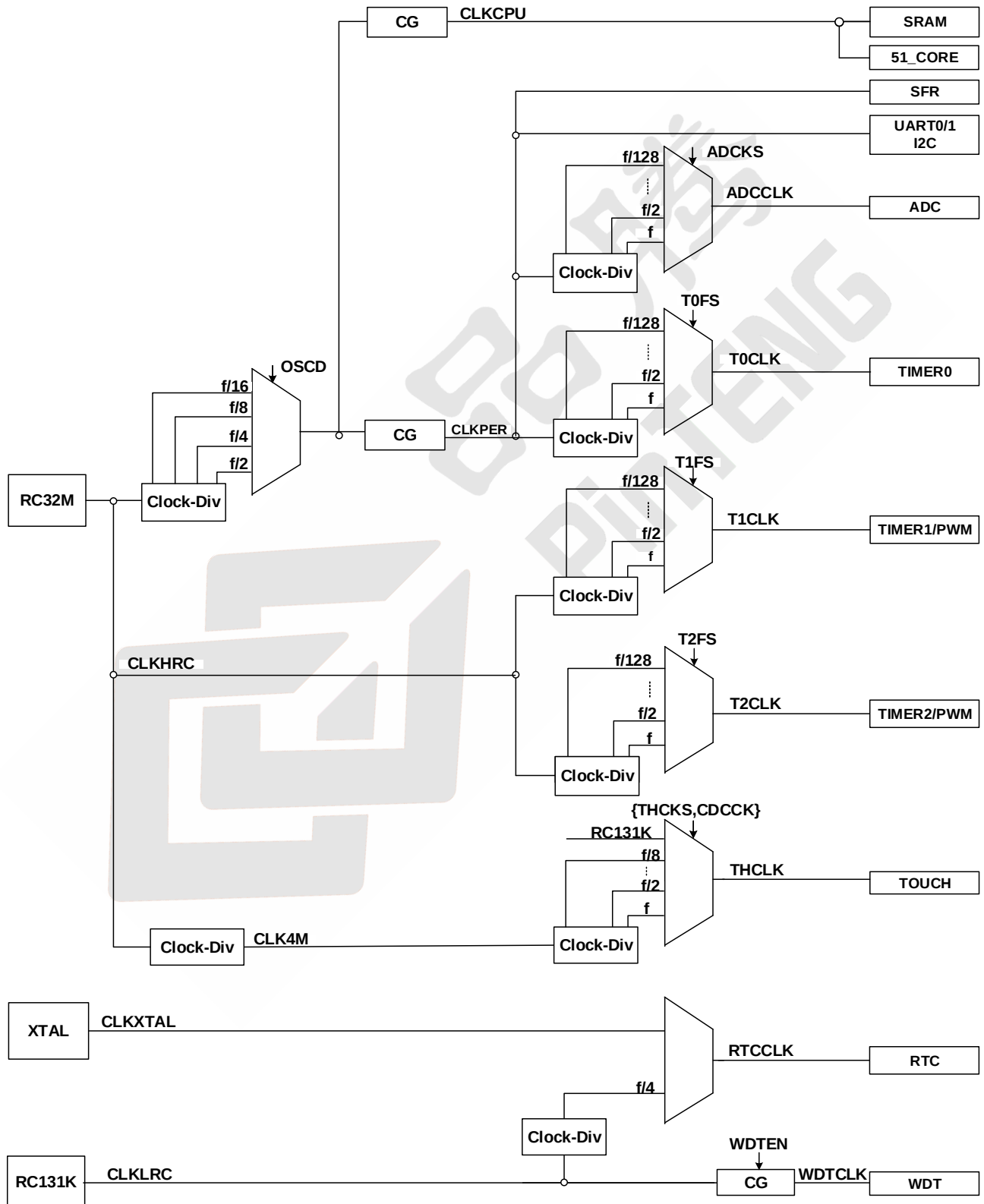


图 6 时钟结构图

注意：ADC、TIMER0\1\2 的分频系数需要在各自模块使能关闭时进行配置改变；TOUCH 模块时钟选择需要在非扫描转换期间配置改变。

RTC 模块的时钟由芯片内部的晶振监测电路在晶振停振时自动切换到内部 RC131K 时钟的四分频，监测到晶振起振后又自动切换到晶振时钟。

品 腾
PinTENG

9. 工作模式

支持 Normal 模式、STOP 模式、IDLE 模式。模式描述如下：

• 模式控制寄存器(PCON, 87h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-						STOP	IDLE
Access	-						R/W	R/W
Default	-						0	0

Bit[1] **STOP:** STOP 模式使能位

- 1: 进入 STOP 模式
- 0: 保持当前模式

Bit[0] **IDLE:** IDLE 模式使能位

- 1: 进入 IDLE 模式
- 0: 保持当前模式

• 系统控制寄存器(SYSCON, 80h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	WRST	CDCVS	XTALF	LRCCAL	OCDSM	LPVMS	CMODFS	XTALO
Access	W	R/W	R	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	0

Bit[7] **WRST:** 软件复位控制位

- 1: 对该位写 1 产生软件复位
- 0: 读该位始终为 0

Bit[6] **CDCVS:** CDCLDO 电压选择

- 1: 2.0V
- 0: 2.3V

Bit[5] **XTALF:** 外部晶振状态位

- 1: 外部晶振正常工作
- 0: 外部晶振停止工作

Bit[4] **LRCCAL:** 内部 LRC(RC131K)自动校准使能

- 1: 使能自动校准
 - 0: 关闭自动校准
- 当 XTALF 为 1 且 RTCEN 有效时才能自动校准

Bit[3] **OCDSM:** 调试模式位

- 1: 进入调试模式
- 0: 未进入调试模式

Bit[2] **LPVMS:** 低功耗电压监测点选择位

- 1: 4.5V
- 0: 4.0V

Bit[1] **CMODFS:** 选择 P01 口作为触摸/AD 模拟口或数字 IO 口

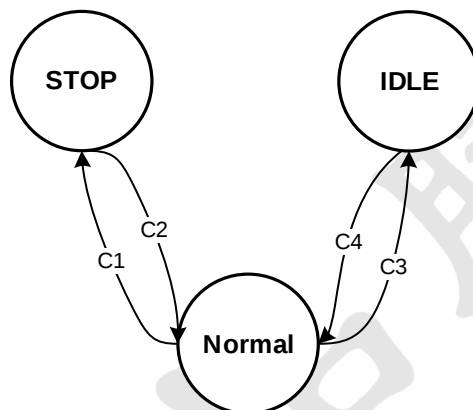
1: 作为触摸/AD 模拟口

0: 作为数字 IO 口

Bit[0] **XTALO**: 时钟输出控制位

1: 当 OPTION 中 OSCOUT 为 1, 外部晶振时钟从 P17 输出, RC32M、RC131K 时钟分别通过 P16、P15 输出。OSCOUT 为 0, 只有外部晶振时钟从 P17 输出。

0: 外部晶振时钟及内部时钟都不输出。



C1:配置PCON寄存器STOP位

C2:从STOP模式唤醒

C3:配置PCON寄存器IDLE位

C4:从IDLE模式唤醒

图 7 工作模式转换图

9.1 STOP模式

STOP 模式下, CPU 及外设都停止工作, 看门狗强制清零后保持运行状态, RC32M 停振, I/O 维持原状。如果使用外部晶振时钟的模块使能有效, 则外部晶振会保持开启, 否则外部晶振关闭。

唤醒方式:

1. 外部中断 0、外部中断 1、RTC 中断、触摸中断、LPVT 中断都可唤醒 STOP 模式
2. WDT 溢出可唤醒 STOP 模式
3. P0 口电平变化可唤醒 STOP 模式

9.2 IDLE模式

IDLE 模式下, 除 CPU 外, 其它外设都可工作。

唤醒方式:

1. 所有的 16 种中断都可唤醒 IDLE 模式
2. WDT 溢出可唤醒 IDLE 模式
3. P0 口电平变化可唤醒 IDLE 模式

说明: STOP/IDLE 模式中断唤醒时, 要求全局中断使能有效, 且相应的中断源使能有效。如果在中断服务程序中进入 STOP/IDLE 模式, 则需要更高优先级中断才能唤醒。

10. 中断系统

系统有如下 16 个中断源，对应不同的中断入口地址。

- 1) 定时器 T0 溢出中断
- 2) 定时器 T1 溢出中断
- 3) 定时器 T2 溢出中断
- 4) INT0 中断
- 5) INT1 中断
- 6) 触摸中断
- 7) ADC 中断
- 8) RTC 中断
- 9) PWM 中断
- 10) LVD 中断
- 11) UART0 接收中断
- 12) UART1 接收中断
- 13) UART0 发送中断
- 14) UART1 发送中断
- 15) LPVT 中断
- 16) I2C 中断

当一个中断进入时，当前的 PC 值被压栈，相应的中断入口地址被载入。CPU 在每个指令周期都会对中断请求进行采样获取，当一个中断请求产生时，如果当前执行的多周期指令未结束或者正在执行 RETI 等指令，CPU 不会立即调用相应的中断服务程序。外部中断的中断标志位在对应的中断使能有效时才会被中断源置 1，其他中断的中断标志位置 1 时无需相应中断使能有效。

16 个中断入口，每个中断可通过 IP[15:0]配置中断优先级，可以实现 2 层中断优先级，即最多可以中断嵌套 1 次。

注意：RTC 中断和 TH 中断均有多个中断源，对应的中断标志在各模块寄存器组中。

LPVT 中断是低功耗电压监测，LVR 使能有效时该功能才开启。当 VDD 从 4V/4.5V 以下跳变到 4V/4.5V 以上时产生，可以唤醒 STOP 模式。监测点由 SYSCON 中的 LPVMS 选择。

10.1 中断结构

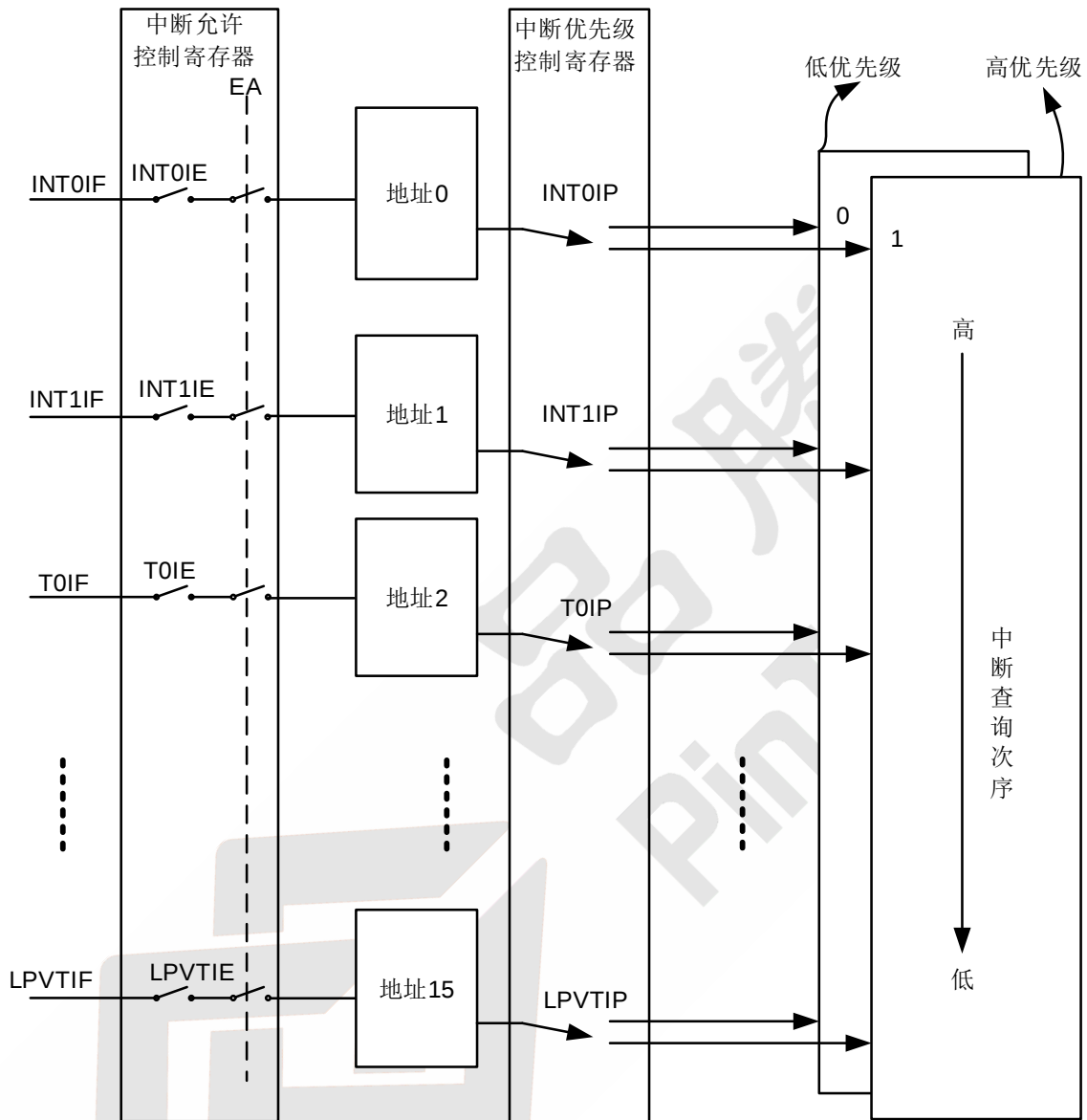


图 8 中断结构框图

全局中断使能 EA 是各个中断的总开关。当 EA 为 0 时，禁止了所有的中断，即使某个中断源使能为 1 也不会产生相应的中断，但对应标志位会置位；开启某些中断时，需要同时置 EA=1，当开启的中断信号发生时，才能产生相应中断。在上电复位后，所有中断使能被清 0。

10.2 中断优先级

中断优先级由中断入口地址、优先级等级寄存器 IP 决定。

1. IP 可配置 2 个优先级等级：0 为低优先级，1 为高优先级。
2. IP 的默认设置值为 0，当中断优先级设置相同且有不同中断请求同时发生时，硬件自动采用查询的方式处理中断服务程序：从中断入口低地址往高地址查询，即低入口地址中断优先执行。
3. 在响应低优先级中断过程中，可响应更高优先级的中断，在响应高优先级中断过程中不响应同级或低级别中断。

• 中断优先级软件配置列表

中断优先级控制寄存器	优先级	Level
0	低 ↓ 高	0
1		1

注意：软件配置中断优先级的过程中，建议关闭相应的中断以避免在修改优先级的过程中产生中断。

10.3 中断向量列表

表 9 中断向量、中断组别、中断优先级、入口、标志位列表

中断号 (C51)	优先级 配置	入口 地址	中断源	中断 使能位	相应中 断标志	标志清除 方式	能否唤醒 STOP
0	INT0IP	03h	外部中断 0	INT0IE	INT0IF	H/W Auto2	能
1	INT1IP	0Bh	外部中断 1	INT1IE	INT1IF	H/W Auto2	能
2	T0IP	13h	TIMER0 中断	T0IE	T0IF	H/W Auto1	不能
3	T1IP	1Bh	TIMER1 中断	T1IE	T1IF	H/W Auto1	不能
4	T2IP	23h	TIMER2 中断	T2IE	T2IF	H/W Auto1	不能
5	THIP	2Bh	触摸中断	THIE	THIF	必须用户清除	能
6	ADCIP	33h	ADC 中断	ADCIE	ADCIF	H/W Auto1	不能
7	I2CIP	3Bh	I2C 中断	I2CIE	I2CIF	H/W Auto1	不能
8	RTCIP	43h	RTC 中断	RTCIE	RTCIF	必须用户清除	能
9	LVDIP	4Bh	LVD 中断	LVDIE	LVDIF	H/W Auto1	不能
10	PWMIP	53h	PWM 中断	PWMIE	PWMIF	H/W Auto1	不能
11	UAT0RIP	5Bh	UART0 接收中断	UAT0RIE	UAT0RIF	H/W Auto1	不能
12	UAT1RIP	63h	UART1 接收中断	UAT1RIE	UAT1RIF	H/W Auto1	不能
13	UAT0TIP	6Bh	UART0 发送中断	UAT0TIE	UAT0TIF	H/W Auto1	不能
14	UAT1TIP	73h	UART1 发送中断	UAT1TIE	UAT1TIF	H/W Auto1	不能
15	LPVTIP	7Bh	LPVT 中断	LPVTIE	LPVTIF	H/W Auto1	能

注意：H/W Auto1 对应中断标志，没有打开相应中断使能位时也可被中断源置位，可以在程序中查询并软件清 0；H/W Auto2 对应中断标志，需要打开相应中断使能位才可被中断源置位，当 GIE 关闭时可以在程序中查询并软件清 0；H/W Auto1、H/W Auto2 在 GIE 打开且相应中断使能位打开时，标志位被中断源置位后跳转到中断入口时硬件会自动清标志位。

中断源只有在打开 GIE 并且打开相应中断使能位时才能唤醒 STOP 或 IDLE。

10.4 中断 SFR

中断模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
A8h	IE0	EA	ADCIE	THIE	T2IE	T1IE	T0IE	INT1IE	INT0IE	0000 0000
B0h	IE1	LPVTIE	UAT1TIE	UAT0TIE	UAT1RIE	UAT0RIE	PWMIE	LVDIE	RTCIE	0000 0000
B1h	IF0	I2CIF	ADCIF	-	T2IF	T1IF	T0IF	INT1IF	INT0IF	00-0 0000
B2h	IF1	LPVTIF	UAT1TIF	UAT0TIF	UAT1RIF	UAT0RIF	PWMIF	LVDIF	-	0000 000-
B6h	IPL	I2CIP	ADCIP	THIP	T2IP	T1IP	T0IP	INT1IP	INT0IP	0000 0000
B5h	IPH	LPVTIP	UAT1TIP	UAT0TIP	UAT1RIP	UAT0RIP	PWMIP	LVDIP	RTCIP	0000 0000
B3h	INTS	-				INT1S[1:0]		INT0S[1:0]		---- 0000
B4h	INTEN	-						EINT1	EINT0	---- --00
D4h	PWMIECFG	PWMIES	-			PWMIS[2:0]				0--- -000
D5h	PWMCNT	PWMCNT[7:0]								0000 0001

• 中断使能寄存器 0(IE0, A8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	EA	ADCIE	THIE	T2IE	T1IE	T0IE	INT1IE	INT0IE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- Bit[7] **EA**: 中断全局使能位
 1: 使能所有未被屏蔽的中断
 0: 禁止所有中断
- Bit[6] **ADCIE**: ADC 中断使能位
 1: 使能 ADC 中断
 0: 禁止 ADC 中断
- Bit[5] **THIE**: 触摸中断使能位
 1: 使能触摸中断
 0: 禁止触摸中断
- Bit[4] **T2IE**: TIMER2 溢出中断使能位
 1: 使能 TIMER2 溢出中断
 0: 禁止 TIMER2 溢出中断
- Bit[3] **T1IE**: TIMER1 溢出中断使能位
 1: 使能 TIMER1 溢出中断
 0: 禁止 TIMER1 溢出中断
- Bit[2] **T0IE**: TIMER0 溢出中断使能位
 1: 使能 TIMER0 溢出中断
 0: 禁止 TIMER0 溢出中断
- Bit[1] **INT1IE**: INT1 中断使能位
 1: 使能外部中断 1

0: 禁止外部中断 1

Bit[0] **INT0IE**: INT0 中断使能位

1: 使能外部中断 0

0: 禁止外部中断 0

• 中断使能寄存器 1(IE1, B0h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	LPVTIE	UAT1TIE	UAT0TIE	UAT1RIE	UAT0RIE	PWMIE	LVDIE	RTCIE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7] **LPVTIE**: 低功耗电压上跳监测中断使能位

1: 使能低功耗电压上跳监测中断

0: 禁止低功耗电压上跳监测中断

Bit[6] **UAT1TIE**: UART1 发送中断使能位

1: 使能 UART1 发送中断

0: 禁止 UART1 发送中断

Bit[5] **UAT0TIE**: UART0 发送中断使能位

1: 使能 UART0 发送中断

0: 禁止 UART0 发送中断

Bit[4] **UAT1RIE**: UART1 接收中断使能位

1: 使能 UART1 接收中断

0: 禁止 UART1 接收中断

Bit[3] **UAT0RIE**: UART0 接收中断使能位

1: 使能 UART0 接收中断

0: 禁止 UART0 接收中断

Bit[2] **PWMIE**: PWM 中断使能位

1: 使能 PWM 中断

0: 禁止 PWM 中断

Bit[1] **LVDIE**: LVD 中断使能位

1: 使能 LVD 中断

0: 禁止 LVD 中断

Bit[0] **RTCIE**: RTC 中断使能位

1: 使能 RTC 中断

0: 禁止 RTC 中断

• 中断标志寄存器 0(IF0, B1h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CIF	ADCIF	-	T2IF	T1IF	T0IF	INT1IF	INT0IF
Access	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Default	0	0	-	0	0	0	0	0

Bit[7] **I2CIF**: I2C 中断标志, 进中断后硬件自动清 0 或软件写 0 清

- Bit[6] **ADCIF**: ADC 中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[4] **T2IF**: TIMER2 中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[3] **T1IF**: TIMER1 中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[2] **T0IF**: TIMER0 中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[1] **INT1IF**: 外部中断 1 标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[0] **INT0IF**: 外部中断 0 标志, 进中断后硬件自动清 0 或软件写 0 清

• 中断标志寄存器 1(IF1, B2h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	LPVTIF	UAT1TIF	UAT0TIF	UAT1RIF	UAT0RIF	PWMIF	LVDIF	-
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Default	0	0	0	0	0	0	0	-

- Bit[7] **LPVTIF**: 低功耗电压上跳监测中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[6] **UAT1TIF**: UART1 发送中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[5] **UAT0TIF**: UART0 发送中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[4] **UAT1RIF**: UART1 接收中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[3] **UAT0RIF**: UART0 接收中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[2] **PWMIF**: PWM 中断标志, 进中断后硬件自动清 0 或软件写 0 清
- Bit[1] **LVDIF**: LVD 中断标志, 进中断后硬件件自动清 0 或软件写 0 清

• 中断优先级控制寄存器 0(IPL, B6h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CIP	ADCIP	THIP	T2IP	T1IP	T0IP	INT1IP	INT0IP
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- Bit[7] **I2CIP**: I2C 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[6] **ADCIP**: ADC 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[5] **THIP**: 触摸中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[4] **T2IP**: TIMER2 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[3] **T1IP**: TIMER1 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[2] **T0IP**: TIMER0 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[1] **INT1IP**: 外部中断 1 优先级配置, 1 为高优先级, 0 为低优先级
- Bit[0] **INT0IP**: 外部中断 0 优先级配置, 1 为高优先级, 0 为低优先级

• 中断优先级控制寄存器 1(IPH, B5h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	LPVTIP	UAT1TIP	UAT0TIP	UAT1RIP	UAT0RIP	PWMIP	LVDIP	RTCIP
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

- Bit[7] **LPVTIP**: LPVT 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[6] **UAT1TIP**: UART1 发送中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[5] **UAT0TIP**: UART0 发送中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[4] **UAT1RIP**: UART1 接收中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[3] **UAT0RIP**: UART0 接收中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[2] **PWMIP**: PWM 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[1] **LVDIP**: LVD 中断优先级配置, 1 为高优先级, 0 为低优先级
- Bit[0] **RTCIP**: RTC 中断优先级配置, 1 为高优先级, 0 为低优先级

• 外部中断触发方式选择寄存器(INTS, B3h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-				INT1S[1:0]		INT0S[1:0]	
Access	-				R/W		R/W	
Default	-				00		00	

Bit[3:2] **INT1S**: 外部中断 1 触发方式选择

- 00: 下降沿触发
- 01: 下降沿触发
- 10: 上升沿触发
- 11: 下降沿或上升沿触发

Bit[1:0] **INT0S**: 外部中断 0 触发方式选择

- 00: 下降沿触发
- 01: 下降沿触发
- 10: 上升沿触发
- 11: 下降沿或上升沿触发

• 外部中断开关寄存器(INTEN, B4h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-						EINT1	EINT0
Access	-						R/W	R/W
Default	-						0	0

Bit[1] **EINT1**: 外部中断 1 开关使能位(打开后即使不响应中断, 外部中断标志在触发后也会置 1)

- 1: 外部中断 1 使能
- 0: 外部中断 1 关闭

Bit[0] **EINT0**: 外部中断 0 开关使能位(打开后即使不响应中断, 外部中断标志在触发后也会置 1)

- 1: 外部中断 0 使能
- 0: 外部中断 0 关闭

• PWM 中断触发选择寄存器(PWMIECFG, D4h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PWMIES	-				PWMIS[2:0]		
Access	R/W	-				R/W		
Default	0	-				000		

Bit[7] **PWMIES:** PWM 中断沿触发选择

1: 下降沿触发

0: 上升沿触发

Bit[2:0] **PWMIS:** PWM 中断触发选择

000: 选择 PWM0 触发中断

001: 选择 PWM1 触发中断

010: 选择 PWM2 触发中断

011: 选择 PWM3 触发中断

100: 选择 PWM4 触发中断

101: 选择 PWM5 触发中断

其他: 选择 PWM0 触发中断

• PWM 计数寄存器(PWMCNT, D5h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PWMCNT[7:0]							
Access	R/W							
Default	0x01							

Bit[7:0] **PWMCNT:** 被选中的 PWM 产生次数寄存器，用于产生 PWM 中断。此寄存器不能配置为 0，否则将产生硬件异常。

11. 实时时钟系统(RTC)

11.1 RTC功能

实时时钟模块 (RTC)，时钟源为外接的 **32.768KHz** 晶体振荡器时钟，当不使能外部晶振或晶振停振时会自动切换使用内部 **131K** 时钟的四分频。模块包含毫秒、秒、分、时、天和星期寄存器，同时内置了闹钟功能，当 **RTC** 计时时间和设定的闹钟时间匹配时，可以产生中断，适用于时钟和闹钟功能的应用。同时模块还可以设置毫秒级中断、半秒中断，毫秒级中断的中断时间可设置。当对计时精度要求不高时，可以不使能外部晶振而使用内部 **131K** 时钟。**RTC** 中断可以唤醒 **STOP/IDLE** 模式。结构如下图所示。

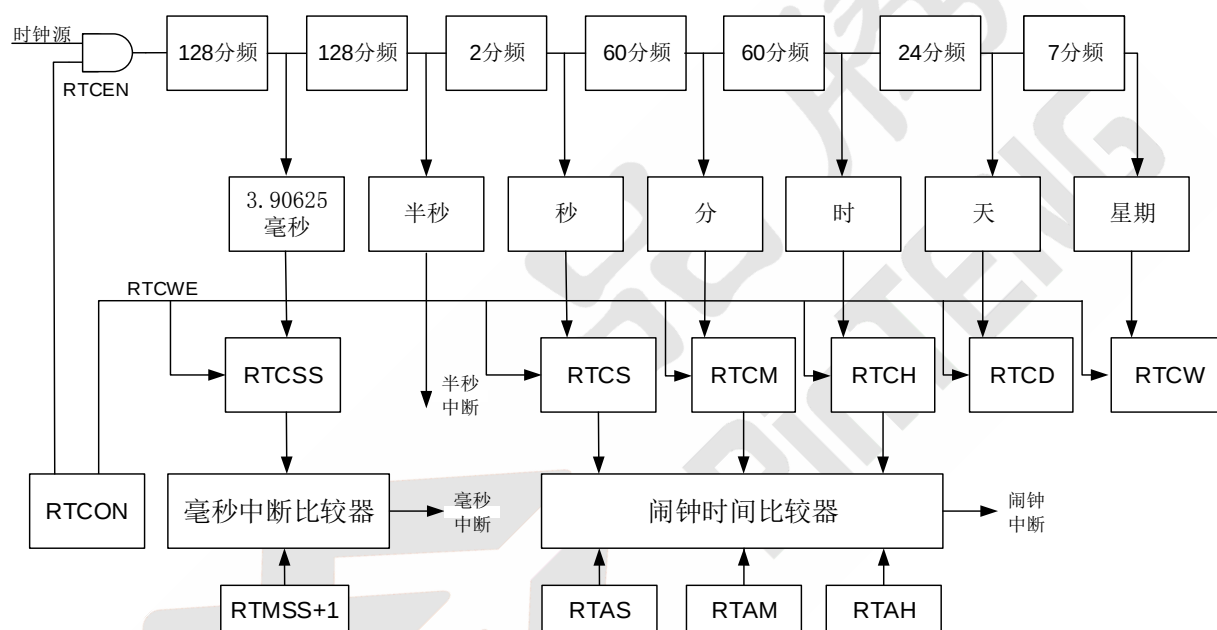


图 9 RTC 结构图

1. RTC使能和关闭

设置 RTCEN=1 后, RTC 开始计时;设置 RTCEN=0 后, RTC 模块计时寄存器的状态都保持不变。RTC 使能后写 RTC 时间寄存器需要等待 150us 读到的值才是写入值,可以通过读 RTCWE 值来判断是否写入。注意,由于 RTC 的时钟源是外挂 32.768KHz 晶振,要等待 32.768KHz 晶振正常起振(读取 SYSCON 的 XTALF 位)后再使能 RTC 模式,否则可能会计时不准确。

2. RTC寄存器读写

RTC 寄存器 (RTCS、RTCM、RTCWH、RTCW、RTCDL、RTCDH) 写入由 RTCWE 置 1 控制, 写入后需要等待一个同步时间 70us, RTCWE 才被硬件清 0, 此时 RTC 寄存器才改写成功, 期间读对应寄存器得到值不确定。(70us 指写信号同步到低速时钟的最长时间)。写入一个非法时间 (超过了有效秒、分或时范围的值) 到 RTC 寄存器将会被当作该寄存器的最大有效值写入。注意, 在 RTCWE 置 1 时, 毫秒寄存器 RTCSS 会被清零以方便精确计时。

3. RTC闹钟功能

当 RTC 计时的秒、分、时寄存器和闹钟设置时间匹配时，将产生闹钟中断，标志位为 RTCAF。用户通过寄存器 RTCAS、RTCAM、RTCAH 来设置闹铃时间，写一个非法的值(超过了有效秒、分

或时范围的值)将会当作该寄存器最大有效值来写入。用户可以设置相应的比较使能位 (HCEN、MCEN、SCEN) 去比较 RTCAS、RTCAM 和 RTCAH 寄存器的一个或多个值, 只要有一个比较使能位有效就开启了闹钟功能。比较使能位设为 0, 其相应的时间项比较会被忽略 (例如设置 HCEN=0、MCEN=1、SCEN=1, 只会比较分、秒寄存器, 小时寄存器会认为已匹配, 每小时会发生一次闹钟中断)。这样就允许闹铃可以在一天的特定时间发生一次 (所有的比较位使能), 或者周期性的一每小时一次、每分钟一次 (通过比较使能位的组合设置来实现)。

11.2 RTC SFR

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
D8h	RTCCON	RTCEN	MSEN	HSEN	SCEN	MCEN	HCEN	RTCWE	CALS	0000 0000
D9h	RTCSS	RTCSS[7:0]								0000 0000
DAh	RTCS	-		RTCS[5:0]						--00 0000
DBh	RTCM	-		RTCM[5:0]						--00 0000
DCh	RTCWH	RTCW[2:0]			RTCH[4:0]					0000 0000
DDh	RTCDL	RTCDL[7:0]								0000 0000
DEh	RTCDH	RTCDH [7:0]								0000 0000
DFh	RTCAS	-		RTCAS[5:0]						--00 0000
E1h	RTCAM	-		RTCAM[5:0]						--00 0000
E2h	RTCAH	-			RTCAH[4:0]					---0 0000
E3h	RTCMSS	RTCMSS[7:0]								0000 0000
E4h	RTCCAL	OPTRC131K[7:0]								1000 0000
F8h	RTCIF	-					RTCAF	RTCHF	RTCMF	---- -000

● RTC 控制寄存器(RTCCON, D8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCEN	MSEN	HSEN	SCEN	MCEN	HCEN	RTCWE	CALS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7] **RTCEN**: RTC 时钟使能, 1 有效

Bit[6] **MSEN**: 毫秒中断使能, 1 有效

Bit[5] **HSEN**: 半秒中断使能, 1 有效

Bit[4] **SCEN**: 闹钟秒比较使能, 1 有效

Bit[3] **MCEN**: 闹钟分钟比较使能, 1 有效

Bit[2] **HCEN**: 闹钟小时比较使能, 1 有效

Bit[1] **RTCWE**: 时间配置值写入, 软件写 1 启动, 配置值写入后自动清 0

Bit[0] **CALS**: 自动校准周期选择

1: 使能自动校准后每分钟校准一次

0: 使能自动校准后每秒钟校准一次

注意: 每校准一次, RC131K 校准值加 1 或减 1。校准到 1%精度停止。

● RTC 中断标志寄存器(RTCIF, F8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-----	------	------	------	------	------	------	------	------

Name	-	RTCAF	RTCHF	RTCMF
Access	-	R/W	R/W	R/W
Default	-	0	0	0

Bit[2] **RTCAF**: RTC 闹钟中断标志, 软件写 0 清, 写 1 无影响

Bit[1] **RTCHF**: RTC 半秒中断标志, 软件写 0 清, 写 1 无影响

Bit[0] **RTCMF**: RTC 毫秒中断标志, 软件写 0 清, 写 1 无影响

● **RTC 毫秒计数器(RTCSS, D9h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCSS[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **RTCSS**: RTC 毫秒计数器, 每 1/256 秒加 1

● **RTC 秒计数器(RTCS, DAh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	RTCS[5:0]						
Access	-	R/W						
Default	-	0x00						

Bit[5:0] **RTCS**: 秒计数器, 每 1 秒加 1, 计数范围为 0~59

● **RTC 分钟计数器(RTCM, DBh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	RTCM[5:0]						
Access	-	R/W						
Default	-	0x00						

Bit[5:0] **RTCM**: 分计数器, 每 1 分加 1, 计数范围为 0~59

● **RTC 小时寄存器(RTCWH, DCh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCW[2:0]			RTCH[4:0]				
Access	R/W			R/W				
Default	000			0x00				

Bit[7:5] **RTCW**: 星期计数器, 计数范围为 1~7, 代表星期一到星期日, 当设置为 0 时, 星期计数功能关闭保持为 0

Bit[4:0] **RTCH**: 小时计数器, 每小时加 1, 计数范围为 0~23

● **RTC 天数寄存器(RTCDL, DDh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCDL[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **RTCDL**: 天数计数器低 8 位

● **RTC 天数寄存器(RTCDH, DEh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCDH[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **RTCDH**: 天数计数器高 8 位, {RTCDH, RTCDL}组成天数计数器, 每天加 1

● **RTC 闹钟秒值设置寄存器(RTCAS, DFh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		RTCAS[5:0]					
Access	-		R/W					
Default	-		0x00					

Bit[5:0] **RTCAS**: 闹钟秒值设置, 计数范围为 0~59

● **RTC 闹钟分钟设置寄存器(RTCAM, E1h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		RTCAM[5:0]					
Access	-		R/W					
Default	-		0x00					

Bit[5:0] **RTCAM**: 闹钟分钟值设置, 计数范围为 0~59

● **RTC 闹钟小时设置寄存器(RTCAH, E2h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			RTCAH[4:0]				
Access	-			R/W				
Default	-			0x00				

Bit[4:0] **RTCAH**: 闹钟小时值设置, 计数范围为 0~23

● **RTC 毫秒中断阈值寄存器(RTCMSS, E3h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	RTCMSS[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **RTCMSS**: RTC 毫秒中断阈值寄存器, 毫秒中断时间=(RTCMSS+1)*128*RTC 时钟周期。
如果 RTC 时钟为 32.768KHz,则设置的时间单位为: $128 * (1/32.768) = 3.90625\text{ms}$

● **RTC 校准 131K 校准值寄存器(RTCCAL, E4h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	OPTRC131K[7:0]							
Access	R/W							
Default	0x80							

Bit[7:0] **OPTRC131K[7:0]**: 读该寄存器为 131K 校准值。LRCCAL 为 0 时, 写该寄存器会改写 131K 校准值。

11.3 RTC应用说明

- 1) RTC 默认使用外部 32.768K 晶振时钟, 当外部晶振还没有开启或停振时自动切换到内部低速时钟, 可通过 SYSCON 中的 XTALF 查询。低速时钟使能不能关闭。
- 2) SYSCON 中 LRCCAL 置 1, 可开启外部晶振校准内部低速时钟, RTCCON 中 CALS 选择校准频率。STOP 模式下校准也正常进行。这样在外部晶振停振切换到内部低速时钟计时的情况下, 仍然可以较准确的计时。
- 3) 应用需要长期低功耗计时且接入稳压电源能自动恢复正常工作时, 可以在进 STOP 前开启 LPVT 中断, 这样不需要频繁的唤醒 MCU 进行电压检测, 也可以在芯片从低压切换到稳压源时唤醒 MCU。

11.4 RTC配置流程

一、RTC 日历配置流程

- 1) 使能 RTC (RTCEN)
- 2) 配置 RTCSS、RTCS、RTCM、RTCWH、RTCDL、RTCDH、RTCW
- 3) RTCWE 位置 1, 读 RTCWE 为 0 表示配置已经写入

二、RTC 闹钟配置流程

- 1) 按日历流程配置完后，开始配置 RTCAS、RTCAM、RTCAH。
- 2) 根据需要配置 SCEN、MCEN、HCEN。
- 3) 使能 IE1 中的 RTCIE，时间匹配时将产生 RTC 中断，中断程序中需要软件清 RTCAF 标志位
- 4) 不使能 RTCIE，不会进中断，可以查询 RTCIF 中 RTCAF 标志位来确定是否时间匹配

三、RTC 毫秒基准计时配置流程

- 1) 使能 RTC (RTCEN)，日历流程第 2 步可设可不设，不影响以下步骤
- 2) 根据需要配置 RTMCSS，毫秒中断使能 (MSEN) 置 1。
- 3) 使能 IE1 中的 RTCIE，每间隔设置时间将产生 RTC 中断，中断程序中需要软件清 RTCMF 标志位
- 4) 不使能 RTCIE，不会进中断，可以查询 RTCIF 中 RTCMF 标志位来确定是否达到间隔时间

四、RTC 半秒基准计时配置流程

- 1) 使能 RTC (RTCEN)，日历流程第 2 步可设可不设，不影响以下步骤
- 2) 半秒中断使能 (HSEN) 置 1。
- 3) 使能 IE1 中的 RTCIE，每间隔半秒将产生 RTC 中断，中断程序中需要软件清 RTCHF 标志位
- 4) 不使能 RTCIE，不会进中断，可以查询 RTCIF 中 RTCHF 标志位来确定是否达到间隔时间

12. 定时器(TIMER)

共包括 3 个 TIMER，其功能分别如下：

TIMER名称	位宽	定时功能	计数功能	PWM功能
TIMER0	8	Y	Y	N
TIMER1	16	Y	N	Y
TIMER2	16	Y	N	Y

12.1 TIMER0

TIMER0 为 8 位向上定时器，其从 T0OVR 开始计数，当其计数值达到 0xFF 后，产生 T0 溢出信号。特性如下：

1. 支持时钟预分频功能；
2. 可配置选择内部或者外部时钟源；
3. 定时功能；
4. 计数功能。

TIMER0 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
9Ah	T0CON	T0M	T0EG	-		T0FS[2:0]			T0EN	10-- 0000
9Bh	T0C	T0C[7:0]								0000 0000
9Ch	T0OVR	T0OVR[7:0]								0000 0000

• **TIMER0 控制寄存器(T0CON, 9Ah)：**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T0M	T0EG	-		T0FS[2:0]			T0EN
Access	R/W	R/W	-		R/W			R/W
Default	1	0	-		000			0

Bit[7] **T0M**：TIMER0 模式选择位

0：计数器模式，T0 作为计数时钟

1：定时器模式

Bit[6] **T0EG**：计数器的有效沿选择位

1：下降沿计数

0：上升沿计数

Bit[3:1] **T0FS**：TIMER0 工作于定时模式时时钟分频选择

000：不分频

001：2 分频

010：4 分频

011：8 分频

100：16 分频

101：32 分频

110: 64 分频

111: 128 分频

Bit[0] **T0EN**: TIMER0 使能位

1: 使能 TIMER0

0: 禁止 TIMER0

• 定时器计数值(T0C, 9Bh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T0C[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **T0C[7:0]**: TIMER0 计数值 8 位

• 定时器计数周期值(T0OVR, 9Ch):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T0OVR[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **T0OVR[7:0]**: 预设置的定时周期寄存器，实际定时周期为: **0xFF - T0OVR + 1**

注意: T0OVR 不能配置为 8'hFF

12.1.1 定时模式

TIMER0 能被配置为普通的定时模式，当 TIMER0 被启动后，定时周期寄存器 T0OVR 的值将会被装载到定时器中，当定时器的计数器计满后将产生 TIMER0 溢出信号，同时 T0OVR 将会由硬件重新自动装载到计数器中。

TIMER0 工作于定时模式时的配置流程如下：

1. 配置 T0M 为定时器模式；
2. 配置 T0OVR；
3. 使能 TIMER0；
4. 计满溢出后，产生溢出信号。

12.1.2 计数模式

TIMER0 能被配置为计数模式，其采样 T0 的上升沿或者下降沿(T0EG 配置上升沿或下降沿的触发方式)，从 T0OVR 开始向上计数。

TIMER0 工作于计数模式时的配置流程如下：

1. 设置 T0M 为计数器模式；
2. 设置 T0EG，配置计数器边沿触发方式；
3. 设置 T0OVR，选择计数器初值；
4. 使能 TIMER0；
5. 计数溢出后，产生溢出信号。

12.2 TIMER1

TIMER1 为 16 位向上定时器，其从 T1OVR 开始计数，当其计数值达到 0xFFFF 后，产生 T1 溢出信号。其特性如下：

1. 支持时钟预分频功能；
2. 定时功能；

TIMER1 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
9Dh	T1CON	-				T1FS[2:0]			T1EN	---- 0000
9Eh	T1OVR	T1OVR[7:0]								0000 0000
9Fh	T1OVRH	T1OVRH[7:0]								0000 0000
A1h	T1CL	T1CL[7:0]								0000 0000
A2h	T1CH	T1CH[7:0]								0000 0000

• **TIMER1 控制寄存器(T1CON, 9Dh)：**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-				T1FS[2:0]			T1EN
Access	-				R/W			R/W
Default	-				000			0

Bit[3:1] **T1FS**：定时器 1 时钟分频选择

- 000：不分频
- 001：2 分频
- 010：4 分频
- 011：8 分频
- 100：16 分频
- 101：32 分频
- 110：64 分频
- 111：128 分频

Bit[0] **T1EN**：TIMER1 使能位

- 1：使能 TIMER1
- 0：禁止 TIMER1

• **TIMER1 预设置的周期寄存器(T1OVR, 9Eh)：**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T1OVR[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **T1OVR[7:0]**：TIMER1 预设置的周期寄存器低 8 位

• TIMER1 预设置的周期寄存器(T1OVRH, 9Fh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T1OVRH[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **T1OVRH[7:0]**: TIMER1 预设置的周期寄存器高 8 位

实际周期为: $0xFFFF - \{T1OVRH, T1OVRH\} + 1$

注意: $\{T1OVRH, T1OVRH\}$ 不能配置为 16'hFFFF

• TIMER1 计数寄存器(T1CL, A1h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T1CL[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **T1CL[7:0]**: TIMER1 计数值低 8 位

• TIMER1 计数寄存器(T1CH, A2h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T1CH[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **T1CH[7:0]**: TIMER1 计数值高 8 位。实际计数值为: $\{T1CH, T1CL\}$

TIMER1 能被配置为普通的定时模式, 当 TIMER1 被启动后, 定时周期寄存器的值将会被装载到定时器中, 当定时器的计数器计满后将产生 TIMER1 溢出信号, 同时周期寄存器将会由硬件重新自动装载到计数器中, 然后启动向上计数。

TIMER1 工作于定时模式时的配置流程如下:

1. 配置 T1OVRH, T1OVRH;
2. 使能 TIMER1;
3. 计满溢出后, 硬件产生 T1 溢出信号。

12.3 TIMER2

TIMER2 为 16 位向上定时器，其从 T2OVR 开始计数，当其计数值达到 0xFFFF 后，产生 T2 溢出信号。其特性如下：

1. 支持时钟预分频功能；
2. 16 位定时功能；

TIMER2 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
A3h	T2CON	-				T2FS[2:0]			T2EN	---- 0000
A4h	T2CL	T2CL[7:0]								0000 0000
A5h	T2CH	T2CH[7:0]								0000 0000
A6h	T2OVR	T2OVR[7:0]								0000 0000
A7h	T2OVRH	T2OVRH[7:0]								0000 0000

• TIMER2 控制寄存器(T2CON, A3h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-				T2FS[2:0]			T2EN
Access	-				R/W			R/W
Default	-				000			0

Bit[3:1] **T2FS**: TIMER2 工作于定时模式时时钟分频选择

- 000: 不分频
- 001: 2 分频
- 010: 4 分频
- 011: 8 分频
- 100: 16 分频
- 101: 32 分频
- 110: 64 分频
- 111: 128 分频

Bit[0] **T2EN**: TIMER2 使能位

- 1: 使能 TIMER2
- 0: 禁止 TIMER2

• 定时器计数值(T2CL, A4h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T2CL[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **T2CL[7:0]**: TIMER2 计数值低 8 位

• 定时器计数值(T2CH, A5h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T2CH[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **T2CH[7:0]**: TIMER2 计数值高 8 位

• 定时器计数周期值(T2OVRH, A6h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T2OVRH[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **T2OVRH[7:0]**: 预设置的定时周期寄存器低 8 位

• 定时器计数周期值(T2OVRH, A7h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	T2OVRH[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **T2OVRH[7:0]**: 预设置的定时周期寄存器高 8 位

实际定时/计数周期为: $0xFFFF - \{T2OVRH, T2OVRH\} + 1$

注意: {T2OVRH, T2OVRH}不能配置为 16'hFFFF

TIMER2 能被配置为普通的 16 位定时模式, 当 TIMER2 被启动后, 定时周期寄存器 {T2OVRH, T2OVRH} 的值将会被装载到定时器中, 当定时器的计数器计满后将产生 TIMER2 溢出信号, 同时 {T2OVRH, T2OVRH} 将会由硬件重新自动装载到计数器中。

TIMER2 工作于定时模式时的配置流程如下:

1. 配置 T2OVRH, T2OVRH;
2. 使能 TIMER2;
3. 计满溢出后, 产生溢出信号。

12.4 PWM

共包括 6 路 PWM, 其中:

- (1) PWM0~PWM2 为 TIMER1 产生;
- (2) PWM3~PWM5 可配置选择是 TIMER1 或 TIMER2 产生。

12.4.1 PWM SFR

PWM 模块相关寄存器:

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
FFEDh	PWMCON0	-			PWM3C	PWM4C	PWM5C	-		---0 00--
FFEEh	PWMCON1	-		PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	--00 0000
FFEFh	PWMCON2	-		PWM5S	PWM4S	PWM3S	PWM2S	PWM1S	PWM0S	--00 0000
FFF0h	PWMD0L	PWMD0L[7:0]								xxxx xxxx
FFF1h	PWMD0H	PWMD0H[7:0]								xxxx xxxx
FFF2h	PWMD1L	PWMD1L[7:0]								xxxx xxxx
FFF3h	PWMD1H	PWMD1H[7:0]								xxxx xxxx
FFF4h	PWMD2L	PWMD2L[7:0]								xxxx xxxx
FFF5h	PWMD2H	PWMD2H[7:0]								xxxx xxxx
FFF6h	PWMD3L	PWMD3L[7:0]								xxxx xxxx
FFF7h	PWMD3H	PWMD3H[7:0]								xxxx xxxx
FFF8h	PWMD4L	PWMD4L[7:0]								xxxx xxxx
FFF9h	PWMD4H	PWMD4H[7:0]								xxxx xxxx
FFFAh	PWMD5L	PWMD5L[7:0]								xxxx xxxx
FFFBh	PWMD5H	PWMD5H[7:0]								xxxx xxxx

• PWM 控制寄存器 0(PWMCON0, FFEDh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			PWM3C	PWM4C	PWM5C	-	
Access	-			R/W	R/W	R/W	-	
Default	-			0	0	0	-	

Bit[4] **PWM3C:** PWM3 产生源选择

- 1: TIMER2 产生
- 0: TIMER1 产生

Bit[3] **PWM4C:** PWM4 产生源选择

- 1: TIMER2 产生
- 0: TIMER1 产生

Bit[2] **PWM5C:** PWM5 产生源选择

- 1: TIMER2 产生
- 0: TIMER1 产生

• PWM 控制寄存器 1(PWMCON1, FFEEh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN
Access	-		R/W	R/W	R/W	R/W	R/W	R/W
Default	-		0	0	0	0	0	0

Bit[5:0] **PWMxEN**: PWMx 使能位(x=0~5)

1: 使能 PWMx

0: 关闭 PWMx

• PWM 控制寄存器 2(PWMCON2, FFEFh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		PWM5S	PWM4S	PWM3S	PWM2S	PWM1S	PWM0S
Access	-		R/W	R/W	R/W	R/W	R/W	R/W
Default	-		0	0	0	0	0	0

Bit[5:0] **PWMxS**: PWMx 输出有效电平选择位(x=0~5)

1: TIMER1/2 计数值小于等于 PWMDx 值时输出低电平, 大于则输出高电平

0: TIMER1/2 计数值小于等于 PWMDx 值时输出高电平, 大于则输出低电平

• PWM 占空比设置寄存器(PWMDxL, FFF0h/FFF2h/FFF4h/ FFF6h/ FFF8h/ FFFAh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PWMDxL[7:0]							
Access	R/W							
Default	0xxx							

Bit[7:0] **PWMDxL[7:0]**: PWMx 占空比设置低 8 位

• PWM 占空比设置寄存器(PWMDxH, FFF1h/FFF3h/FFF5h/FFF7h/FFF9h/FFFBh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PWMDxH[7:0]							
Access	R/W							
Default	0xxx							

Bit[7:0] **PWMDxH[7:0]**: PWMx 占空比设置高 8 位

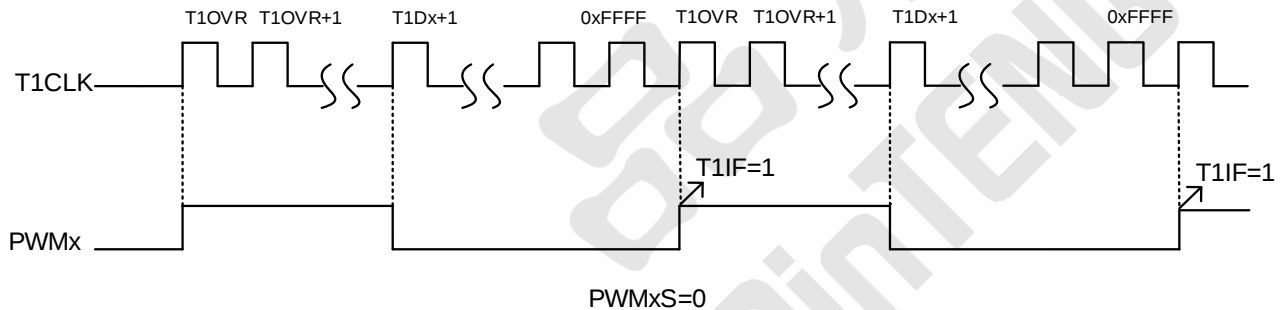
12.4.2 PWM配置流程

PWM 配置流程如下：

1. 配置 PWMx 对应 IO 口为输出并关闭开漏使能
2. 配置 PWMxC
3. 配置 PWMx 对应 TIMER 的 TxOVRH,TxOVR
4. 配置 PWMDxH,PWMDXL;
5. 配置 PWMxS;
6. 配置 PWMIS 及 PWMCNT;
7. 使能 PWMxEN;
8. 使能 TIMER1 或 TIMER2;

如下为选择 TIMER1 产生 PWMx 为例,周期为(0xFFFF - T1OVR+1),PWMx 高电平脉宽为 PWMDx。

PWMx 波形输出如下:



13. 通用输入输出端口(GPIO)

共有 4 组 26 个 GPIO 口，其中 P0、P1、P2 口都为 8 脚 I/O，P3 口为 3 脚 I/O。

13.1 GPIO SFR

GPIO 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
88h	P0	P07	P06	P05	P04	P03	P02	P01	P00	xxxx xxxx
89h	P0OD	P0OD7	P0OD6	P0OD5	P0OD4	P0OD3	P0OD2	P0OD1	P0OD0	0000 0000
8Ah	P0PH	P0PH7	P0PH6	P0PH5	P0PH4	P0PH3	P0PH2	P0PH1	P0PH0	1111 1111
8Bh	P0PD	P0PD7	P0PD6	P0PD5	P0PD4	P0PD3	P0PD2	P0PD1	P0PD0	0000 0000
8Ch	P0OE	P0OE7	P0OE6	P0OE5	P0OE4	P0OE3	P0OE2	P0OE1	P0OE0	1111 1111
D6h	P0WK	P0WK7	P0WK6	P0WK5	P0WK4	P0WK3	P0WK2	P0WK1	P0WK0	0000 0000
90h	P1	P17	P16	P15	P14	P13	P12	P11	P10	xxxx xxxx
8Dh	P1OD	P1OD7	P1OD6	P1OD5	P1OD4	P1OD3	P1OD2	P1OD1	P1OD0	0000 0000
8Eh	P1PH	P1PH7	P1PH6	P1PH5	P1PH4	P1PH3	P1PH2	P1PH1	P1PH0	1111 1111
8Fh	P1PD	P1PD7	P1PD6	P1PD5	P1PD4	P1PD3	P1PD2	P1PD1	P1PD0	0000 0000
91h	P1OE	P1OE7	P1OE6	P1OE5	P1OE4	P1OE3	P1OE2	P1OE1	P1OE0	1111 1111
98h	P2	P27	P26	P25	P24	P23	P22	P21	P20	xxxx xxxx
D7h	P2OD	P2OD7	P2OD6	P2OD5	P2OD4	P2OD3	P2OD2	P2OD1	P2OD0	0000 0000
93h	P2PH	P2PH7	P2PH6	P2PH5	P2PH4	P2PH3	P2PH2	P2PH1	P2PH0	1111 1111
94h	P2PD	P2PD7	P2PD6	P2PD5	P2PD4	P2PD3	P2PD2	P2PD1	P2PD0	0000 0000
95h	P2OE	P2OE7	P2OE6	P2OE5	P2OE4	P2OE3	P2OE2	P2OE1	P2OE0	1111 1111
A0h	P3	-					P32	P31	P30	---- -xxx
96h	P3OD	-					P3OD2	P3OD1	P3OD0	---- -000
AFh	P3PH	-					P3PH2	P3PH1	P3PH0	---- -111
97h	P3PD	-					P3PD2	P3PD1	P3PD0	---- -000
99h	P3OE	-					P3OE2	P3OE1	P3OE0	---- -111
86h	IOSET	IOPHS	IOPDS	OLDIS[1:0]		PUDIS[2:0]		VTHS		0010 1000

说明：

下述对于寄存器 Px,PxOD,PxPH,PxPD,PxOE,PxPHS,PxPDS 的描述中：

当 x=0,1,2 时，相应寄存器为 8 位，即 P0/1/2 口都分别有 8 个 GPIO 口；

当 x=3 时，相应寄存器为 3 位，即 P3 口只有 3 个 GPIO 口。

• Px 端口读写数据寄存器(Px, 88h/90h/98h/A0h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	x	x	x	x	x	x	x	x

Bit[7:0] Px[i]: Px[i]端口读写数据

读该寄存器：如果是做输入时，读的数据是外部输入；如果是做输出时，读的数据是 Px 寄存器值写该端口，为输出模式时写数据从 PAD 输出

• **Px 开漏使能寄存器(PxOD, 89h/8Dh/D7h/96h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PxOD7	PxOD6	PxOD5	PxOD4	PxOD3	PxOD2	PxOD1	PxOD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7:0] **PxOD[i]**: Px[i]开漏使能

1: 使能

0: 禁止

• **Px 上拉电阻使能寄存器(PxPH, 8Ah/8Eh/93h/AFh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PxPH7	PxPH6	PxPH5	PxPH4	PxPH3	PxPH2	PxPH1	PxPH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

Bit[7:0] **PxPH[i]**: Px[i]内部上拉使能

1: 禁止

0: 使能

• **Px 下拉电阻使能寄存器(PxPD, 8Bh/8Fh/94h/97h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PxPD7	PxPD6	PxPD5	PxPD4	PxPD3	PxPD2	PxPD1	PxPD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7:0] **PxPD[i]**: Px[i]的下拉电阻使能

1: 使能下拉电阻

0: 禁止下拉电阻

• **Px I/O 方向控制寄存器(PxOE, 8Ch/91h/95h/99h):**

PxOE 设为“1”表示该脚为输入（高阻抗），设为“0”表示该脚为输出，PxOE 寄存器可读写，系统复位以后设置为输入（高阻抗）。

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PxOE7	PxOE6	PxOE5	PxOE4	PxOE3	PxOE2	PxOE1	PxOE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

• P0 P0 口唤醒使能寄存器(P0WK, D6h):

P0WK[i]设为“1”表示 P0.i 的变化可以把 MCU 从 STOP 或 IDLE 模式唤醒

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	P0WK7	P0WK6	P0WK5	P0WK4	P0WK3	P0WK2	P0WK1	P0WK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

• IO 设置寄存器(IOSET, 86h)

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	IOPHS	IOPDS	OLDIS[1:0]		PUDIS[2:0]			VTHS
Access	R/W	R/W	R/W		R/W			R/W
Default	0	0	10		100			0

Bit[7] **IOPHS**: I/O 上拉电阻选择

1: IO 选择弱上拉电阻 37.5K

0: IO 选择强上拉电阻 25K

Bit[6] **IOPDS**: I/O 下拉电阻选择

1: IO 选择弱下拉电阻 37.5K

0: IO 选择强下拉电阻 25K

Bit[5:4] **OLDIS[1:0]**: I/O 口灌电流配置

00: 驱动电流为 5mA

01: 驱动电流为 15mA

10: 驱动电流为 60mA

11: 驱动电流为 120mA

Bit[3:1] **PUDIS[2:0]**: I/O 口拉电流配置

000: 驱动电流为 2mA

001: 驱动电流为 5mA

010: 驱动电流为 8mA

011: 驱动电流为 11mA

100: 驱动电流为 14mA

101: 驱动电流为 20mA

110: 驱动电流为 30mA

111: 驱动电流为 100mA

Bit[0] **VTHS**: I/O 输入电压门限选择

1: 输入高电压=0.50VDD, 输入低电压=0.2VDD

0: 输入高电压=0.70VDD, 输入低电压=0.3VDD

13.2 IO配置说明

表 10 IO 配置说明

模式	数据	开漏	上拉	下拉	说明
输入	0	无效	1	1	输出中间电平，读为0(为避免漏电流，数字输入关闭)
输入	0/1	无效	0	0/1	内部读为端口实际状态
输入	0/1	无效	0/1	0	内部读为端口实际状态
输出	0/1	0	自动关闭	自动关闭	依据内部输出输出0/1，读为寄存器的值
输出	1	1	1	1	输出中间电平(为避免漏电流，数字输入关闭)，读为寄存器值
输出	1	1	0	0	输出高阻，读为寄存器的值
输出	1	1	0	1	输出低电平，读为寄存器的值
输出	1	1	1	0	输出高电平，读为寄存器的值
输出	0	1	0/1	0/1	输出为0，读为寄存器的值

表 11 IO 复用功能对 IO 配置的影响

功能	优先级	高	中	低	备注
OSCI/O		√	-	-	P26、P27口会自动切换为模拟口，且自动屏蔽ADC、CDC功能
OCDS		√	-	-	硬件自动切换方向，下拉、开漏固定关闭，且自动屏蔽ADC、CDC功能
RST		√	-	-	硬件自动切换，P00固定为数字输入口，上拉开启、下拉关闭，且自动屏蔽ADC、CDC功能
ADC		-	√	-	配置（ADCCHx）后对应端口会自动切换为模拟IO口（输入口、上下拉关闭、数字输入关闭）
CDC		-	√	-	CDC配置的IO口只在其扫描时自动切换为模拟口，其它时刻受IO口配置控制
UART		-	-	√	软件根据需要配置为输入/出，不会自动切换功能
I2C		-	√	-	使能有效后自动切换输入/出，但需软件配置使能OD及上拉功能
PWM		-	-	√	软件配置为输出，不会自动切换功能
INT		-	-	√	软件配置输入，不会自动切换功能
WAKE		-	-	√	软件配置为输入，不会自动切换功能
GPIO		-	-	√	根据需要配置，不会自动切换

- P0/P1/P2/P3 由相应的上拉/下拉控制位(P0PH/D、P1PH/D、P2PH/D、P3PH/D 寄存器)来使能内部上/下拉。
 - IO 设置为输出模式，开漏不使能则内部上/下拉功能自动关闭
 - IO 设置为开漏使能且输出数据为 1 时，上/下拉功能有效，此时同时开启上下拉选择不同强弱上下拉电阻时可输出不同的中间电压，不开启则输出高阻。上下拉同时开启时会把数字输入关闭，避免出现漏电流
 - P0/P1/P2/P3 配置为输入模式，同时开启上拉、下拉使能可以输出中间电压。选择不同强弱上下拉电阻，可以输出不同电压。上下拉同时开启时会把数字输入关闭，避免出现漏电流
- ADC 功能开启时，对应 IO 在其配置寄存器（ADCCHx）中设置后，会自动切换为模拟 IO 口（为输入口、上下拉关闭、数字输入关闭）

3. CDC 配置的 IO 口只在其扫描时自动切换为模拟口，其它时刻受 IO 口配置控制
4. 外部晶振在 OPTION 中使能后，P26、P27 口会自动切换为模拟口，且自动屏蔽 CDC 功能
5. LED、PWM、UART、IO 口操作、计数、外部中断这些数字功能，使用到的 IO 口要根据需要配置方向及初始值，不会自动切换
6. 进入在线仿真模式后，其选择的两个仿真 IO 口，PCL 固定为数字输入口，下拉关闭；PDA 固定为数字输入输出口，方向根据协议自动由硬件切换，下拉、开漏固定关闭，复用功能开启时自动屏蔽 ADC、CDC 功能
7. P00 选择为外部复位脚选择后固定为数字输入口，上拉开启、下拉关闭，复用功能开启时自动屏蔽 ADC、CDC 功能
8. P01 口作 ADC 内部参考输出或 ADC 参考电压输入或 CDC 的 CMOD 脚时，建议把 CMODFS 位置 1，否则 P01 会有漏电流
9. GPIO 口只选择 ADC、CDC 一种功能，否则会互相影响而导致性能变差

14. 端口资源映射

支持以下端口资源映射(MAP)方式:

1. T0 可映射到 P05、P06、P07 口输入。
2. INT[i] (i=0~1)可映射到所有 IO 口输入。
3. PWM0 及 PWM1 可映射到所有非 PWM 的 IO 口输出。
4. UART1_TX 可映射到 P32 端口, UART1_RX 可映射到 P02 端口。
5. I2C_SDA 可映射到 P31 端口, I2C_SCL 可映射到 P30 端口。

资源映射模块相关寄存器:

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
FFE8h	MAP0	-		I2CSDAM	I2CSCLM	UAT1RXM	UAT1TXM	T0M[1:0]		--00 0000
FFE9h	MAP1	-		INT0M[4:0]						---0 0101
FFEAh	MAP2	-		INT1M[4:0]						---0 0110
FFEBh	MAP3	-		PWM0M[4:0]						---0 0111
FFECb	MAP4	-		PWM1M[4:0]						---0 0111

● 资源映射寄存器 0(MAP0, FFE8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		I2CSDAM	I2CSCLM	UAT1RXM	UAT1TXM	T0M[1:0]	
Access	-		R/W	R/W	R/W	R/W	R/W	
Default	-		0	0	0	0	00	

Bit[5] **I2CSDAM:**

- 1: I2C_SDA 映射到 P31 端口
- 0: I2C_SDA 默认在 P03 端口

Bit[4] **I2CSCLM:**

- 1: I2C_SCL 映射到 P30 端口
- 0: I2C_SCL 默认在 P04 端口

Bit[3] **UAT1RXM:**

- 1: UART1_RX 映射到 P02 端口
- 0: UART1_RX 默认在 P04 端口

Bit[2] **UAT1TXM:**

- 1: UART1_TX 映射到 P32 端口
- 0: UART1_TX 默认在 P03 端口

Bit[1:0] **T0M[1:0]:** T0 输入 IO 口按如下方式配置:

T0M[1:0]	IO口
00	P05
01	P06
10	P07
11	P05

● 资源映射寄存器 1/2(MAP1/2, FFE9h/FFEAh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			INT[i]M[4:0]				
Access	-			R/W				
Default	-			0x05/0x06				

Bit[4:0] INT[i]M: INT[i]输入 IO 口按如下方式配置:

INT[i]M	IO口	INT[i]M	IO口	INT[i]M	IO口	INT[i]M	IO口
0x00	P00	0x08	P10	0x10	P20	0x18	P30
0x01	P01	0x09	P11	0x11	P21	0x19	P31
0x02	P02	0x0A	P12	0x12	P22	0x1A	P32
0x03	P03	0x0B	P13	0x13	P23	Others	无外部中断
0x04	P04	0x0C	P14	0x14	P24		
0x05	P05	0x0D	P15	0x15	P25		
0x06	P06	0x0E	P16	0x16	P26		
0x07	P07	0x0F	P17	0x17	P27		

说明:

1. 上述 i=0~1, 即两路 INT
2. 上述 INT[i]M 的 Default 值说明:
 - 1) INT0M 的 Default 值为 0x05, 即 INT0 默认从 P05 口输入
 - 2) INT1M 的 Default 值为 0x06, 即 INT1 默认从 P06 口输入
3. 上述表格中 Others (超出有效配置范围) 时无外部中断功能
4. 若软件错误地配置 INT0~INT1 从同一个 IO 口输入时, 此 IO 口可同时用于 INT0~INT1。

资源映射寄存器 3(MAP3, FFEbH):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			PWM0M[4:0]				
Access	-			R/W				
Default	-			0x07				

Bit[4:0] **PWM0M**: PWM0 输出 IO 口按如下方式配置:

PWM0M	IO口	PWM0M	IO口	PWM0M	IO口
0x00	P00	0x08	P11	0x10	P25
0x01	P01	0x09	P12	0x11	P26
0x02	P02	0x0A	P13	0x12	P27
0x03	P03	0x0B	P20	0x13	P30
0x04	P04	0x0C	P21	0x14	P31
0x05	P05	0x0D	P22	0x15	P32
0x06	P06	0x0E	P23	Others	P07
0x07	P07	0x0F	P24		

● **资源映射寄存器 4(MAP4, FFEcH):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			PWM1M[4:0]				
Access	-			R/W				
Default	-			0x07				

Bit[4:0] **PWM1M**: PWM1 输出 IO 口按如下方式配置:

PWM1M	IO口	PWM1M	IO口	PWM1M	IO口
0x00	P00	0x08	P11	0x10	P25
0x01	P01	0x09	P12	0x11	P26
0x02	P02	0x0A	P13	0x12	P27
0x03	P03	0x0B	P20	0x13	P30
0x04	P04	0x0C	P21	0x14	P31
0x05	P05	0x0D	P22	0x15	P32
0x06	P06	0x0E	P23	Others	P10
0x07	P10	0x0F	P24		

说明:

- 1) PWM0M 的 Default 值为 0x07, 即 PWM0 默认从 P07 口输出
- 2) PWM1M 的 Default 值为 0x07, 即 PWM1 默认从 P10 口输出

15. 电容触摸(CDC)

触摸模块(TOUCH)采用电容数字转换器 (Cap Digital Convertor, CDC) 结构, 把触摸通道的电容值转换为计数值, 程序读取其数值后进行算法处理, 实现稳定可靠的触摸按键检测, 最多支持 25 个通道。

- 触摸控制寄存器 4(THCON4, C1h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			-		TLCMS[2:0]		
Access	-			-		R/W		
Default	-			-		000		

Bit[2:0] TLCMS[2:0]: 触摸共用 LED 时 COM 脚选择

000: 关闭

001: P25

010: P24

011: P23

100: P22

101: P21

110: P20

111: P32

16. 模数转换器(ADC)

集成了一个 12 位 ADC，其采用 SAR ADC 结构。

16.1 ADC SFR

ADC 模块相关寄存器：

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
C8h	ADCON0	ADCVO	ADCCUR	ADCE	ADCS	ADCEN	ADCVREF[2:0]			0100 0000
C9h	ADCON1	ADCCK[2:0]			-	ADCSPT[3:0]				000- 0000
CAh	ADCON2	-			ADCCALD	ADCCAL[3:0]				---0 0000
CBh	ADCON3	-			ADCADDR[4:0]					---0 0000
CCh	ADCCFG0	ADCCH7	ADCCH6	ADCCH5	ADCCH4	ADCCH3	ADCCH2	ADCCH1	ADCCH0	0000 0000
CDh	ADCCFG1	ADCCH15	ADCCH14	ADCCH13	ADCCH12	ADCCH11	ADCCH10	ADCCH9	ADCCH8	0000 0000
CEh	ADCCFG2	-	ADCCH22	ADCCH21	ADCCH20	ADCCH19	ADCCH18	ADCCH17	ADCCH16	-000 0000
CFh	ADCOL	ADCOL[7:0]								xxxx xxxx
D1h	ADCOH	-				ADCOH[3:0]				---- xxxx

• ADC 控制寄存器 0(ADCON0, C8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	ADCVO	ADCCUR	ADCE	ADCS	ADCEN	ADCVREF[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Default	0	1	0	0	0	000		

Bit[7] **ADCVO**: ADC 参考电压端口输入输出选择位

0: VREF 端口不输出参考电压

1: VREF 端口输出参考电压，可用于测试

(配置 VREF 端口输出参考电压前请把 CMODFS 位置 1)

Bit[6] **ADCCUR**: ADC 电流选择位(ADC Current Select)

1: ADC 模块电流 0.5mA

0: ADC 模块电流 1mA

Bit[5] **ADCE**: ADC 转换完成标志位

1: ADC 转换完成，由软件清 0；每次重新启动 ADCS 时硬件会清 ADCE

0: ADC 转换未完成

Bit[4] **ADCS**: ADC 转换启动位

0: 未进行转换或表示转换已完成

1: 正在转换或转换未完成，硬件不会清 0 此位，由软件清 0 终止转换

Bit[3] **ADCEN**: ADC 使能位

1: 使能 ADC

0: 禁止 ADC

Bit[2:0] **ADCVREF**: ADC 参考电压选择位

000: 选择 VDD 做为参考电压

001: 选择外部参考电压 VREF

- 010: 选择内部基准电压 1.28V 做为参考电压
 011: 选择内部基准电压 2.048V 做为参考电压
 100: 选择内部基准电压 3.072V 做为参考电压
 101: 选择内部基准电压 4.096V 做为参考电压
 Others: 无参考电压

• ADC 控制寄存器 1(ADCON1, C9h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	ADCCK[2:0]			-	ADCSPT[3:0]			
Access	R/W			-	R/W			
Default	000			-	0000			

Bit[7:5] **ADCCK**: ADC 时钟选择位 (CLKPER=RC32M/OSCD 分频)

- 000: ADC_CLK= CLKPER
 001: ADC_CLK= CLKPER /2
 010: ADC_CLK= CLKPER /4
 011: ADC_CLK= CLKPER /8
 100: ADC_CLK= CLKPER /16
 101: ADC_CLK= CLKPER /32
 110: ADC_CLK= CLKPER /64
 111: ADC_CLK= CLKPER /128

Bit[3:0] **ADCSPT**: ADC 采样时间选择

采样时间=(ADCSPT[3:0]+1)*T, T 为 ADC 时钟, 软件需配置此选项, 例如: ADCSPT[3:0]=8。

• ADC 控制寄存器 2(ADCON2, CAh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			ADCCALD	ADCCAL[3:0]			
Access	-			R/W	R/W			
Default	-			0	0000			

Bit[4] **ADCCALD**: ADC 失调校准的校准方向(ADC Offset Calibration Direction)

- 1: 负向校准
 0: 正向校准

Bit[3:0] **ADCCAL**: ADC 失调校准的校准值(ADC Offset Calibration)

失调电压值调整, 绝对值越大, 失调电压调整越大

• ADC 控制寄存器 3(ADCON3, CBh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-			ADCADDR[4:0]				
Access	-			R/W				
Default	-			00000				

Bit[4:0] **ADCADDR**: ADC 通道选择

ADCADDR[4:0]=x(0~22), 表示选择外部输入通道 ADCx, 如 ADCADDR[4:0]=2, 表示选择外

部输入通道 ADC2

ADCADDR[4:0]=23: 表示选择内部特殊通道 (内部 VDD/4)

ADCADDR[4:0]=24: 表示选择内部特殊通道 (内部 GND)

ADCADDR[4:0]=25: 表示选择内部特殊通道 (内部 VREF1.28V)

ADCADDR[4:0]=其他值: 表示选择内部特殊通道 (内部 GND)

• **ADC 通道配置寄存器 0(ADCCFG0, CCh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	ADCCH7	ADCCH6	ADCCH5	ADCCH4	ADCCH3	ADCCH2	ADCCH1	ADCCH0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7] **ADCCH7:** ADCCH7=1, AD7 对应的 IO 口做为 ADC 功能

Bit[6] **ADCCH6:** ADCCH6=1, AD6 对应的 IO 口做为 ADC 功能

Bit[5] **ADCCH5:** ADCCH5=1, AD5 对应的 IO 口做为 ADC 功能

Bit[4] **ADCCH4:** ADCCH4=1, AD4 对应的 IO 口做为 ADC 功能

Bit[3] **ADCCH3:** ADCCH3=1, AD3 对应的 IO 口做为 ADC 功能

Bit[2] **ADCCH2:** ADCCH2=1, AD2 对应的 IO 口做为 ADC 功能

Bit[1] **ADCCH1:** ADCCH1=1, AD1 对应的 IO 口做为 ADC 功能

Bit[0] **ADCCH0:** ADCCH0=1, AD0 对应的 IO 口做为 ADC 功能

• **ADC 通道配置寄存器 1(ADCCFG1, CDh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	ADCCH15	ADCCH14	ADCCH13	ADCCH12	ADCCH11	ADCCH10	ADCCH9	ADCCH8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Bit[7] **ADCCH15:** ADCCH15=1, AD15 对应的 IO 口做为 ADC 功能

Bit[6] **ADCCH14:** ADCCH14=1, AD14 对应的 IO 口做为 ADC 功能

Bit[5] **ADCCH13:** ADCCH13=1, AD13 对应的 IO 口做为 ADC 功能

Bit[4] **ADCCH12:** ADCCH12=1, AD12 对应的 IO 口做为 ADC 功能

Bit[3] **ADCCH11:** ADCCH11=1, AD11 对应的 IO 口做为 ADC 功能

Bit[2] **ADCCH10:** ADCCH10=1, AD10 对应的 IO 口做为 ADC 功能

Bit[1] **ADCCH9:** ADCCH9=1, AD9 对应的 IO 口做为 ADC 功能

Bit[0] **ADCCH8:** ADCCH8=1, AD8 对应的 IO 口做为 ADC 功能

• **ADC 通道配置寄存器 2(ADCCFG2, CEh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	ADCCH22	ADCCH21	ADCCH20	ADCCH19	ADCCH18	ADCCH17	ADCCH16
Access	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	-	0	0	0	0	0	0	0

Bit[6] **ADCCH22**: ADCCH22=1, AD22 对应的 IO 口做为 ADC 功能
 Bit[5] **ADCCH21**: ADCCH21=1, AD21 对应的 IO 口做为 ADC 功能
 Bit[4] **ADCCH20**: ADCCH20=1, AD20 对应的 IO 口做为 ADC 功能
 Bit[3] **ADCCH19**: ADCCH19=1, AD19 对应的 IO 口做为 ADC 功能
 Bit[2] **ADCCH18**: ADCCH18=1, AD18 对应的 IO 口做为 ADC 功能
 Bit[1] **ADCCH17**: ADCCH17=1, AD17 对应的 IO 口做为 ADC 功能
 Bit[0] **ADCCH16**: ADCCH16=1, AD16 对应的 IO 口做为 ADC 功能

• **ADC 转换结果低 8 位寄存器(ADCOL, CFh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	ADCOL[7:0]							
Access	R							
Default	0x00							

Bit[7:0] **ADCOL**: ADC 转换结果低 8 位
 其与 ADCOH 组成 12 位转换结果值: {ADCOH[3:0], ADCOL[7:0]}

• **ADC 转换结果高 4 位寄存器(ADCOH, D1h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-				ADCOH[3:0]			
Access	-				R			
Default	-				0000			

Bit[3:0] **ADCOH**: ADC 转换结果高 4 位
 其与 ADCOL 组成 12 位转换结果值: {ADCOH[3:0], ADCOL[7:0]}

16.2 ADC配置流程

1. ADC 失调校准：上电时进行校准，只需校准一次。

- 1) 配置 ADC 时钟 (ADCCK) 和采样时间周期 (ADCSPT)。
- 2) 配置 ADC 参考电压 ADCVREF 为内部基准。
- 3) 使能 ADC (ADCEN=1)，设置输入通道为内部 GND。
- 4) 软件等待 100us 左右，即等待 AD 初始化稳定。
- 5) 启动 ADC 连续转换。
- 6) 失调校准：
 - a) 初始化校准量为正向最大值：ADCCON2[4:0]=0FH;
 - b) 逐位从正向最大校准量调整到负，即 ADCCON2[4:0]校准顺序为 0FH→00H→10H→1FH，找到第一个 ADC 转换结果为 1 对应的校准量即为目标值，如果未找到 1 值，则找 2 值。若两者均未找到，则校准量 00 H 为目标值。

2. ADC 单次转换应用时的配置流程如下：

- 1) 配置 ADC 时钟 (ADCCK) 和采样时间周期 (ADCSPT)。
- 2) 配置 ADC 通道 ADCCHx 和 ADCADDR。
- 3) 配置 ADC 参考电压 ADCVREF。
- 4) 使能 ADC (ADCEN=1)，如需中断则 ADCIE =1。
- 5) 软件等待 100us 左右，即等待 AD 初始化稳定。
- 6) 配置 ADCON1 的 ADCS=1，启动 ADC 转换，直到软件清 0 才停止 ADC 转换。
- 7) 不开 ADC 中断时：判断 ADCE 是否置 1，置 1 则可以读取 ADC 转换数据。
- 8) 开 ADC 中断时：直接进中断读取 ADC 转换数据。
- 9) 软件清 ADCE 及 ADCS，完成一次 ADC 序列转换。

3. ADC 连续转换应用时的配置流程如下：

- 1) 配置 ADC 时钟 (ADCCK) 和采样时间周期 (ADCSPT)。
- 2) 配置 ADC 通道 ADCCHx 和 ADCADDR。
- 3) 配置 ADC 参考电压 ADCVREF。
- 4) 使能 ADC (ADCEN=1)，同时 ADC 中断也要打开 (ADCIE=1)。
- 5) 软件等待 100us，即等待 AD 初始化稳定。
- 6) 配置 ADCON1 的 ADCS=1，启动 ADC 转换（连续转换时 ADCS 只需启动一次，以后一直保持为 1，软件清 0 才停止 ADC 转换），中断发生后进中断程序读取 ADC 转换数据。

17. 低电压检测(LVD)

低电压检测（LVD）用于监控芯片自身的供电 VDD，可设置检测电压阈值。当 VDD 小于所设定的电压值时，可触发中断。LVD 相关寄存器如下：

● **LVD 控制寄存器(LVDCON0, E6h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	LVDEN	LVDOUT	TRDY	-				
Access	R/W	R	R	-				
Default	0	0	1	-				

Bit[7] **LVDEN**: LVD 使能位

0: 禁止 LVD

1: 使能 LVD

Bit[6] **LVDOUT**: LVD 比较结果位

0: VDD 高于设定阈值

1: VDD 低于设定阈值

Bit[5] **TRDY**: FLASH option 已读标志位

0: 未读

1: 已读

● **LVD 控制寄存器(LVDCON1, E7h):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-					LVDSEL[2:0]		
Access	-					R/W		
Default	-					000		

Bit[2:0] **LVDSEL**: LVD 检测电压选择位

000: 2.0V

001: 2.2V

010: 2.5V

011: 2.7V

100: 2.9V

101: 3.2V

110: 4.0V

111: 4.3V

使能 LVD 中断后，当 VDD 持续低于所设置电压时，只会产生一次中断，直到 VDD 高于所设置电压后，再次低于门限电压才会产生再次中断。

18. 同步串行总线(Mini I2C)

Mini I2C 模块支持芯片与外围 I2C 器件以标准 I2C 协议进行串行数据传输，可以配置为主机从机模式，最高支持 400Kbits/s。其特性如下：

- 1 I2C 主机功能：产生时钟；产生起始和停止信号；发送/接收数据；发送/接收 ACK
- 2 I2C 从机功能：可编程 I2C 从机地址；发送/接收数据；发送/接收 ACK；检测起始和停止信号
- 3 支持通用广播
- 4 支持标准模式和快速模式

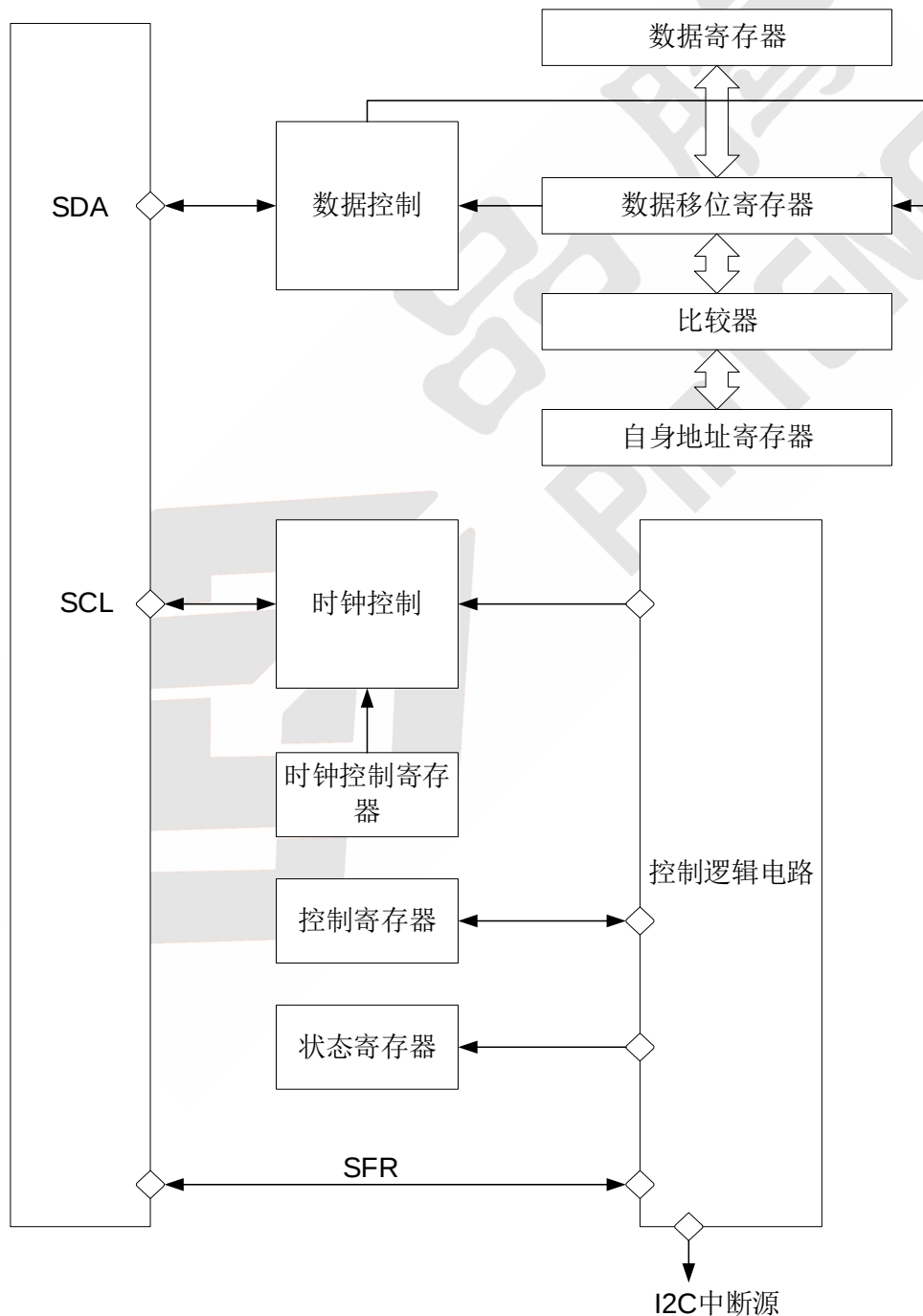


图 10 I2C 模块框图

18.1 I2C SFR

I2C 模块相关寄存器

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值	
E8h	I2CCON0	I2CEN	I2CSTA	I2CSTO	I2CMOD	I2CIE	-		GC	0000 0--0	
E9h	I2CSPD	I2CSPD[7:0]								1010 0000	
EAh	I2CSTATE	STASTA	STOSTA	GCSTA	-		I2CESTA[2:0]			000- -000	
EBh	I2CDAT	I2CDAT[7:0]									0000 0000
ECh	I2CCON1	I2CADR[6:0]							I2CWR		1010 0000

● I2C 控制寄存器(I2CCON0, E8h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CEN	I2CSTA	I2CSTO	I2CMOD	I2CIE	-		GC
Access	R/W	R/W	R/W	R/W	R/W	-		R/W
Default	0	0	0	0	0	-		0

Bit[7] **I2CEN**: I2C 使能位

0: 禁止 I2C

1: 使能 I2C

Bit[6] **I2CSTA**: 主机模式下, 发送起始位使能; 从机模式下无效

0: 无任何操作

1: 发送起始位

Bit[5] **I2CSTO**: 主机模式下, 发送停止位使能; 从机模式下无效

0: 无任何操作

1: 发送停止位

Bit[4] **I2CMOD**: 主机从机模式选择位

0: 从机模式

1: 主机模式

Bit[3] **I2CIE**: I2C 中断使能位

0: 中断无效

1: 中断有效

Bit[0] **GC**: 通用广播应答位

0: 对通用广播地址不应答

1: 对通用广播地址应答

● I2C 速率寄存器(I2CSPD, E9h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CSPD[7:0]							
Access	R/W							
Default	1010_0000							

Bit[7:0] **I2CSPD**: 该寄存器用于产生 SCL。计算公式为: $D_{i2cspd} = \frac{f_{clkper}}{f_{scl}}$

其中 f_{clkper} 为 I2C 模块的工作时钟频率, CLKPER=RC32M/OSCD 分频; f_{scl} 为 I2C 模块 SCL 的频率; D_{i2cspd} 为 I2CSPD 寄存器的数值, 其中 I2CSPD 取值范围为(0x28~0xFF)。

● I2C 状态寄存器(I2CSTATE, EAh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	STASTA	STOSTA	GCSTA	-	-	I2CESTA		
Access	R	R	R	-	-	R		
Default	0	0	0	-	-	000		

Bit[7] **STASTA**:

0: 主机模式时, 表示 START 位发送未完成; 从机模式时, 表示没有接收到 START 位

1: 主机模式时, 表示 START 位发送已完成; 从机模式时, 表示已经接收到 START 位。进入数据传输阶段, 自动清除

Bit[6] **STOSTA**:

0: 主机模式时, 表示 STOP 位发送未完成; 从机模式时, 表示没有接收到 STOP 位

1: 主机模式时, 表示 STOP 位发送已完成, 配置发送 START 信号会自动清除; 从机模式时, 表示已经接收到 STOP 位, 接收到 START 信号会自动清除

Bit[5] **GCSTA**:

0: 表示接收到的地址为设备地址

1: 表示接收到的地址为通用广播地址

注: 只有在支持通用广播模式时有效, 不支持通用广播地址操作时, 读取该位的值恒为 0

Bit[2:0] **I2CESTA**: I2C 错误状态类型

000: 无错误

001: 主机模式下地址不匹配从机回 NACK

010: 主机模式下发数据时从机回 NACK

011: 从机发送模式下主机回 NACK

● I2C 数据寄存器(I2CDAT, EBh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CDAT[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **I2CDAT**: 该寄存器寻址两个寄存器: 一个移位寄存器和一个接收锁存寄存器 I2CDAT 的写入将发送字节到移位寄存器中

I2CDAT 的读取将返回接收锁存器中的内容

注：因 I2CDAT 寻址两个寄存器，读出的值与写入的值不一定相同，故软件不应对此寄存器用“读-修改-写”类型的指令。

● I2C 地址寄存器(I2CCON1, ECh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	I2CADR[6:0]							I2CWR
Access	R/W							R
Default	1010000							0

Bit[7:1] **I2CADR**: 该寄存器从机模式有效，设定设备 I2C 从机地址

Bit[0] **I2CWR**: 从机模式下，I2C 模块的读写控制字

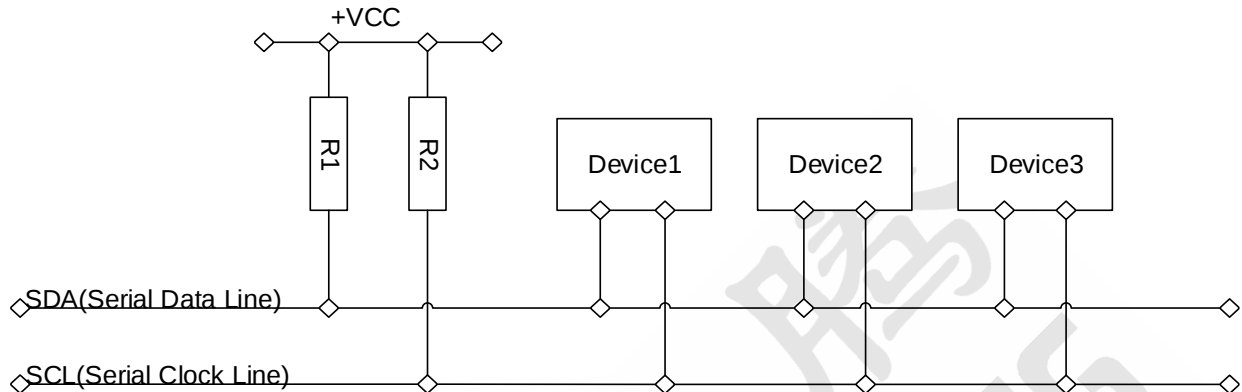
0: 从机模式下，I2C 模块收数据

1: 从机模式下，I2C 模块发数据

18.2 I2C功能描述

1. 物理层

I2C 总线由两个信号组成，称为串行时钟线(SCL)和串行数据线(SDA)。要传输的数据通过 SDA 线发送，并与 SCL 的时钟信号同步。I2C 网络上的所有设备都连接到相同的 SCL 和 SDA 线。



两条 I2C 总线都是开漏结构，所以 I2C 总线上必须外接上拉电阻或者打开芯片内部上拉，每个连接到总线上的设备都有一个唯一的 7 位地址。

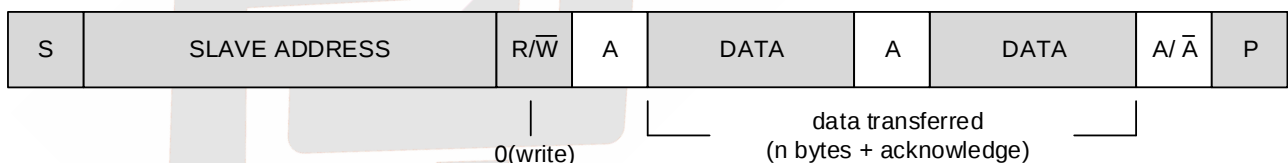
2. 模式选择

I2C 可以在以下 4 种模式中的一种运行：从机发送模式、从机接收模式、主机发送模式、主机接收模式。默认情况下，I2C 处于从机模式。

3. 数据传输协议

主机和从机之间遵循以下协议以在它们之间传输数据。

数据通过单个 SDA 数据线在主机和从机之间传输，通过 0 和 1 的模式序列，每个 0 和 1 序列称为 1 个事务，每个事务中的数据结构如下：



From master to slave

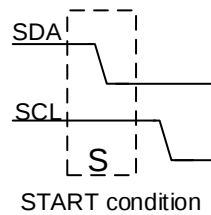


From slave to master

A = acknowledge(SDA LOW)
 $\bar{A}$ = acknowledge(SDA HIGH)
 S = START condition
 P = STOP condition

• 起始位

每当主机决定开始一个事务时，它会在 SCL 保持为高电平时产生 SDA 的下降沿。一旦主机发送启动条件，所有从设备即便处于睡眠状态也会被唤醒，并等待地址为。



- **地址位**

由 7 位组成，并填充主机需要发送/接收数据的从机地址，I2C 总线上所有从机将接收到的地址与其唯一地址进行比较。

- **读/写位**

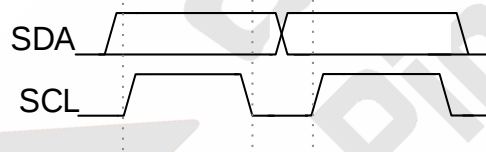
指定数据传输方向。如果主机需要向从机发送数据，则该位置 0。如果主机需要向从机获取数据，则该位置 1。

- **ACK/NACK 位**

指示已确定/未确定位，如果从机接收到的地址和其唯一地址相匹配，则该位的值由从机置 0，否则它保持 1。

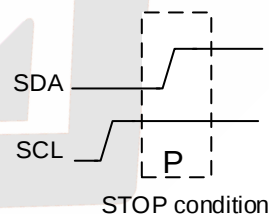
- **数据块**

由 8 位组成，它由发送方设置需要传输到接收方的数据位。数据传输时，在 SCL 高电平时必须保持 SDA 数据稳定，SDA 数据变化只能发生在 SCL 低电平区间。



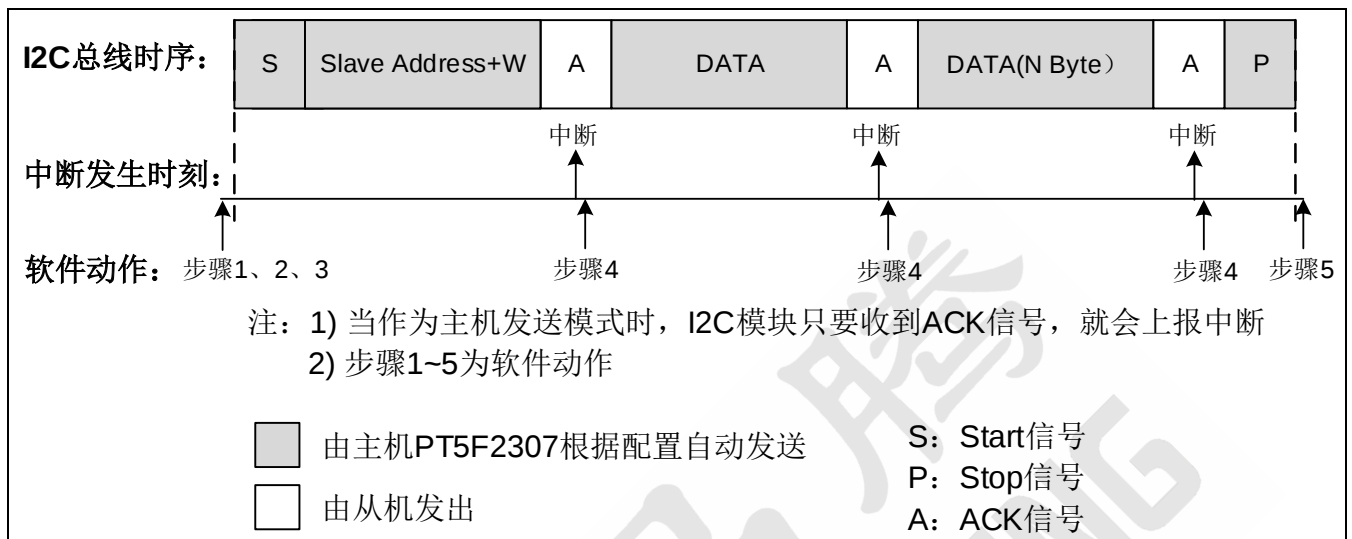
- **停止位**

在通过 SDA 传输完所需要的数据块后，主机在 SCL 高电平时产生 SDA 上升沿。

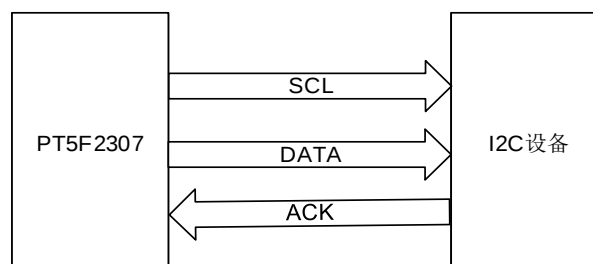


18.3 I2C通讯过程

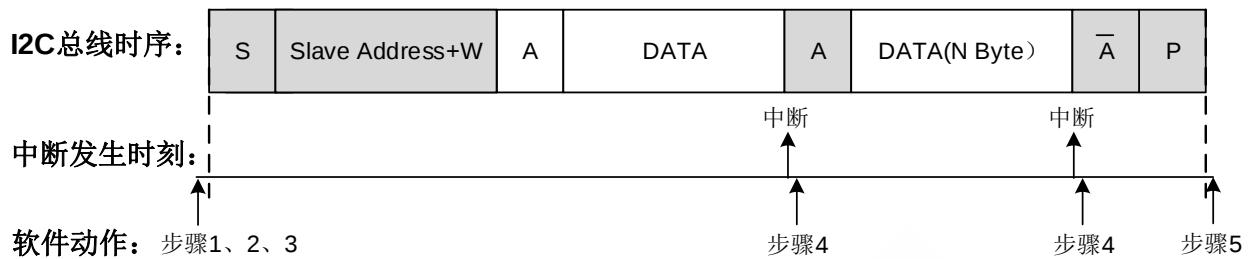
18.3.1 主机发送模式软件配置流程



- I2C 初始化
 - 配置 I2C 通信速度 I2CSPD。
 - 配置 I2CCON 的 I2CMOD、I2CIE、I2CEN 位。
- 配置从机地址和写信号
 - 配置 I2CDAT, 发送目标从机地址和写信号 SLA+W。
- 发送起始信号
 - 配置 I2CCON 的 I2CSTA 位和 I2CSTO 发送 Start 信号 (I2CSTA=1, I2CSTO=0)。
- 开始写数据
 - 若开启中断使能, 则进中断, 写入 I2CDAT。直到最后一个数据, 进中断后, 则配置 I2CCON 的 I2CSTA 和 I2CSTO 位发送 STOP 信号 (I2CSTA=0, I2CSTO=1)。
 - 若未开启中断使能, 查询到中断标志位后, 清除标志位, 写入 I2CDAT。直到最后一个数据, 查询到中断标志位后, 清除中断标志位, 则配置 I2CCON 的 I2CSTA 和 I2CSTO 位发送 STOP 信号 (I2CSTA=0, I2CSTO=1)。
- 待 I2CSTATE 的 STOSTA 为“1”时, 代表 STOP 信号发送完成, 返回步骤 2 开始新一轮的传输。
- 超时检测
 - 若开启中断使能, 当进中断后, 计数器清零。长时间未进中断则判为超时, 关闭 I2CEN, 返回步骤 1.a。
 - 若未开启中断使能, 当查询到中断标志位时, 计数器清零。长时间未查询到中断标志位, 则判为超时, 关闭 I2CEN, 返回步骤 1.a。



18.3.2 主机接收模式软件配置流程

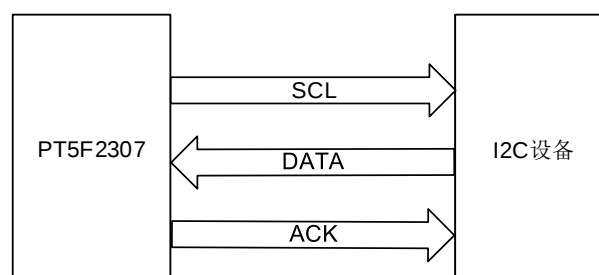


注：1) 当作为主机发送模式时，I2C模块只要收到ACK信号，就会上报中断
2) 步骤1~5为软件动作

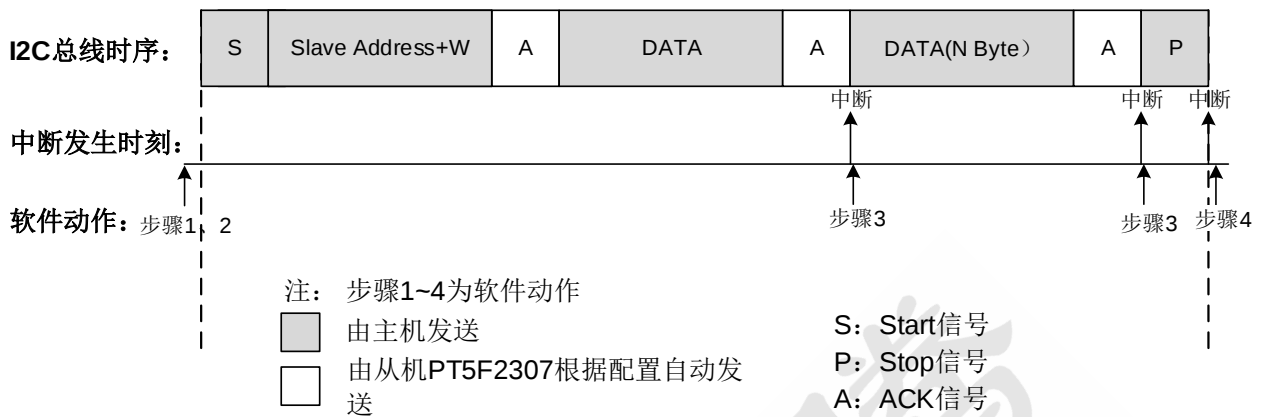
■ 由主机PT5F2307根据配置自动发送
□ 由从机发出

S: Start信号
P: Stop信号
A: ACK信号
 $\bar{A}$: NACK信号

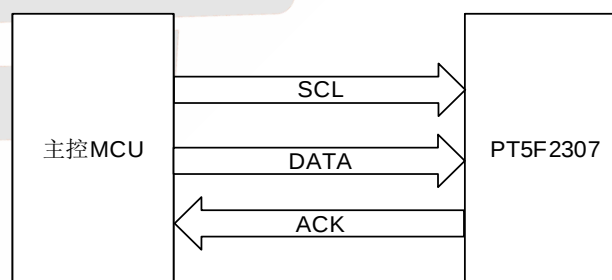
- I2C 初始化
 - 配置 I2C 通信速度 I2CSPD。
 - 配置 I2CCON 的 I2CEN、I2CMOD、I2CIE 位。
- 配置从机地址和读信号
 - 配置 I2CDAT，发送目标从机地址和读信号 SLA+R。
- 发送起始信号
 - 配置 I2CCON 的 I2CSTA 位和 I2CSTO 发送 Start 信号 (I2CSTA=1, I2CSTO=0)。
- 开始读数据
 - 若开启中断使能，则进中断，读取 I2CDAT。直到最后一个数据，进中断后，则配置 I2CCON 的 I2CSTA 和 I2CSTO 位发送 STOP 信号 (I2CSTA=0, I2CSTO=1)。
 - 若未开启中断使能，查询到中断标志位后，清除标志位，读取 I2CDAT，直到最后一个数据，查询到中断标志位后，则先配置 I2CCON 的 I2CSTA 和 I2CSTO 位发送 STOP 信号 (I2CSTA=0, I2CSTO=1) 再读取 I2CDAT。
- 待 I2CSTATE 的 STOSTA 为“1”时，代表 STOP 信号发送完成，返回步骤 2 开始新一轮的传输。
- 超时检测
 - 若开启中断使能，当进中断后，计数器清零。长时间未进中断则判为超时，关闭 I2CEN，返回步骤 1.a。
 - 若未开启中断使能，当查询到中断标志位时，计数器清零。长时间未查询到中断标志位则判为超时，关闭 I2CEN，返回步骤 1.a。



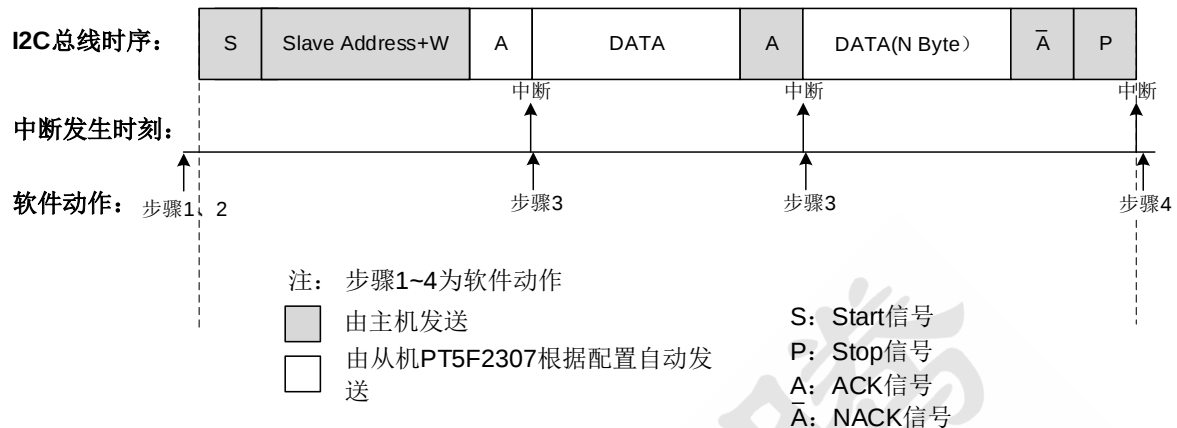
18.3.3 从机接收模式软件配置流程



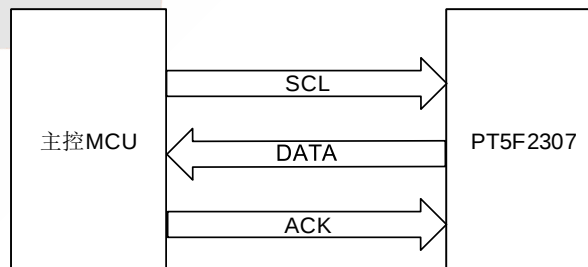
- I2C 初始化
 - 配置 I2CCON 的 I2CMOD、GC、I2CIE 位。
- 配置从机地址
 - 配置 I2CADR 的从机地址。
 - 配置 I2CCON 的 I2CEN 位使能 I2C 模块。
- 开始接收数据
 - 若开启中断使能，则进中断，读取 I2CDAT(第一次进中断需判断 I2CWWR)。
 - 若未开启中断使能，查询到中断标志位后，清除标志位，读取 I2CDAT(第一次查询到中断标志位需判断 I2CWWR)。
- 待 I2CSTATE 的 STOSTA 为“1”时，代表已经接收到主机发出的 STOP 信号，模块上报中断，数据传输结束。
- 超时检测
 - 若开启中断使能，当进中断后，计数器清零。长时间未进中断则判为超时，关闭 I2CEN，返回步骤 1.a。
 - 若未开启中断使能，当查询到中断标志时，计数器清零。长时间未查询到中断标志位则判为超时，关闭 I2CEN，返回步骤 1.a。



18.3.4 从机发送模式软件配置流程



- I2C 初始化
 - 配置 I2CCON 的 I2CMOD、GC、I2CIE 位。
- 配置从机地址
 - 配置 I2CADR 的从机地址。
 - 配置 I2CCON 的 I2CEN 位使能 I2C 模块
- 开始发送数据
 - 若开启中断使能，则进中断，写入 I2CDAT(第一次进中断需判断 I2CWR)。
 - 若未开启中断使能，查询到中断标志位后，清除标志位，写入 I2CDAT(第一次查询到中断标志位需判断 I2CWR)。
- 待 I2CSTATE 的 STOSTA 为“1”时，代表已经接收到主机发出的 STOP 信号，模块上报中断，数据传输结束。
- 超时检测
 - 若开启中断使能，当进中断后，计数器清零。长时间未进中断则判为超时，关闭 I2CEN，返回步骤 1.a。
 - 若未开启中断使能，当查询到中断标志时，计数器清零。长时间未查询到中断标志位则判为超时，关闭 I2CEN，返回步骤 1.a。



4. I2CESTA错误状态解释:

- 主发模式和主收模式
 - 主机在发送完地址后，从机回复 NACK。I2C 上报中断，I2CESTA 为 0x01;
 - 主机在发送完数据后，从机回复 NACK。I2C 上报中断，I2CESTA 为 0x02;

2. 从发模式:

- a) 从机在发送完数据后, 主机回复 NACK。但主机继续传输数据, I2C 上报中断, I2CESTA 为 0x03;

18.4 I2C时钟设置

作为从机时, SCL 的时钟由主机输入, 和从机的时钟配置无关。作为主机时, SCL 的输出时钟由 I2CSPD 决定。

19. 通用异步收发器(UART)

集成了两个硬件功能一样的 UART，最大波特率 115200。

UART_TX 口可根据配置不同模式输出，该端口需配置为输出，不使能 OD 为 CMOS 输出，使能 OD 为开漏输出，同时使能该端口上下拉可输出中间电压。选择强或弱上下拉，输出配置电压可为 3/5VDD、1/2VDD。

UART_RX 口建议配置为输入且使能上拉。

19.1 UART SFR

UART 寄存器列表

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
E5h	UAT0CON	-	UAT0EN	-	UAT0SM	-	UAT0RB8	-	UAT0TB8	-0-0 -0-0
F9h	UAT1CON	-	UAT1EN	-	UAT1SM	-	UAT1RB8	-	UAT1TB8	-0-0 -0-0
FAh	UAT0SBUF	UAT0SBUF[7:0]								0000 0000
FBh	UAT1SBUF	UAT1SBUF[7:0]								0000 0000
FCh	UAT0SREL	UAT0SREL[7:0]								1101 1001
FDh	UAT0SREH	-				UAT0SREH[3:0]				---- 0011
FEh	UAT1SREL	UAT1SREL[7:0]								1101 1001
FFh	UAT1SREH	-				UAT1SREH[3:0]				---- 0011

● UART0 控制寄存器(UAT0CON, E5h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	UAT0EN	-	UAT0SM	-	UAT0RB8	-	UAT0TB8
Access	-	R/W	-	R/W	-	R/W	-	R/W
Default	-	0	-	0	-	0	-	0

Bit[6] **UAT0EN**: UART0 使能位

0: 禁止 UART0

1: 使能 UART0

Bit[4] **UAT0SM**: UART0 串行工作模式选择

0: 8 位异步方式，可变波特率

1: 9 位异步方式，可变波特率

Bit[2] **UAT0RB8**: UART0 在模式 2 下接收的第 9 位

Bit[0] **UAT0TB8**: UART0 在模式 2 下发送的第 9 位，由软件置 1 或清 0

● UART1 控制寄存器(UAT1CON, F9h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	UAT1EN	-	UAT1SM	-	UAT1RB8	-	UAT1TB8
Access	-	R/W	-	R/W	-	R/W	-	R/W
Default	-	0	-	0	-	0	-	0

Bit[6] **UAT1EN**: UART1 使能位

0: 禁止 UART1

1: 使能 UART1

Bit[4] **UAT1SM**: UART1 串行工作模式选择

0: 8 位异步方式, 可变波特率

1: 9 位异步方式, 可变波特率

Bit[2] **UAT1RB8**: UART1 在模式 2 下接收的第 9 位

Bit[0] **UAT1TB8**: UART1 在模式 2 下发送的第 9 位, 由软件置 1 或清 0

● **UART0 BUF 寄存器(UAT0SBUF, FAh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	UAT0SBUF[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **UAT0SBUF**: 该寄存器寻址两个寄存器: 一个移位寄存器和一个接收锁存寄存器

UAT0SBUF 的写入将发送字节到移位寄存器中

UAT0SBUF 的读取将返回接收锁存器中的内容

注: 因 UAT0SBUF 寻址两个寄存器, 读出的值与写入的值不一定相同, 故软件不应对此寄存器用“读-修改-写”类型的指令。

● **UART1 BUF 寄存器(UAT1SBUF, FBh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	UAT1SBUF[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **UAT1SBUF**: 该寄存器寻址两个寄存器: 一个移位寄存器和一个接收锁存寄存器

UAT1SBUF 的写入将发送字节到移位寄存器中

UAT1SBUF 的读取将返回接收锁存器中的内容

注: 因 UAT1SBUF 寻址两个寄存器, 读出的值与写入的值不一定相同, 故软件不应对此寄存器用“读-修改-写”类型的指令。

● **UART0 波特率寄存器低 8 位(UAT0SREL, FCh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	UAT0SREL[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **UAT0SREL**: UART0 波特率发生器计数器溢出值低 8 位寄存器

● **UART0 波特率寄存器高 4 位(UAT0SREH, FDh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	UAT0SREH[3:0]			
Access	-	-	-	-	R/W			
Default	-	-	-	-	0011			

Bit[3:0] **UAT0SREH**: UART0 波特率发生器计数器溢出值高 4 位寄存器

● **UART1 波特率寄存器低 8 位(UAT1SREL, FEh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	UAT1SREL[7:0]							
Access	R/W							
Default	0x00							

Bit[7:0] **UAT1SREL**: UART1 波特率发生器计数器溢出值低 8 位寄存器

● **UART1 波特率寄存器高 4 位(UAT1SREH, FFh):**

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-	-	-	-	UAT1SREH[3:0]			
Access	-	-	-	-	R/W			
Default	-	-	-	-	0011			

Bit[3:0] **UAT1SREH**: UART1 波特率发生器计数器溢出值高 4 位寄存器

19.2 UART工作模式

UART 有 2 种工作方式。在通讯之前用户必须先初始化 UATxCON，使能 UART 和波特率发生器，并选择工作模式。输入的起始位初始化接收，外部发送器通讯以发送起始位开始。

表 12 UART 工作方式列表

UATxSM	模式	类型	波特率	帧长度	起始位	停止位	第 9 位
0	模式	异步	自带波特率发生器的溢出率	10 位	1	1	无
1	模式	异步	自带波特率发生器的溢出率	11 位	1	1	0, 1

模式 1：8 位 UART，可变波特率，异步全双工

模式 1 提供 10 位全双工异步通讯，10 位由一个起始位（逻辑 0），8 个数据位（低位为第一位）和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 UATxSBUF 中而停止位储存在寄存器 UATxCON[2]的 UATxRB8 中。模式 1 的波特率可灵活配置，12bit 波特率发生器的计数时钟为 CPUCLK。

发送数据时，数据位时间与波特率移位时钟是同步的，与对 UATxSBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后会再输出 1 位 1，随后停止位在 TXD 引脚上移出，在停止位发出后同时 UATxTIF 标志置 1。

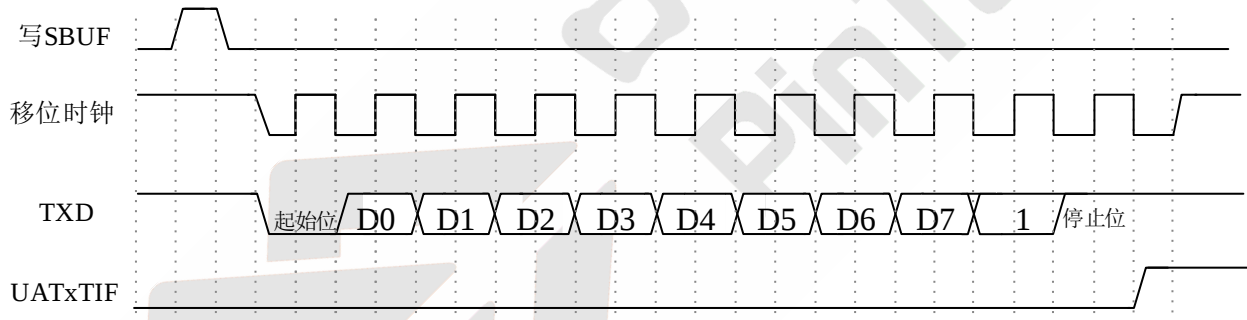


图 11 模式 1 的发送数据时序

接收数据时，当 RXD 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RXD 数据位的正中进行采样接收。如果所接收的第 1 位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RXD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 UATxSBUF 和 UATxRB8 中，RI 置 1。但必须满足 UATxRIF 等于 0 或者接收的停止位等于 1。

如果这些条件被满足，那么停止位装入 UATxRB8，8 个数据位装入 UATxSBUF，UATxRIF 被置 1。否则接收的数据帧会丢失。这时，接收器将重新去探测 RXD 端的另一个下降沿。用户必须用软件清 UATxRIF，然后才能再次接收。

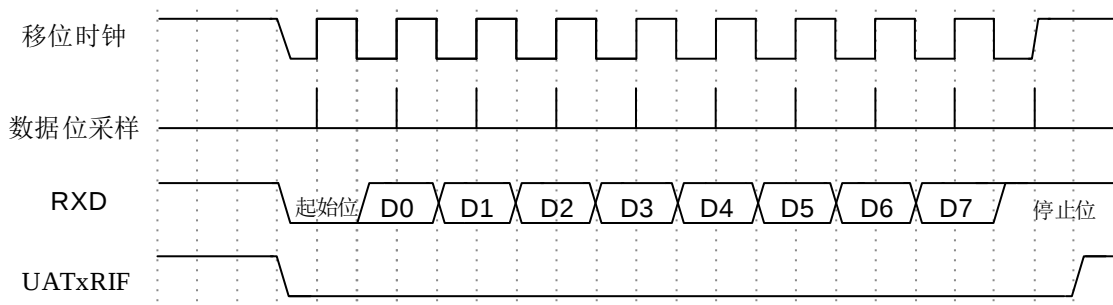


图 12 模式 1 的接收数据时序

模式 2：9 位 UART，固定波特率，异步全双工

这个模式使用异步全双工通讯中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位为第 1 位），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。

在数据传送时，第 9 数据位（寄存器 UATxCON[3]中的 UATxTB8）可以写 0 或 1，例如，可写入 PSW 中的奇偶位 P。当接收到数据时，第 9 数据位移入 UATxRB8 而停止位不保存。

发送数据时，同时也将 UATxTB8 载入到发送移位寄存器的第 9 位中。数据位时间与波特率移位时钟是同步的，与对 UATxSBUF 的写操作不同步。起始位首先在 TXD 引脚上移出，然后是 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TXD 引脚上移出，在停止位开始发送时 UATxTIF 标志置 1。

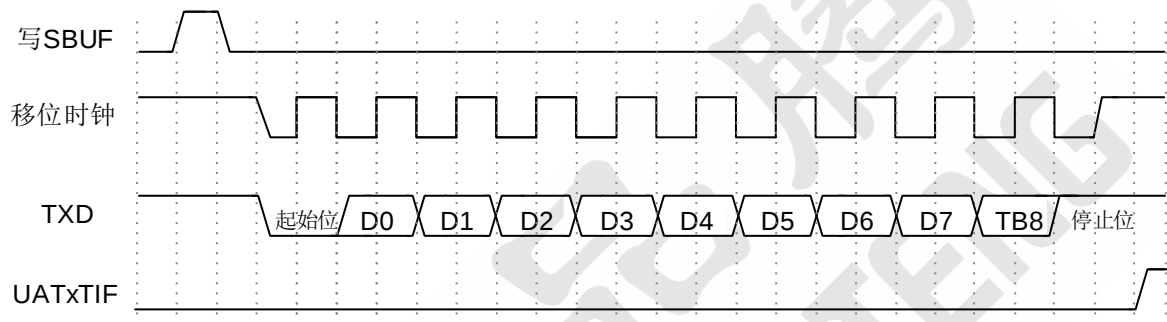


图 13 模式 2 的发送数据时序

接收数据时。当 RXD 引脚检测到下降沿时串行口开始接收串行数据。CPU 在 RXD 数据位的正中进行采样接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RXD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 UATxSBUF 和 UATxRB8 中，UATxRIF 置 1。但必须满足 UATxRIF=0 或者接收的第 9 位=1，接收的字节必须符合约定从机地址。

如果这些条件被满足，那么第 9 位移入 UATxRB8，8 位数据移入 UATxSBUF，UATxRIF 被置 1。否则接收的数据帧会丢失。在停止位的当中，接收器将重新去探测 RXD 端的另一个下降沿。用户必须用软件清 UATxRIF，然后才能再次接收。

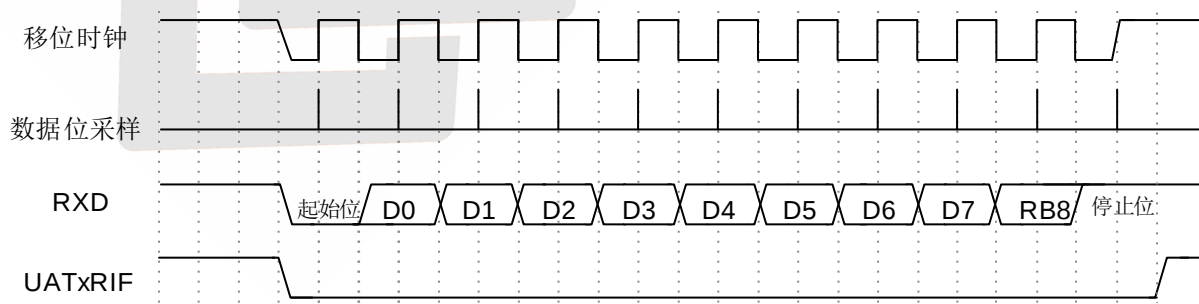


图 14 模式 2 的接收数据时序

19.3 UART波特率

UART 自带一个波特率发生器。它实质上是一个 12 位递增计数器，其计数时钟为 CPUCLK。用户配置寄存器 UATxSREH 和 UATxSREL 来配置计数器的溢出值，从而实现需要的波特率。计算公式为：

$$SRE = \frac{f_{clkper}}{f_{bps}} - 1$$

表 13 UART 波特率列表

波特率 BPS (Hz)	实际波特率 (Hz)	理想偏差 (%)	典型偏差 (%) 考虑时钟偏差±1%	计数时钟 (MHz)	计数器溢出值
4800	4800.480	0.010	1.01	16/8	D02h/681h
9600	9603.842	0.040	1.04	16/8	681h/340h
19200	19207.683	0.040	1.04	16/8	340h/19Fh
38400	38461.538	0.160	1.16	16/8	19Fh/0CFh
57600	57761.733	0.281	1.28	16/8	114h/08Ah
115200	115942.029	0.644	1.64	16/8	089h/044h

注意：UART 的端口可以单独使用，另一个端口可做普通 GPIO；UART 使用的端口需要软件事先配置好方向，TX 口还需配置初始输出高电平。

20. 在线编程(IAP)

集成了 1 路在线编程模块。

20.1 IAP SFR

地址	符号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	复位值
AAh	IAPTRI	IAPTRI[7:0]								0000 0000
A9h	IAPCON1	INFEN	IAPLOCK	-			IAPCMD[2:0]			00-- -000
ABh	IAPADRL	-		IAPADR[5:0]						--00 0000
ACH	IAPADRH	IAPADR[13:6]								0000 0000
ADh	IAPDAT	IAPDAT[7:0]								0000 0000
A Eh	IAPCON2	PRTIME[3:0]				ERTIME[3:0]				0000 0000

• IAP 触发寄存器(IAPTRI, AAh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	IAPTRI[7:0]							
Access	W							
Default	0x00							

Bit[7:0] **IAPTRI**: IAP 写操作触发使能,

IAPTRI 连续写入 0x48, 0x65, 0x88, IAP 擦除及写操作有效。

IAPTRI 写 0 退出擦除及写操作状态。

• IAP 控制寄存器(IAPCON1, A9h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	INFEN	IAPLOCK	-			IAPCMD[2:0]		
Access	R/W	R/W	-			W		
Default	0	0	-			000		

Bit[7] **INFEN**: 此位为 1 时选择 Information 区, 为 0 选择 CODE 区。

选择 Information 区时只能对地址 0x00~0x7F 区域页擦除、页烧录。可读全部 Information 区。

Bit[6] **IAPLOCK**: FLASH 编程状态, 为 1 时 IAP 擦除及写命令无效。

此位为 1 时, 擦除及写命令处于锁定状态, 需要软件写 0 清除或重新上电;

不按文档规定的 IAPTRI 触发顺序值写入时, 此位才置 1;

当该位为 0, 且按文档规定的 IAPTRI 触发顺序值写入后, 才能进行擦除及写操作。

Bit[2:0] **IAPCMD**: IAP 操作选择

000: 无操作。

001: 块擦除 (每块 4096 字节)。

010: 页擦除 (每页 64 字节)。

011: 读数据

100: 清除页缓存

101: 页烧录

其它：无操作

注意：

1. 配置为有效命令，执行完后 IAPCMD 自动清零。每配置一次读数据命令或写一次 IAPDAT，IAPADR_L 自动加 1，但 IAPADR_H 不会改变。写 IAPDAT 前一定要先执行清除页缓存。
2. 可以单次或多次写 IAPDAT，最多写 64 次。把 IAPCMD 配置为页编程，此时会把写入 IAPDAT 的值编程进 IAPADR_H 配置的页。一页内没有写入数据的地址会固定写入 FFH。

• IAP 操作地址低位寄存器(IAPADR_L, ABh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	-		IAPADR[5:0]					
Access	-		R/W					
Default	-		0x00					

Bit[5:0] **IAPADR**: IAP 操作地址低 6 位。每配置一次读数据命令或写一次 IAPDAT，该 6 位地址自动加 1，加到溢出时会回复到 0，但不往高 8 位进位。

注意：此寄存器与 IAPADR_H（8 Bit）配合构成完整的地址。

• IAP 操作地址高位寄存器(IAPADR_H, ACh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	IAPADR[13:6]							
Access	R/W							
Default	0x00							

Bit[7:0] **IAPADR**: IAP 操作地址高 8 位

• IAP 操作数据寄存器(IAPDAT, ADh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	IAPDAT[7:0]							
Access	R/W							
Default	0xXX							

Bit[7:0] **IAPDAT**: IAP 读写数据 8 位寄存器。

当用户进入擦除及写操作状态后，数据写入此寄存器，读此寄存器为前次写入值。

只有当用户退出擦除及写操作状态，并进行了读出操作后，读此寄存器为读地址值。

• IAP 控制寄存器(IAPCON2, AEh):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	PRTIME[3:0]				ERTIME[3:0]			
Access	R/W				R/W			
Default	0x0				0x0			

Bit[7:4] **PRTIME[3:0]**: FLASH 编程脉冲宽度调整。

调整单位为 0.128ms，调整方法为 默认编程脉宽(2ms)+0.128*PRTIME。

PRTIME 为有符号数，0001 为正 1，1111 为负 1。

Bit[3:0] **ERTIME[3:0]**: FLASH 擦除脉冲宽度调整。

调整单位为 0.128ms，调整方法为 默认擦除脉宽(3ms)+0.128*ERTIME。

ERTIME 为有符号数，0001 为正 1，1111 为负 1。

注意：IAPCON2 的值通过使用 IAP 读操作读取 Information 区地址 0x87 得到。

品 騰
PinTENG

20.2 IAP操作流程

1. CODE区写入数据操作步骤:

- 1) 读 Information 区 0x0087 地址数据, 并写入 IAPCON2;
- 2) 对 IAPTRI 顺序写入 0x48, 0x65, 0x88, 使能 IAP 的擦除和写模式。注意: 如果往 IAPTRI 写入其他任何值, 则进入 IAPLOCK 状态。进行重新上电或通过软件写 IAPLOCK 位为 0 来解除锁定, 软件置 0 后, 需要重新开始对 IAPTRI 写入
- 3) 设置 IAPADR, 即设置数据要写入的某个地址。
- 4) 设置 INFEN 为 0, IAPCMD 为 001 或 010, 会擦除 IAPADR 地址所在的块或页。
- 5) 设置 IAPCMD 为 100, 清除页缓存。
- 6) 把数据写入 IAPDAT, 可单次或多次连续写入, 每次写入 IAPADRL 会自动加 1。
- 7) 设置 IAPCMD 为 101;
- 8) 如果要继续写其它页, 则回到步骤(3)开始。如果要退出写入操作, 则到步骤(9);
- 9) 对 IAPTRI 写入 0, 退出写操作。

注意: IAP 只能页操作, 修改一个地址数据时需要先读出整页地址的内容保存, 再进行页擦除, 将读取的内容和需要修改的数据重新写入页。

例 1 写 CODE 区数据

```
MOV    IAPADRH,    #02H    //读地址 0087H
MOV    IAPADRL,    #07H
MOV    IAPCON1,    #83H    //读 Information 操作
MOV    A,          IAPDAT
MOV    IAPCON2,    A        //Information 区 0087H 地址数据赋给 IAPCON2
MOV    IAPTRI,     #48H    //进入写 MODE
MOV    IAPTRI,     #65H
MOV    IAPTRI,     #88H
MOV    IAPADRH,    #10H    //写入地址 1000H
MOV    IAPADRL,    #00H
MOV    IAPCON1,    #02H    //擦除 1000H~103FH
MOV    IAPCON1,    #04H    //配置为写入模式
MOV    IAPDAT,     #A5H    //写入数据 A5H 到 1000H
MOV    IAPDAT,     #55H    //写入数据 55H 到 1001H
MOV    IAPCON1,    #05H    //数据页烧录入 FLASH。1002H~103F 烧录为 0xFF
MOV    IAPTRI,     #00H    //退出写 MODE
```

2. CODE区读出数据操作步骤:

- 1) 设置 IAPADR, 即设置数据要读出的某个地址;
- 2) 设置 INFEN 为 0, IAPCMD 为 011;
- 3) 把 IAPDAT 中数据读出到 A 中;
- 4) 如果要连续读, 则回到步骤(2)开始。

例 2 读 CODE 数据

```
MOV    IAPADRH,    #10H    //读地址 1000H
MOV    IAPADRL,    #00H
MOV    IAPCON1,    #03H    //读操作
MOV    A,          IAPDAT    //00H 地址数据赋给 A
```

21. 看门狗定时器(WDT)

看门狗定时器（WDT）无需任何外部电路即可工作，睡眠模式同样工作，启用看门狗使超时发生后 MCU 重启复位。

在一般操作或睡眠模式情况下，看门狗定时器溢出会导致 MCU 复位同时 WDTF 位被置 1。

看门狗溢出时间基准为 2ms，溢出时钟可以通过设置 WDTCON .PS[2:0]分频周期最大达到 256ms。再配置 WDTSET[7:0]，溢出周期最大可为 64 秒。

STOP 下的 WDT 溢出复位会导致芯片复位，在进睡眠之前未完成的操作不会继续。

对 WDTCON 中 WDTIC 位写 1 能使分频器和计数器清零。

对 PCON 中 STOP 位或 IDLE 位写 1 能使分频器和计数器清零。

• WDT 控制寄存器(WDTCON, D3h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	LRCEN	WDTSEL	WDTEN	WDTF	WDTC	PS[2:0]		
Access	R/W	R/W	R/W	R	W	R/W		
Default	1	0	1	0	0	111		

Bit[7] **LRCEN**: 内部 RC131K 时钟使能位

1: 使能 RC131K 时钟

0: 禁止 RC131K 时钟

Bit[6] **WDTSEL**: WDT 功能选择位

1: WDT 溢出唤醒，可以唤醒 IDLE 或 STOP 模式

0: WDT 溢出复位

Bit[5] **WDTEN**: WDT 使能位

1: 使能 WDT 溢出，溢出后系统复位或唤醒 IDLE/STOP 模式

0: 禁止 WDT 溢出

Bit[4] **WDTF**: WDT 溢出标志

1: 看门狗定时器溢出

0: 看门狗定时器无溢出。

POR、LVR、RST 复位会清 0 该标志位；WDTC、STOP、IDLE 位写 1 也会清该标志位。

Bit[3] **WDTC**: 清 WDT 定时器

1: 清 WDT 定时器

0: 不清 WDT 定时器

Bit[2:0] **PS[2:0]**: WDT 基准时钟选择控制位

000: WDT Rate=2ms

001: WDT Rate=4ms

010: WDT Rate=8ms

011: WDT Rate=16ms

100: WDT Rate=32ms

101: WDT Rate=64ms

110: WDT Rate=128ms

111: WDT Rate=256ms

• WDT 计数寄存器(WDTSET, D2h):

BIT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Name	WDTSET[7:0]							
Access	R/W							
Default	0xFF							

WDT 溢出周期为 $\text{WDT Rate} * (\text{WDTSET} + 1)$

22. 复位系统(RESET)

有 5 种复位方式：

1. 上电复位(POR)：上电时电压低于上电复位电压芯片保持复位状态，直到高于上电复位电压。
2. 外部复位(RST)：当外部 RESET 脚上出现大于 2us 的低电平脉冲时，复位系统。外部信号 (RST) 通过一个噪声滤波器滤除小于 2us 脉冲宽度的信号；
3. 低压复位(LVR)：当检测到电压低于 LVR 电压值时对芯片复位。
4. 看门狗复位(WDT)：看门狗超时后 MCU 重启复位。
5. 软件复位(WRST)：对 PCON 寄存器中 WRST 写 1 产生软件复位。

当内部的 POR、LVR、外部复位脉冲加载完成后，PWRT 开始计数。看门狗复位、软件复位固定 56us。

PWRT 溢出以后，OST 开始计数延迟。复位时间(RESET Time) = PWRT + OST

OST 延迟完成以后，复位锁存器清零芯片开始正常运行。

PWRT(Power-up Reset TIMER)：上电复位计数器

上电复位计数器延迟时间由 OPTION 中的 SUT[2:0]设置，只要 PWRT 在运行，IC 就一直保持在复位状态。

VDD、温度和其它变化而会影响 PWRT 控制 IC 的延迟时间。

复位时间：

Oscillator Mode	POR/LVR/RST	WDT/SOFT/Reset
RC	16ms /8ms /256 ms /128ms	56us

OST(Oscillator Start-up TIMER)：振荡启动计数器

在复位过程中，PWRT 延迟时间(16/8/256/128ms)或固定复位时间之后振荡启动计数器会再提供一个 128 个周期的延迟以使内部 RC32M 时钟稳定，当 OST 计数时，IC 保持为复位状态，直到 OST 计满后复位撤除。

23. 程序下载和仿真

通过任意一组 PCL、PDA 及 VDD、VSS 四线连接来实现程序下载和在线仿真。
更多仿真功能的细节可参考仿真器的相关文档介绍。



24. 电气特性

24.1 电气特性极限参数

表 14 极限参数

参数	标号	条件	范围	单位
供电电压	V_{DD}	-	-0 to +6.0	V
输入电压	V_I	所有 I/O 口	-0.3 to $V_{DD} + 0.3$	V
工作温度	T_A	-	-40 to + 85	°C
储藏温度	T_{STG}	-	-50 to + 125	°C

24.2 直流特性

表 15 如无特殊说明 $V_{DD} = 1.8V \sim 5.5V$, Temp = 25°C

参数	标号	条件	最小值	典型值	最大值	单位
工作电压	V_{DD}	ADC、Touch DISable, $F_{CPU}=8MHz$	1.8		5.5	V
		ADC、Touch DISable, $F_{CPU}=16MHz$	2.4		5.5	V
		ADC ENable, $F_{CPU}=8MHz$	2.7		5.5	V
		TOUCH ENable, $F_{CPU}=8MHz$	2.5		5.5	V
CPU 时钟	F_{CPU}	$V_{DD}=2.5V \sim 5.5V$		16		MHz
		$V_{DD}=1.8V \sim 5.5V$		8		
输入高电压	V_{IH1}	$V_{THS}=0$	$0.7V_{DD}$			V
输入低电压	V_{IL1}	$V_{THS}=0$			$0.3V_{DD}$	V
输入高电压	V_{IH2}	$V_{THS}=1$	$0.5V_{DD}$			V
输入低电压	V_{IL2}	$V_{THS}=1$			$0.2V_{DD}$	V
输出拉电流	I_{OH}	$V_{DD}=5V$, $VOH=0.7V_{DD}$ OLDIS=0~7		2		mA
				5		
				8		
				11		
				14		
				20		
				30		
				100		
输出灌电流	I_{OL}	$V_{DD}=5V$, $VOL=0.3V_{DD}$ PUDIS=0~3		5		mA
				15		
				60		
				120		
输入强上拉电阻	RPH1	$V_{in}@V_{DD}/2$, $V_{DD}=5V$, IOPHS=0		25		K Ω
输入弱上拉电阻	RPH2	$V_{in}@V_{DD}/2$, $V_{DD}=5V$ IOPHS=1		37.5		K Ω
输入强下拉电阻	RPL1	$V_{in}@V_{DD}/2$, $V_{DD}=5V$, IOPHS=0		25		K Ω
输入弱下拉电阻	RPL2	$V_{in}@V_{DD}/2$, $V_{DD}=5V$ IOPHS=1		37.5		K Ω

WDT 电流	IWDT	$V_{DD}=3V$		0.8		μA
		$V_{DD}=5V$		1.1		
LVR/LPVT 电流	ILVR	$V_{DD}=3V$		1.1		μA
		$V_{DD}=5V$		1.5		
正常工作模式电流	INM	$V_{DD}=5V$, ADC enable 0.5mA 档		4		mA
		$V_{DD}=5V$, ADC disable		3.5		
		$V_{DD}=3V$, ADC enable 0.5mA 档		3.0		
		$V_{DD}=3V$, ADC disable		2.5		
低功耗模式电流	ISM	STOP mode, $V_{DD}=3V$, WDT enable, LVR/LPVT enable		3.0		μA
		STOP mode, $V_{DD}=3V$, WDT disable, LVR/LPVT enable		2.0		
		STOP mode, $V_{DD}=5V$, WDT enable, LVR/LPVT enable		4.0		
		STOP mode, $V_{DD}=5V$, WDT disable, LVR/LPVT enable		3.0		
		STOP mode, $V_{DD}=3V$, WDT enable, LVR/LPVT enable, RTC enable		5.0		

24.3 振荡器特性

表 16 如无特殊说明 $V_{DD}=5.0V$, $T_A=25^{\circ}C$

参数	标号	条件	最小值	典型值	最大值	单位
高速时钟频率		$V_{DD}=5.0V/25^{\circ}C$	31.68	32	32.32	MHz
高速时钟精度		$V_{DD}=5.0V/25^{\circ}C$	-0.5	—	0.5	%
		$V_{DD}=2.2V\sim 5.5V/25^{\circ}C$	-1.5	—	+1.5	%
		$V_{DD}=2.2V\sim 5.5V/-40\sim 85^{\circ}C$	-2	—	+2	%
高速时钟电流		$V_{DD}=5.0V$	—	400	600	μA
低速时钟频率		$V_{DD}=5.0V/25^{\circ}C$	128.38	131.072	133.73	KHz
低速时钟精度		$V_{DD}=5.0V/25^{\circ}C$	-0.5	—	0.5	%
		$V_{DD}=2.2V\sim 5.5V/25^{\circ}C$	-1.5	—	+1.5	%
		$V_{DD}=2.2V\sim 5.5V/-40\sim 85^{\circ}C$	-2.5	—	+2.5	%
低速时钟电流		$V_{DD}=5.0V$	—	1.5	2	μA
RTC 时钟电流		$V_{DD}=3.0V$	—	1	2	μA

24.4 ADC特性

表 17 如无特殊说明 $V_{DD} = 5.0V$, $T_A = 25^{\circ}C$

参数	标号	条件	最小值	典型值	最大值	单位
工作电压范围	V_{AD}		1.8	5.0	5.5	V
精度	N_R	$V_{REF}=5.0V$	---	11	12	bit
输入电压	V_{AIN}		GND	---	V_{Ref}	V
输入电阻	R_{AIN}	$V_{IN}=5.0V$	2	---	---	$M\Omega$
内部参考电压	V_{REFIN}	$V=5.0V$	1.26 2.02 3.04 4.05	1.28 2.048 3.072 4.096	1.3 2.07 3.1 4.15	V
内部参考电压温度系数	TC_{vref}				60	ppm/ $^{\circ}C$
外部参考电压	V_{REF}		1.28	---	VDD	V
输入源推荐阻抗	Z_{AIN}		---	---	1	$k\Omega$
A/D 转换电流	I_{AD}	ADC 模块工作, $V_{DD}=5.0V$	---	1.0	1.5	mA
A/D 输入电流	I_{ADIN}	$V_{DD}=5.0V$	---	---	10	μA
微分非线性误差	D_{LE}	$V_{DD}=5.0V$, $V_{REF}=5.0V$ $ADC_CLK \leq 2MHz$	---	---	± 1	LSB
积分非线性误差	I_{LE}	$V_{DD}=5.0V$, $V_{REF}=5.0V$ $ADC_CLK \leq 2MHz$	---		± 2	LSB
满刻度误差	E_F	$V_{DD}=5.0V$, $V_{REF}=5.0V$ $ADC_CLK \leq 2MHz$	---	± 1	± 3	LSB
偏移量误差	E_Z	$V_{DD}=5.0V$, $V_{REF}=5.0V$ $ADC_CLK \leq 2MHz$	---	± 0.5	± 2	LSB
总绝对误差	E_{AD}	$V_{DD}=5.0V$, $V_{REF}=5.0V$ $ADC_CLK \leq 2MHz$	---	---	± 3	LSB
ADC 时钟周期	t_{AD}		0.3	---	80	μs
ADC 采样时间	t_{SAMP}		2	4	---	μs
总转换时间	T_{CON}		16	---	29	t_{AD}

25. 封装尺寸图

25.1 SSOP28封装

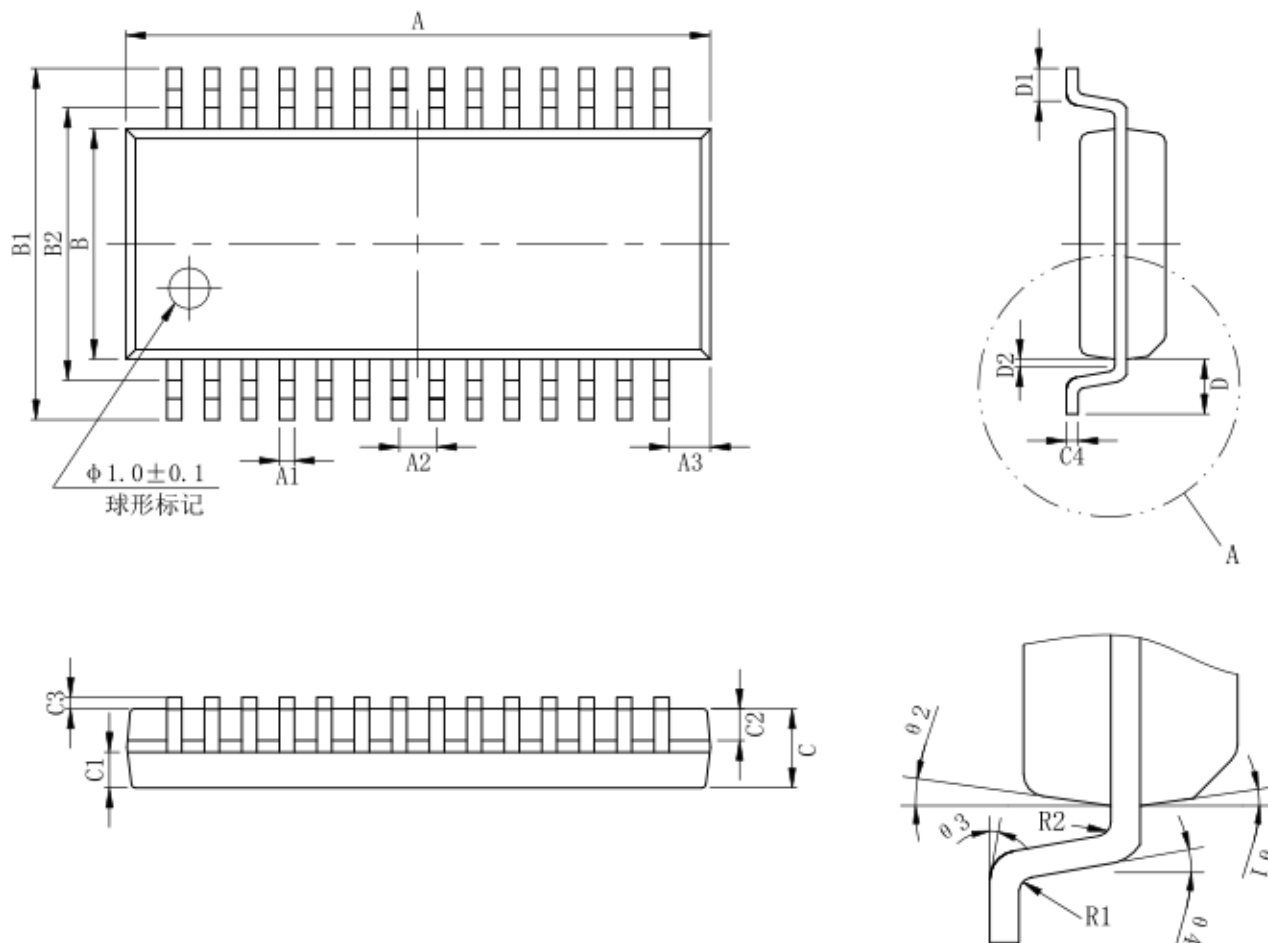


图 15 SSOP28 封装图

表 18 SSOP28 封装尺寸

符号	尺寸 (mm 单位)			符号	尺寸 (mm 单位)		
	最小值	典型值	最大值		最小值	典型值	最大值
A	9.8	-	10.00	C3	0.05	-	0.25
A1	-	0.254	-	C4	0.203	-	0.233
A2	-	0.635	-	D1	0.40	-	0.70
A3	-	0.695	-	D	1.05		
B	3.85	-	3.95	θ 1	8°~12°TYP4		
B1	5.84	-	6.24	θ 2	8°~12°TYP4		
B2		5.00-		θ 3	0°~8°		
C	1.4	-	1.6	R1	0.20TYP		
C1	0.61	-	0.71	R2	0.20TYP		
C2	0.54	-	0.64				

25.2 QFN24封装

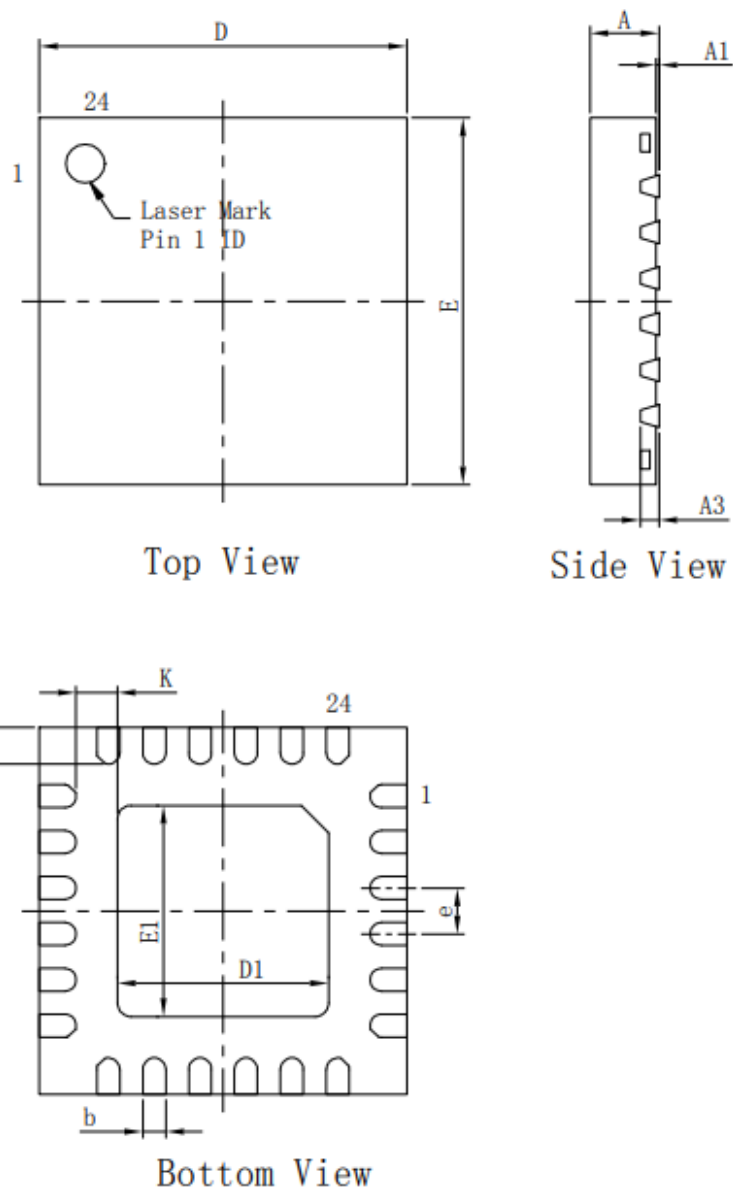


图 16 QFN24 封装图

表 19 QFN24 封装尺寸

符号	尺寸 (mm 单位)			符号	尺寸 (mm 单位)		
	最小值	典型值	最大值		最小值	典型值	最大值
A	0.70	0.75	0.80	D1	2.35	2.45	2.55
A3	0.203			E1	2.35	2.45	2.55
b	0.20	0.25	0.30	e	0.50		
D	3.90	4.00	4.10	K	0.365	0.375	0.385
E	3.90	4.00	4.10	L	0.30	0.40	0.50

26. 历史记录

版本号	修改记录	发布日期
V1.0	初版	2024-10-25
V1.1	修改16M最低工作电压点	2025-02-13
V1.2	更新QFN24脚位图 删除SSOP20,SOP28封装	2025-02-17
V1.3	TOUCH模块添加LDO描述	2025-04-18
V1.4	添加触摸复用LED驱动时，COM引脚选择SFR	2025-04-28
V1.5	更新GPIO描述笔误	2025-09-15

最终版本以官网为准，请及时下载查阅！