

PCF8563T (I²C) 时钟/日历芯片的原理与应用

摘要: 本文概括介绍了 PCF8563 时钟芯片的特点和基本组成, 通过实例详细说明了有关功能的应用软件。关于 PCF8563 各寄存器的详细位控功能请参考 PHILIPS 公司的相应产品资料。

概述

PCF8563 是 PHILIPS 公司推出的一款带 I²C 总线, 具有极低功耗的多功能时钟/日历芯片。PCF8563 的多种报警功能、定时器功能、时钟输出功能以及中断输出功能能完成各种复杂的定时服务, 甚至可为单片机提供看门狗功能。内部时钟电路、内部振荡电路、内部低电压检测电路 (1.0V) 以及两线制 I²C 总线通讯方式不但使外围电路及其简洁而且也增加了芯片的可靠性。当然作为时钟芯片, PCF8563 亦解决了 2000 年问题。因而, PCF8563 是一种性价比极高的时钟芯片, 它广泛应用于电话、传便携式仪器以及电池供电的仪器仪表等产品领域。下面将主要的性能指标作一综合:

1. 宽电压范围 1.0~5.5V, 复位电压 $V_{low}=1.0V$;
2. 超低功耗: 典型值为 $0.25\mu A$;
3. 四种报警功能和定时器功能;
4. 内部复位电路、内部振荡器电路和内部低压检测电路;
5. 中断输出和可编程时钟输出功能;
6. 400kHz I²C 总线。

PCF8563 的基本组成和工作原理

1. PCF8563 的管脚排列及描述如下图及表所示



2. PCF8563 内部寄存器

PCF8563 共有 16 个寄存器, 其中 00H~01H 为控制方式寄存器; 09H~0CH 为报警功能寄存器; 0DH 为时钟输出寄存器; 0EH 和 0FH 为定时器功能寄存器; 02H~08H 为秒~年时间寄存器。各寄存器的位描述综合于下表。

二进制格式寄存器概况

地址	寄存器名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00H	控制/状态寄存器 1	TEST	0	STOP	0	TESTC	0	0	0
01H	控制/状态寄存器 2	0	0	0	TI/TP	AF	TF	AIE	TIE
0DH	CLKOUT 输出寄存器	FE	-	-	-	-	-	FD1	FD0
0EH	定时器控制寄存器	TE	-	-	-	-	-	TD1	TD0
0FH	定时器倒数计数数值寄存器	定时器倒数计数数值(二进制)							

BCD 格式寄存器概况

地址	寄存器名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
02h	秒	VL	00～59BCD 码格式数						
03h	分钟	－	00～59BCD 码格式数						
04h	小时	－	－	00～59BCD 码格式数					
05h	日	－	－	01～31BCD 码格式数					
06h	星期	－	－	－	－	－	0～6		
07h	月/世纪	C	－	－	01～12 BCD 码格式数				
08h	年	00～99 BCD 码格式数							
09h	分钟报警	AE	00～59 BCD 码格式数						
0Ah	小时报警	AE	－	00～23 BCD 码格式数					
0BH	日报警	AE	－	01～31 BCD 码格式数					
0CH	星期报警	AE	－	－	－	－	0～6		

注:标明“—”的位无效

PCF8563 与微控制器的接口软件及功能应用举例

按 I²C 总线规约, PCF8563 有唯一的器件地址 A2H。下面首先给出基本的接口软件, 然后举例说明各种功能的应用。

1. 时钟的读取和写入

读时钟: 下面的程序将秒~年共七个字节的时间信息读出并放入 40H 为首址的接收缓冲区中。注意, 时间读出后需进行整理(屏蔽无效位)方能得出正确的信息。

RCV8563:

```

MOV     SlvAdr, #0A2H    ;取器件地址
MOV     SubAdr, #02H     ;取读时间的首字节地址(从秒开始读)
MOV     ByteCnt, #7      ;读七个时间信息
LCALL   RcvData          ;读取时间并放入接收缓冲区中
MOV     A, 40H           ;取秒字节
ANL     A, #7FH          ;屏蔽无效位
MOV     40H, A
MOV     A, 41H           ;取分钟字节
ANL     A, #7FH          ;屏蔽无效位
MOV     41H, A
MOV     A, 42H           ;取小时字节
ANL     A, #3FH          ;屏蔽无效位
MOV     42H, A
MOV     A, 43H           ;取天字节
ANL     A, #3FH          ;屏蔽无效位
MOV     43H, A
MOV     A, 44H           ;取星期字节
ANL     A, #07H          ;屏蔽无效位
MOV     44H, A
MOV     A, 45H           ;取月字节
ANL     A, #1FH          ;屏蔽无效位
MOV     45H, A
RET

```

写时钟：下面的程序将 2000 年 6 月 20 日星期 3 下午 3 点（15 点）59 分 30 秒的时间写入 PCF8563。

SEND8563:

```
ACALL  LOAD8563      ;将时间装入发送缓冲区(首址为 50H) 中
MOV    S1vAdr, #0A2H ;取器件地址
MOV    SubAdr, #00H  ;取写入寄存器的首字节地址(从 00H 开始写)
MOV    ByteCnt, #9    ;写七个时间信息和 2 个控制命令
LCALL  SendData      ;写时间
RET
```

LOAD8563:

```
MOV    50H, #00H      ;启动时钟
MOV    51H, #1FH      ;设置报警及定时器中断, 定时器中断为脉冲形式
MOV    52H, #30H      ;以下分别将秒至年的时间写入发送缓冲区中
MOV    53H, #59H
MOV    54H, #15H
MOV    55H, #20H
MOV    56H, #02H
MOV    57H, #06H
MOV    58H, #00H
RET
```

以上程序调用了 I²C 总线的基本数据发送(SendData)和接收(RcvData)模块, 这些模块的源程序清单在附录中给出。

下面的程序亦使用了这些模块。

2. 主要功能的应用

PCF8563 是一多功能时钟芯片, 必须谨慎的使用这些功能(其中最主要的就是正确的设置功能参数), 否则会产生意外的错误。下面给出一些可能会用到的设置程序。

A. 报警功能的设置

PCF8563 共有四种报警方式, 分别为小时报警(每小时的同一分钟时刻报警)、天报警(每天的同一小时时刻报警)、月报警(每月的同一天时刻报警)和星期报警(每星期的同一天时刻报警)。发生报警时 AF 位变为 1。**四种报警只可设置其一而不能同时使用。**设置报警有效的方法是将相应报警寄存器的最高位 AE 置 1。若同时置 AIE=1 则在 AF 置 1 的同时将在/INT 引脚产生一个中断(低电平有效), 清除中断信号的方法是软件清 AF。由此看出, AIE 相当于单片机中的中断允许控制位, 而 AF 相当于中断申请标志位。

例: 让 PCF8563 在每小时的 30 分钟时产生报警并在/INT 端产生一个中断给单片机 P87LPC764。

取原控制信息(目的是不破坏原来的配置):

```
MOV    S1vAdr, #0A2H ;取器件地址
MOV    SubAdr, #01H  ;取中断控制字节地址
MOV    ByteCnt, #1
LCALL  RcvData      ;读中断控制字节信息
```

中断配置:

```
MOV    A, 40H
ORL    A, #02H      ;置 AIE=1
MOV    50H, A
MOV    SubAdr, #01H ;取中断控制字节地址
MOV    ByteCnt, #1
```

```
LCALL SendData ;送中断控制字节命令
```

报警配置:

```
MOV 50H, #30H ;30 分报警时刻送发送缓冲区
MOV SubAdr, #09H ;取小时报警控制字节地址
MOV ByteCnt, #1
LCALL SendData ;送报警信息
```

以上配置完成后,即可在/INT 脚产生中断信号,在软件清除 AF 位之前,该中断信号一直有效。清除中断信号的程序如下:
取原控制信息(目的是不破坏原来的配置):

```
MOV SlvAdr, #0A2H ;取器件地址
MOV SubAdr, #01H ;取中断控制字节地址
MOV ByteCnt, #1
LCALL RcvData ;读中断控制字节信息
```

中断清除:

```
MOV A, 40H
ANL A, #17H ;设置成 AF=0,但保持其它位不变
MOV 50H, A
MOV SubAdr, #01H ;取中断控制字节地址
MOV ByteCnt, #1
LCALL SendData ;送中断清除命令
```

B. 定时器功能的设置

PCF8563 的定时器为倒数计时器,当 TE=1 时有效,倒计数值为 0FH 中的二进制数,当倒计数值计为 0 时 TF 位置 1。若置 TIE=1,则在 TF 置 1 的同时将在/INT 引脚产生一个中断(低电平有效)。与报警中断不同的是,定时器中断信号有两种方式,由 TI/TP 位控制。设置 TI/TP=0,中断信号和报警中断信号相同均为低电平方式,置 TF=0 可清除中断信号。设置 TI/TP=1,则中断信号为脉冲方式,其脉冲(低电平)宽度约为 15ms,此时可不考虑 TF 位的影响。由此看出, TIE 相当于单片机中的定时中断允许控制位,而 TF 相当于定时中断申请标志位。

注: 定时器功能可以和报警功能同时有效。

例: 让 PCF8563 每秒钟产生一次报警并在/INT 端产生一个脉冲给单片机 P87LPC764。在中断服务程序中可以读取时钟以供显示(这是显示时钟的方法之一)。

取原控制信息(目的是不破坏原来的配置):

```
MOV SlvAdr, #0A2H ;取器件地址
MOV SubAdr, #01H ;取中断控制字节地址
MOV ByteCnt, #1
LCALL RcvData ;读中断控制字节信息
```

中断配置:

```
MOV A, 40H
ORL A, #01H
MOV 50H, A
MOV SubAdr, #01H ;取中断控制字节地址
MOV ByteCnt, #1
LCALL SendData ;送中断控制字节命令
```

定时配置:

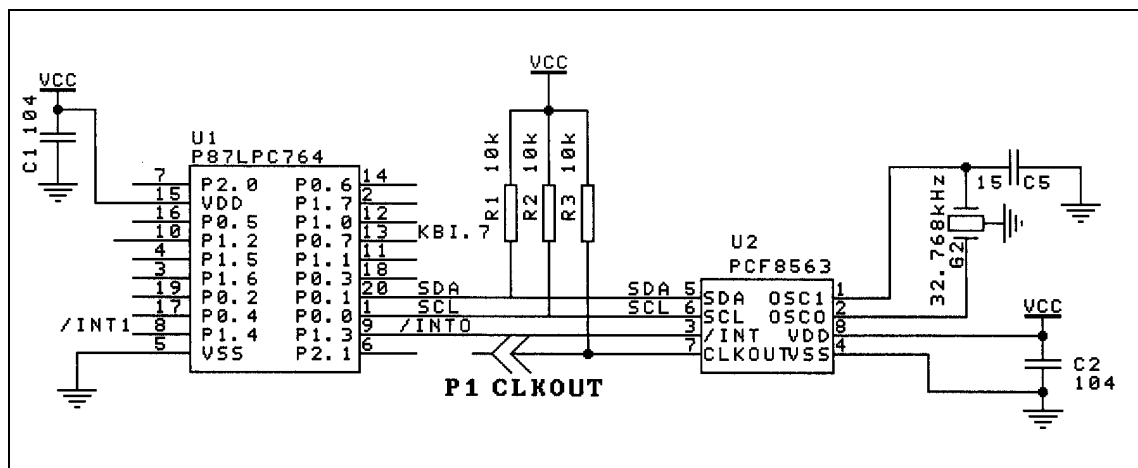
```
MOV    50H, #81H    ;启动定时器命令及时钟频率(64Hz)送发送缓冲区
MOV    51H, #64      ;倒计数值为64
MOV    SubAdr, #0EH  ;取定时器控制字节首地址
MOV    ByteCnt, #2    ;写两个字节
LCALL  SendData      ;写8563
```

以上配置完成后,即可在/INT脚产生周期为1s的脉冲中断信号。清除脉冲中断的方法有三:即将TIE、TE或0FH寄存器三者中任一的内容清0即可。

C. 时钟输出功能的应用

例:在PCF8563的CLKOUT脚输出一32.768kHz的方波

```
MOV    50H, #80H      ;时钟输出使能命令及32.768kHz频率选择送发送缓冲区
MOV    SubAdr, #0DH    ;取时钟输出控制字节地址
MOV    ByteCnt, #1     ;写一个字节
LCALL  SendData      ;开始时钟输出
```



注:电容C5的取值范围为1~20pF

PCF8563应用电路原理图

附录: I²C总线数据发送与接收模块源程序清单

; I²C总线模块化程序 ZM-I²C -5124-V0.1

; 87LPC762/4单主控器模拟I²C总线发送接受数据程序

;说明:本程序是利用87LPC764单片机的普通I/O口(如P0.1/P0.0)模拟实现I²C总线的功能,对I²C总线上的器件(本程序采用PCF8563)进行读写操作。

; 被控器地址在SlvAdr,单元地址在SubAdr中,所发送的数据字节数在ByteCnt中,所发送的数据在XmtDat中,所接收的数据在RcvDat中。

;87LPC762/4单主控器模拟I²C总线发送接受数据程序头文件

;器件地址

PCF8563 EQU 0A2H

;内存数据定义

```

BitCnt      DATA      30H      ; I2C 数据位计数器
ByteCnt     DATA      31H      ; I2C 数据字节计数器
SlvAdr      DATA      32H      ;被控器地址
SubAdr      DATA      33H      ;被控器单元地址
RcvDat      DATA      40H      ;接收数据缓冲区
XmtDat      DATA      50H      ;发送数据缓冲区

```

;端口位定义

```

SDA         BIT        P0.1      ;模拟 I2C 数据传送位
SCL         BIT        P0.0      ;模拟 I2C 时钟控制状态标志
ACK         BIT        20H      ;接收数据非应答标志

```

; 发送数据程序

;名称:SendData

;描述:发送 (ByteCnt) 个字节给被控器 PCF8563

;被控器地址在 SlvAdr 中, 单元地址在 SubAdr 中

;所发送数据的字节数 ByteCnt 在中, 发送的数据在 XmtDat 缓冲区中

;发送数据正常, 返回标志 F0=0, F0=1 表示被控器无应答或损坏。

SendData:

```

ACALL  START      ;发送 I2C 总线起始条件
MOV    A, SlvAdr   ;取被控器总线地址
ACALL  SentByte    ;发送被控器总线地址
JB     F0, SendReturn ;出错返回
MOV    A, SubAdr   ;取单元地址
ACALL  SentByte    ;发送单元地址
JB     F0, SendReturn ;出错返回
MOV    R0, #XmtDat ;取发送数据缓冲区首址

```

SentNext:

```

MOV    A, @R0
ACALL  SentByte    ;发送一次数据
JB     F0, SendReturn ;出错返回
INC    R0          ;取下一个数据
DJNZ   ByteCnt, SentNext ;重复操作直到发送完最后一个数据
ACALL  STOP        ;发送 I2C 总线停止条件

```

Delay10:

```

MOV    R7, #30H      ;延时 10ms, 等待数据写完
D1:
MOV    R6, #34H
D2:
DJNZ   R6, D2
DJNZ   R7, D1

```

SendReturn:

RET

; 接收数据程序;

;名称:RcvData:

;描述:从被控器 PCF8563 接收 (ByteCnt) 个字节数据

;被控器地址在 SlvAdr 中, 单元地址在 SubAdr 中

;所接收数据的字节数 ByteCnt 在中, 接收的数据在 XmtDat 缓冲区中

;接收数据正常, 返回标志 F0=0, F0=1 表示被控器无应答或损坏。

RcvData:

ACALL	START	;发送 I ² C 总线起始条件
MOV	A, SlvAdr	;取被控器总线地址
ACALL	SentByte	;发送被控器总线地址
JB	F0, RcvReturn	;出错返回
MOV	A, SubAdr	;取单元地址
ACALL	SentByte	;发送单元地址
JB	F0, RcvReturn	;出错返回
ACALL	START	;发送 I ² C 总线重复起始条件
MOV	A, SlvAdr	;取被控器总线地址
SETB	ACC. 0	;取总线读操作位
ACALL	SentByte	;发送被控器总线地址
JB	F0, RcvReturn	;出错返回
CLR	ACK	;清接收非应答标志, 准备接收数据
MOV	R0, #RcvDat	;取接收数据缓冲区首址
DJNZ	ByteCnt, RcvNext	;如果还未接收数据则转去接收前 (ByteCnt)-1 个数据
SJMP	RcvLast	;只剩最后一个数据未接收转去接收最后一个数据

RcvNext:

ACALL	RcvByte	;接收数据
MOV	@R0, A	;将数据放入接收缓冲区中
INC	R0	;指向下一个缓冲区地址
DJNZ	ByteCnt, RcvNext	;重复操作直到剩下最后一个数据

RcvLast:

SETB	ACK	;置接收非应答标志, 准备在接收完最后一个数据后发出非应答信
		;号以停止接收数据
ACALL	RcvByte	;接收最后一个数据
MOV	@R0, A	;将最后一个数据放入接收缓冲区中
ACALL	STOP	;发送 I ² C 总线停止条件

RcvReturn:

RET

; 子程序

;名称:START

;描述:启动 I²C 总线子程序—发送 I²C 起始条件

START:

```

    SETB  SDA          ;发送起始条件的数据信号
    NOP
    SETB  SCL          ;发送起始条件的时钟信号
    NOP              ;起始条件建立时间大于 4.7 μs
    NOP
    NOP
    NOP
    NOP
    CLR   SDA          ;发送起始信号
    NOP              ;起始条件锁定时间大于 4 μs
    NOP
    NOP
    NOP
    CLR   SCL          ;钳住 I2C 总线，准备发送或接收数据
    RET

```

;名称:STOP

;描述:停止 I²C 总线子程序—发送 I²C 总线停止条件

STOP:

```

    CLR   SDA          ;发送停止条件的数据信号
    NOP
    SETB  SCL          ;发送停止条件的时钟信号
    NOP              ;起始条件建立时间大于 4 μs
    NOP
    NOP
    NOP
    NOP
    SETB  SDA          ;发送 I2C 总线停止信号
    NOP
    NOP
    NOP
    RET

```



```

;-----
;名称:SentByte
;描述:字节数据传送子程序: 发送一个字节数据或地址给被控器 PCF8563
;要发送的数据在 ACC 中
;发送数据正常, 返回标志 F0=0, F0=1 表示被控器无应答或损坏。
;-----
SentByte:
    MOV    BitCnt, #08H           ;要传送的数据长度为 8 位
SentB:
    RLC    A                     ;要发送的数据左移, 发送位入 C
    JC     Sent1                 ;发送位为“1”转发送数据位“1”
    CLR    SDA                   ;发送数据位“0”
    SJMP   Sent                 ;转发送数据位
Sent1:
    SETB   SDA                   ;发送数据位“1”
    NOP
Sent:
    NOP
    SETB   SCL                   ;置时钟线为高, 通知被控器开始接收数据位
    NOP                               ;保证时钟高周期大于 4 μs
    NOP
    NOP
    NOP
    CLR    SCL                   ;钳住总线准备接收下一个数据位
    DJNZ   BitCnt, SentB         ;8 位没发送完继续发送
    NOP
    NOP
    SETB   SDA                   ;8 位发送完后释放数据线, 准备收应答位
    NOP
    NOP
    SETB   SCL                   ;开始接收应答信号
    NOP
    NOP
    CLR    F0                    ;预先清发送数据出错标志
    JNB    SDA, AckEnd           ;判断是否接收到应答信号, 正常转 AckEnd
    SETB   F0                    ;未收到应答, 置位错误标志
AckEnd:
    NOP
    CLR    SCL                   ;发送结束钳住总线, 准备下一步发送或接收数据或进行其它处理
    RET

```

```

;-----
;名称:RcvByte
;描述:字节数据接收子程序: 从被控器 PCF8563 接收一个字节数据, 然后根据主程序要求发送应答位
;接收的字节数据在 ACC 中
;-----
RcvByte:
    SETB  SDA                ;置数据线为输入方式
    MOV   BitCnt, #08H       ;要传送的数据长度为 8 位
Rcv:
    NOP
    CLR   SCL                ;置时钟线为低, 准备接收数据位
    NOP                ;时钟低周期大于 4.7 μs
    NOP
    NOP
    NOP
    NOP
    SETB  SCL                ;置时钟线为高使数据线上数据有效
    NOP
    NOP
    CLR   C                  ;准备将接收的数据位放入 ACC 中
    JNB   SDA, Rcv0          ;读数据位, 若为“0”则置 C=0 并转 Rcv0 接收数据位
    SETB  C                  ;读数据位, 若为“1”则置 C=1

Rcv0:
    RLC   A                  ;接收的数据位放入 ACC 中
    NOP
    NOP
    DJNZ  BitCnt, Rcv        ;8 位没收完继续接收
    CLR   SCL                ;8 位接收完置时钟线和数据线为低准备发送应答或非应答信号
    NOP
    NOP
    NOP
    CLR   SDA
    JNB   ACK, SentAckB      ;判断是否需要继续接收字节数据, 若需要则转 SentAckB 发送应答位
    SETB  SDA                ;发送非应答信号
SentAckB:
    ;以下程序发送应答位
    NOP
    NOP
    SETB  SCL                ;置时钟线为高使应答位有效
    NOP                ;时钟高周期大于 4 μs
    NOP
    NOP

```

NOP
NOP
CLR SCL ;清时钟线钳住 I²C 总线以便继续接收字接数据或发送停止条件
CLR ACK ;清接收非应答标志
RET

;

END