RTC　　使用手册

南京博芯电子技术有限公司

2009-04

否认书：
南京博芯电子技术有限公司保留对文档随时进行修改的权利，无须任何申明。南京博芯电子技术有限公司所提供的信息是精确可靠的。对于它的应用以及由于应用而导致违反专利权或是第三方的其他权利，本公司不负任何责任。

版本历史

日期	版本	描述	备注
2009-04	1.0	初稿	Jack

目　　录

版本历史.....2
目　　录.....3
一. RTC在SEP4020 中的位置.....4
二. RTC介绍.....4
2.1 功能介绍.....4
2.1.1 基本性能.....4
2.2 寄存器介绍.....5
三. 实现原理.....5
3.1 硬件原理.....5
3.1.1 接口定义.....5
3.2 软件原理.....7
3.2.1 头文件定义说明.....7
3.2.2 核心数据结构声明.....8
3.2.3 代码实现流程图.....8
3.2.4 主要函数及参数，返回值介绍.....8
四.测试说明.....12
4.1 测试流程.....12
4.2 测试结果.....12
五.其他注意事项.....13

一. RTC 在 SEP4020 中的位置

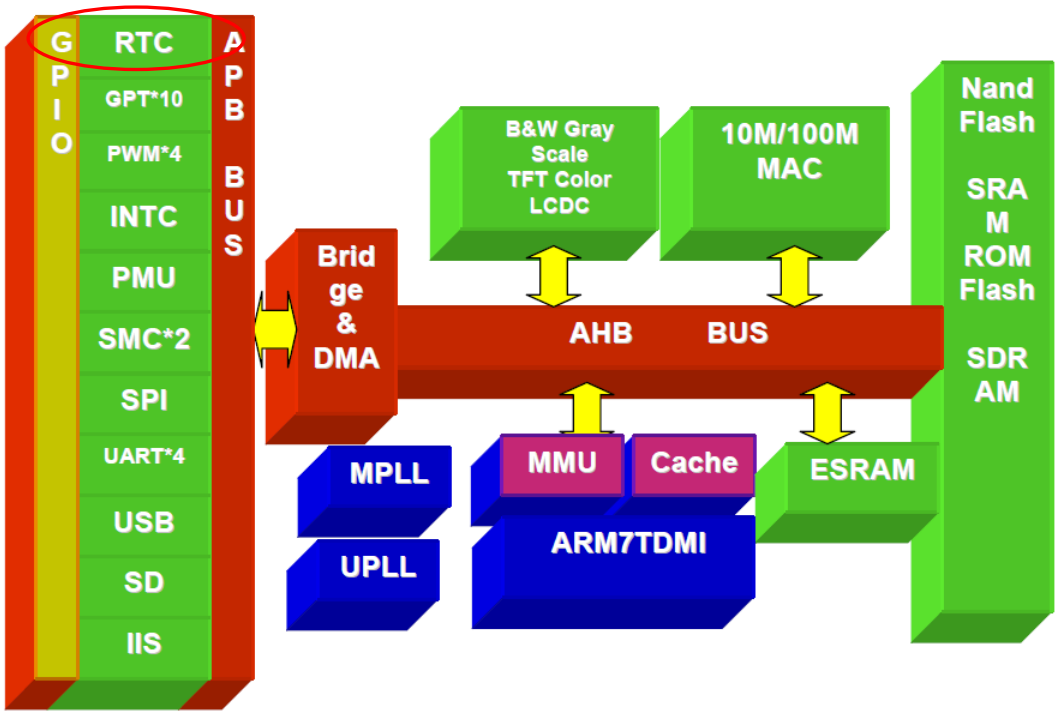


图 1　　RTC 在 SEP4020 中的位置

二. RTC 介绍

2.1 功能介绍

2.1.1 基本性能

（1）外部十进制输入，提供年，月，日，时，分，秒计时，时间表示采用BCD 编码。

（2）可设置定时中断。当前时间与设置时间相同时，RTC 即发出中断，提供月/日/小时/分钟的定时，不精确到秒。

（3）提供 WatchDog 功能。如果一次timeout 事件发生，以下两种选择：产生一次 WatchDog reset，系统复位。首先产生一次中断，如果在下次timeout 发生时还没有得到软件服务，产生WatchDogreset，系统复位。

（4）提供 1/256 秒～1 秒软件可配置的连续采样中断。实时操作系统可以使用此中断作为进程切换的时间单位。

（5）提供秒中断、分中断、采样中断、定时中断和 WatchDog 中断。

（6）支持 Pause 模式。设置一位寄存器pause。当pause = 0，芯片正常工作；当pause =1 时，WatchDog 计数器停止计数，软件对该位配0 可以使计数器继续计数，另外IRQ、FIQ 也可以使该位清零。

（7）提供闰年判断机制。判断每一个月的天数是 31 天、30 天、29 天还是28 天。在闰年判断上，直接设定2004 到2024 年为闰年，其它闰年年份硬件都不考虑。

2.2 寄存器介绍

RTC模块的基址：0x10002000

名称	偏移地址	复位值	描述
STA_YMD	0x00	0x00000000	年、月、日计数寄存器
STA_HMS	0x04	0x00000000	小时、分钟、秒寄存器
ALARM_ALL	0x08	0x00000000	定时月、日、时、分寄存器
CTR	0x0c	0x00000000	控制寄存器
INT_EN	0x10	0x00000002	中断使能寄存器
INT_STS	0x14	0x00000000	中断状态寄存器
SAMP	0x18	0x00000000	采样周期寄存器
WD_CNT	0x1C		Watch-Dog 计数值寄存器
CONFIG_CHECK	0x24	0x00000000	配置时间确认寄存器（在配置时间之前先写0xaaaaaaaa）
KEY0	0x2c	0x00000000	密钥寄存器 0

三. 实现原理

3.1 硬件原理

3.1.1 接口定义

该模块在 APB 总线上使用PCLK 总线时钟，在PAD 中使用CLK32 提供的时钟，所以RTC模块属于多时钟域设计。其中Sync_Pto32 与Sync_32toP 模块是保证两个时钟域同步而设置的。APB 接口模块完成数据从PAD 到APB 域的接口转换，CLK32 域的Time，Sample，WatchDog模块用来计时或形成中断。

对应的框图如下：

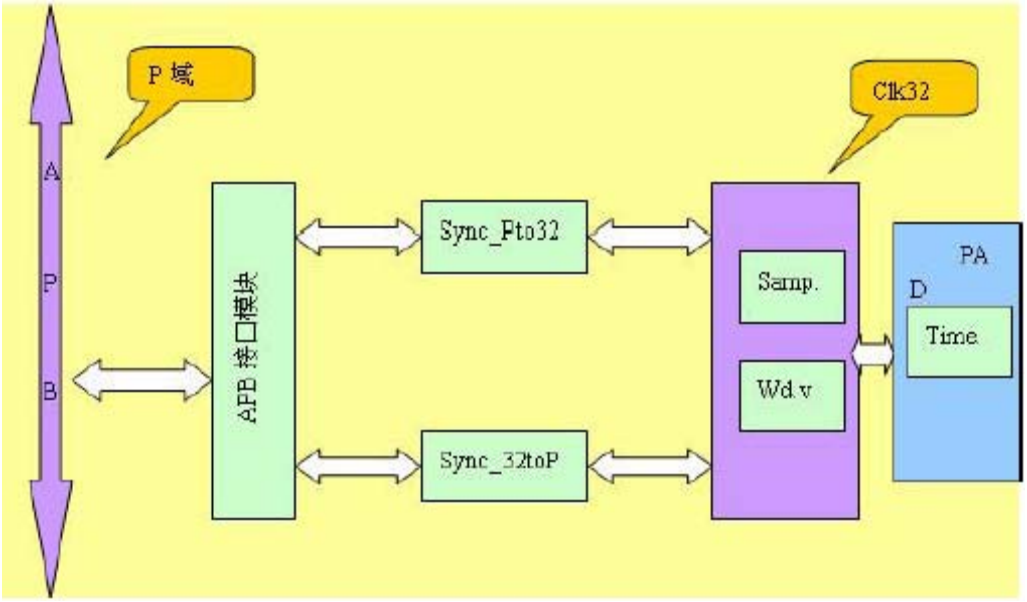


图　RTC原理结构图

WatchDog 模块用来计时或形成中断。图示如下：

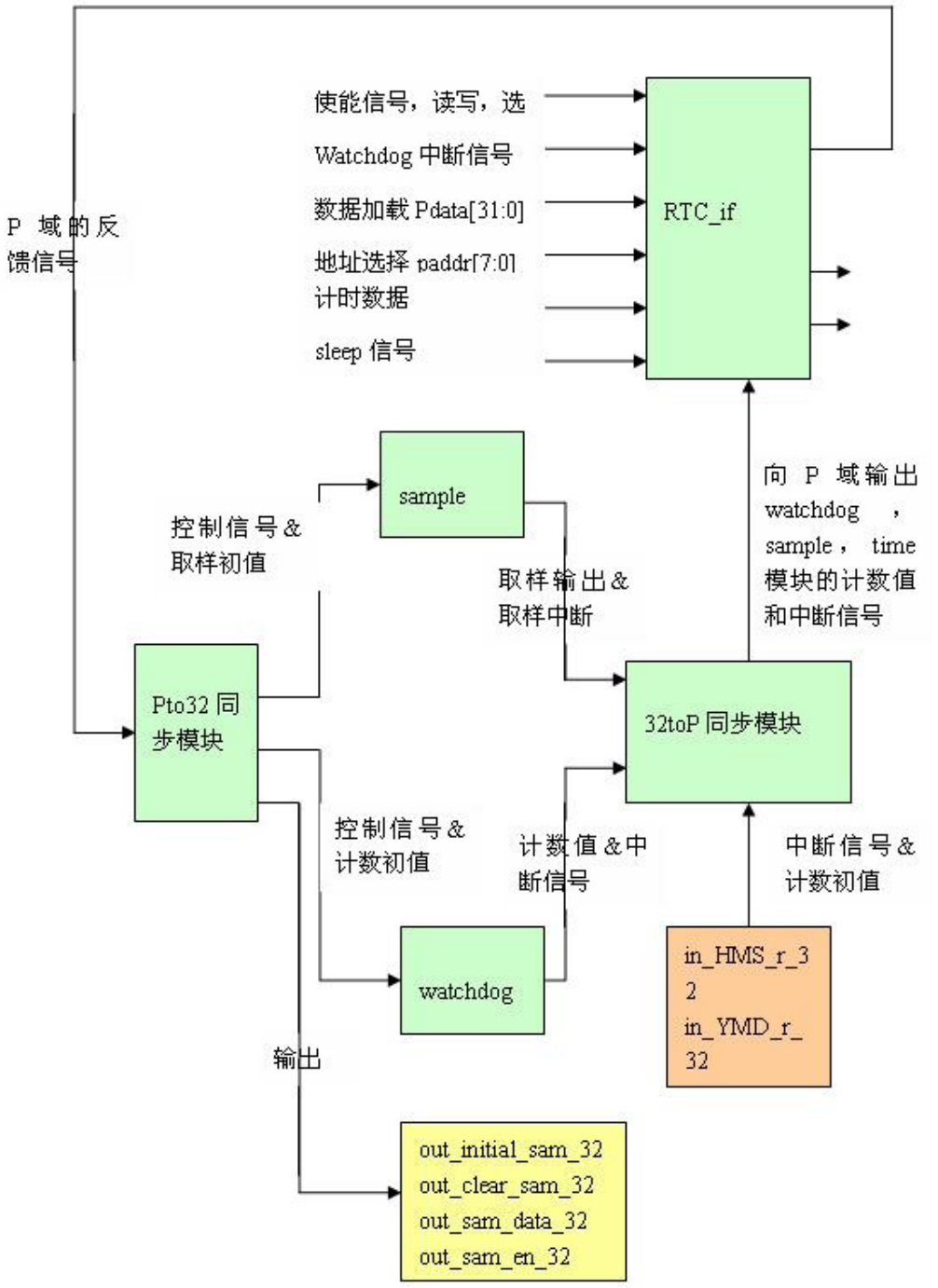


图　结构框图及接口信号

3.2 软件原理

3.2.1 头文件定义说明

#include　〈stdio.h〉　　标准输入输出函数库

#include　"sep4020.h"　　所有程序中用到的Typedef，Error Codes，PMU 模块时钟

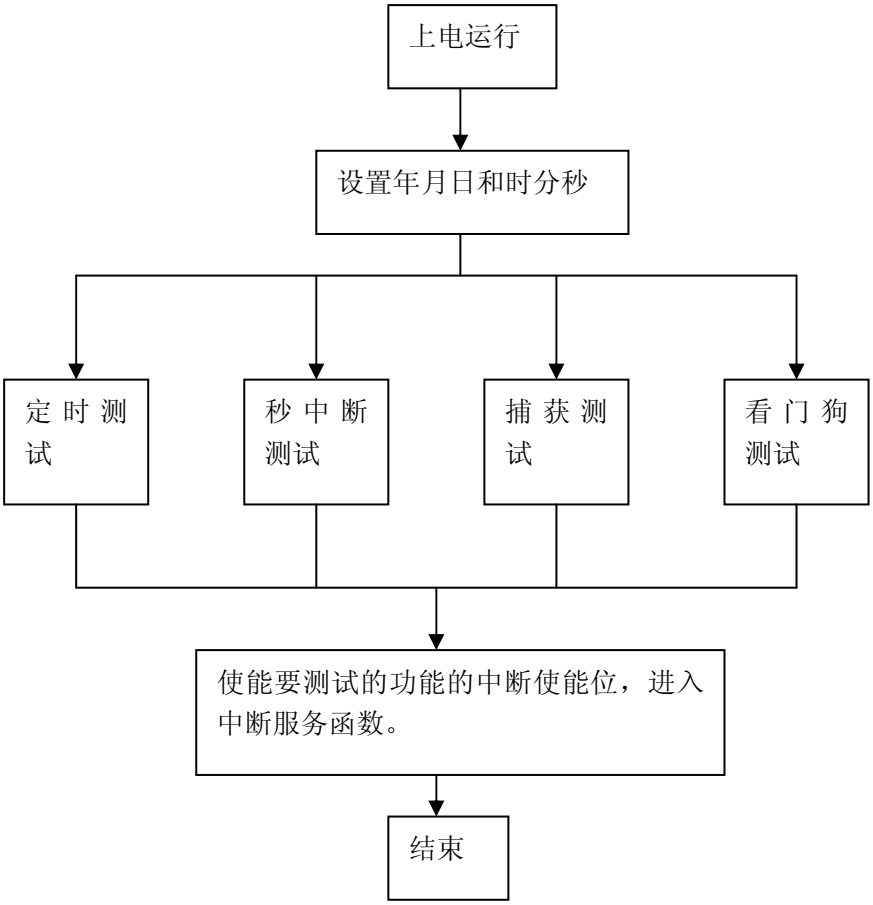
#include　"intc.h"　　INTC 模块的中断源，中断处理，定义中断的向量结构体

#include　"rtc.h"　　对rtc的操作函数

3.2.2 核心数据结构声明

INT_VECTOR　中断向量结构体包括：中断号和中断处理函数

3.2.3 代码实现流程图



3.2.4 主要函数及参数，返回值介绍

（1）int RTCSetDate(U32 year, U32 month, U32 day)

描述：设定RTC的年、月、日

输入：

U32 year:　　年

U32 month:　　月

U32 day:　　日

输出：　-1:error

0:ok

使用位置：

（2）void RTCGetDate(DATE * pdate)

描述：获得RTC的年、月、日

输入：struct RTCDateStruct * date 日期结构体指针

输出：无

使用位置：

（3）int RTCSetTime(U32 hour,U32 minute,U32 second)

描述：设定RTC的时、分、秒

输入：U32 hour: 小时

U32 minute: 分钟

U32 second: 秒钟

输出：-1:error

0:ok

（4）void RTCGetTime(TIME * ptime)

描述：获取RTC的时、分、秒

输入：struct RTCTimeStruct * time 时间结构体指针

输出：无

使用位置：

（5）void RTCSetAlarm(U32 month,U32 day,U32 hour, U32 minute)

描述：设定RTC的定时器

输入：U32 month 月

U32 day 天

U32 hour 时

U32 minute 分

输出： 无

使用位置：

（6）void RTCSetSample(U32 frequency)

描述：设定RTC的Sample计数值

输入： U32 frequency采样计数值。

计数时间 T = frequency/32768秒。

frequency = 0x0080， T = 1/256秒

frequency = 0x8000， T = 1秒

frequency = 0x0001， T = 0秒

输出：无

使用位置：

（7）void RTCEnableInt(U32 int_num)

描述：使能RTC的中断

输入：int_num:中断号

0:Sample

1:WatchDog

2:Second

3:Minute

4:Alarm

5:Reset

输出：无

使用位置：

（8）void RTCDisableInt(U32 int_num)

描述：不使能RTC的中断

输入： int_num:中断号

0:Sample

1:WatchDog

2:Second

3:Minute

4:Alarm

5:Reset

输出：无

使用位置：

(9) void RTCEnableSample(void)

描述：使能RTC的Sample功能

输入：无

输出：无

使用位置：

（10）void RTCDisableSample(void)

描述：不使能RTC的Sample功能

输入：无

输出：无

使用位置：

（11）void RTCWatchDogInit(U32 ClockSource,U32 ClockDivider,U32 ResetType)

描述：初始化WatchDog

输入：U32 ClockSource: WatchDog时钟源

0：RTC时钟（32.768k）

1：系统分频时钟

U32 ClockDivider：系统时钟分频因子

U32 ResetType：复位类型

0：WatchDog中断一次后复位

1：WatchDog中断两次后复位

输出：无

使用位置：

（12）void RTCWatchDogPause(U32 pause)

描述：暂停WatchDog

输入：U32 pause

1：暂停

0：继续

输出：无

使用位置：

（13）void RTCEnableWatchDog(void)

描述：使能RTC的WatchDog功能

输入：无

输出：无

使用位置：

（14）void RTCDisableWatchDog(void)

描述：不使能RTC的WatchDog功能

输入：无

输出：无

使用位置：

（15）void RTCSetWatchDog(U32 watchdog)

描述：设定RTC的WatchDog计数值

输入：U32 watchdog WatchDog计数值

输出：无

使用位置：

（16）void RTCWatchDogService(void)

描述：　　设定RTC的WatchDog,喂狗

输入：无

输出：无

使用位置：

（18）void RTCSetKey(U32 key)

描述：设定RTC的密钥

输入：U32 key：密钥值

输出：无

使用位置：

（18）void RTCIntHandler(void)

描述： RTC中断处理函数

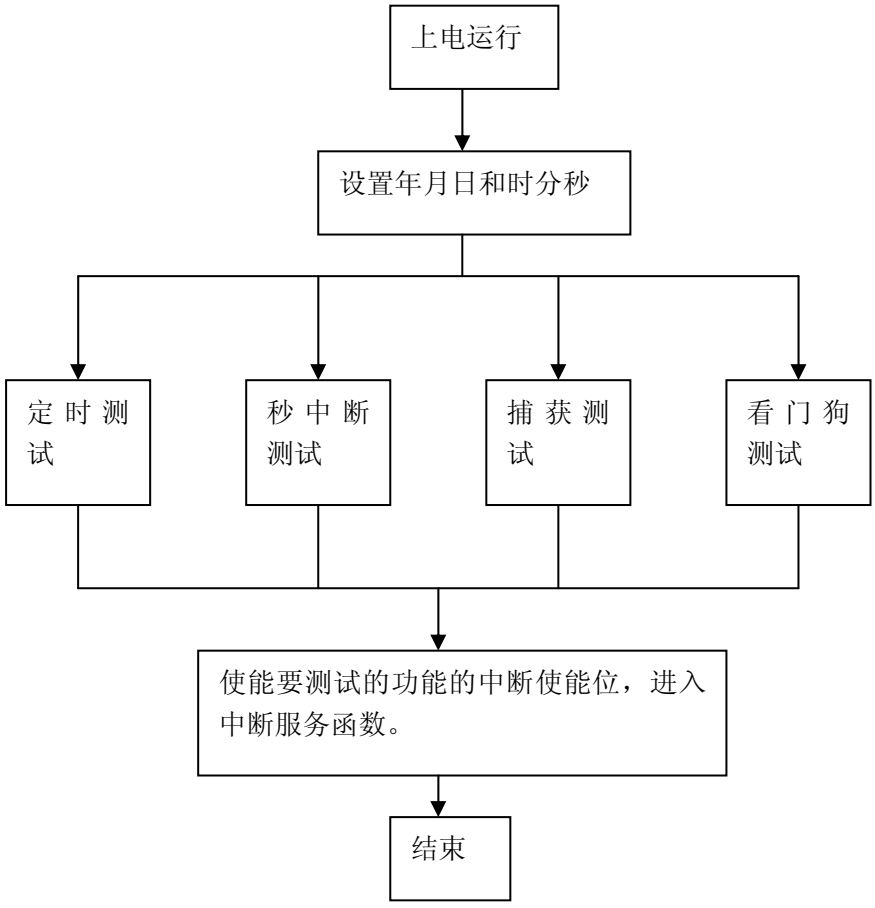
输入：无

输出：无

使用位置：

四.测试说明

4.1测试流程



4.2测试结果

在console窗口中说出：例如

12:50:0

there is one INT or more!

the INT number is 2(0:SAMPLE,1:WATCHDOG,2:SECOND,3:MINUTE,4:ALARM,5:RESET)

五.其他注意事项

配置时间确认寄存器在：配置时间 YMD HMS 之前config_check 先写0xAAAAAAAA，当时间配置完毕后注意要将config_check 配置寄存器关闭。

考虑到RTC 中两个时域的同步问题，年月日，时分秒寄存器设置新值后不能立即读取到设置的新值。建议经过1~2 个CLK32 时钟周期之后才可以正确读取到更新的新值。

考虑到RTC 中两个时域的同步问题，向中断状态寄存器中各标志位写“1”不能立即清除该标志位。建议经过1~2 个CLK32 时钟周期之后相应的写操作才能清除对应的标志位。

注意WatchDog 和Sample 使用前一定要重新写入计数值（包括使用系统上电默认值，也要手动写入），否则不能正常计数。

WatchDog 的使用请注意以下几点：

WatchDog 基础计数器计数范围是0x0~0x3fff，上层计数器的计数值由程序设置相应寄存器（WD_CNT）得到，基础计数器每计够一轮，上层计数器减一。当上层计数器计数值减到1 时，将会引发WatchDog 中断（在使能此中断的情况下），并且自动重装载WD 上层计数器的计数值。每次通过程序重新设定上层计数器的计数值和重装载值，不改变当前基础计数器的计数值。

WatchDog 基础计数器使用的工作时钟可以有两种选择：当RTC 使用第二晶振（通常是32.768KHz 晶振）时，直接使用第二晶振的频率，基础计数器计完一周所用时间的典型值为(1/32768)*0x3fff=0.5(s)；当系统没有为RTC 专门提供第二晶振的时候，RTC 所提供的所有功能中只有WatchDog 仍能使用，这时应将控制寄存器第24 位置1，选择总线的分频时钟为WD 工作时钟，控制寄存器中设定的分频因子决定WD 时钟与总线时钟的分频比，基础计数器计完一周所用时间跟总线时钟频率和WD 分频因子有关，须根据情况自行计算。

WatchDog 复位系统有两种方式：一次WD 中断引发复位，和两次WD 中断引发复位。常用的是两次WD 中断引发复位，这种方式的使用方法如下：使能WatchDog 中断，使能复位功能，选择两次中断引发复位模式，加载合适的计数值，然后在WatchDog 中断服务程序中给WD_INT_CNT_CLR（控制寄存器CTR 的第4 位）置1。如此设置后将会在每次WD 中断到来时进入服务程序，清除中断次数的累计，使系统继续运行，如果WD 硬件中断产生后得不到服务程序的清除，则下次中断到来时将引发系统复位。另一种变通的使用方法是：使能WatchDog 中断，使能复位功能，选择一次中断引发复位模式，加载计数值为2 秒，然后在秒中断服务程序中重置计数值。