

# SOFT\_RTC库

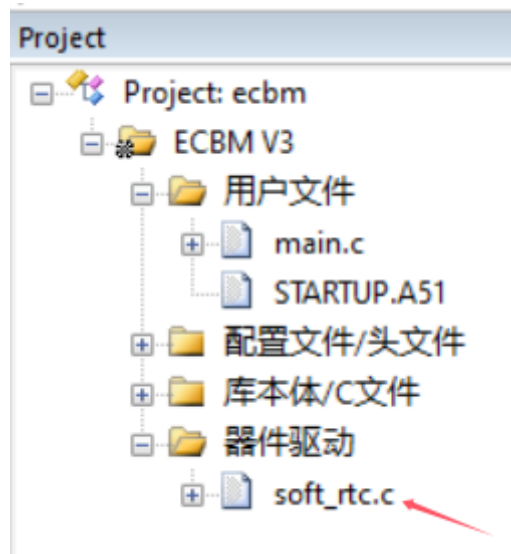
---

本库是以软件形式跑时间（RTC）函数库，同时也可作为外置RTC芯片的上层操作函数。比如常见的DS1302。

用于软件跑时间时，可以帮助一些没有RTC硬件的型号，完成简单的计时作用。其断电后时间会消失的特性亦可用在一些一次性模块或防拆功能上。

用于上层操作函数时，可以建立起校时上位机（比如最新版的STC-ISP）和外置RTC之间沟通的桥梁。

本库隶属于device，不是单片机本身自带的外设，因此需要手动将soft\_rtc.c添加到工程文件中：



推荐如图所示操作，在keil的左侧栏双击【器件驱动】，然后添加soft\_rtc.c进来。然后在main.c里添加头文件#include "soft\_rtc.h"即可。

## API

---

### soft\_rtc\_get\_day\_max

函数原型：u8 soft\_rtc\_get\_day\_max(u16 year,u8 month);

#### 描述

获取天数最大值函数。

#### 输入

- year：年份。
- month：月份。

#### 输出

无

## 返回值

- 该年份的该月最大的天数值。

## 调用例程

```
u8 day; //定义一个变量。  
day=soft_rtc_get_day_max(2024,9); //获取2024年9月的最大天数，day的值就是30。
```

## 注意事项

1. 这个函数主要是用于走时和设置日期函数。用来防止类似“2月份有31天”的奇葩情况出现。因此当运行函数和设置函数的使能都关闭的时候，本函数也会停止编译。

## soft\_rtc\_run

函数原型：void soft\_rtc\_run(void);

## 描述

软件RTC的运行函数。

## 输入

无

## 输出

无

## 返回值

无

## 调用例程

```
#include "ecbm_core.h" //加载库函数的头文件。  
#include "soft_rtc.h" //加载软件RTC库函数的头文件。  
bit one_sec_flag=0; //1秒标志。  
void main(void){ //main函数，必须的。  
    system_init(); //系统初始化函数，也是必须的。  
    timer_init(); //定时器初始化。  
    timer_set_timer_mode(0,1000); //设置定时器0为1ms。  
    timer_start(0); //启动定时器0。  
    while(1){  
        if(one_sec_flag){ //1秒标志为1时，执行。  
            one_sec_flag=0; //清除标志。  
            soft_rtc_run(); //运行软件RTC。  
        }  
    }  
}  
  
u16 div_1ms_1s; //降频变量，用于把1ms时间降为1s。  
void timer0_it_callback(void){ //定时器0中断回调函数。  
    div_1ms_1s++; //降频变量累加。  
    if(div_1ms_1s>=1000){ //加到1000的时候，就到了1秒。  
        div_1ms_1s=0; //清零。  
        one_sec_flag=1; //设置1秒标志。  
    }  
}
```

```
}  
}
```

## 注意事项

1. 本函数最好1秒执行一次，如果不是1秒执行一次，那时间肯定是不对了。
2. 本库不提供timer库的联动，请自行设置好定时器参数，定时器的设置可参考TIMER库.pdf。
3. 使用本函数之前先确认函数使能打开，如图所示：

Option	Value
SOFT-RTC运行函数使能	<input checked="" type="checkbox"/>
SOFT-RTC时间更新函数使能	<input checked="" type="checkbox"/>
SOFT-RTC时间设置函数使能	<input checked="" type="checkbox"/>

## soft\_rtc\_set\_date

函数原型：u8 soft\_rtc\_set\_date(u16 year,u8 month,u8 day,u8 week);

## 描述

设置日期函数。

## 输入

- year：年份，u16型基本可以随意输入，有生之年不存在千年虫问题。
- month：月份，可输入1~12。
- day：日期，可输入1~31，但不能超过当月的最大值，比如平年2月就不能超过28。
- week：星期，可输入1~7。

## 输出

无

## 返回值

- SOFTRTC\_OK：设置正确。
- SOFTRTC\_MON\_ERR：月份设置错误。
- SOFTRTC\_DAY\_ERR：日期设置错误。
- SOFTRTC\_WEEK\_ERR：星期设置错误。

## 调用例程

```
#include "ecbm_core.h" //加载库函数的头文件。  
#include "soft_rtc.h" //加载软件RTC库函数的头文件。  
bit one_sec_flag=0; //1秒标志。  
void main(void){ //main函数，必须的。  
    system_init(); //系统初始化函数，也是必须的。  
    timer_init(); //定时器初始化。  
    timer_set_timer_mode(0,1000); //设置定时器0为1ms。  
    timer_start(0); //启动定时器0。  
    soft_rtc_set_date(2024,9,14,6); //初始值设置为2024年9月14日星期六。  
    while(1){  
        if(one_sec_flag){ //1秒标志为1时，执行。  
            one_sec_flag=0; //清除标志。  
            soft_rtc_run(); //运行软件RTC。  
        }  
    }  
}
```

```

    }
}
}
u16 div_1ms_1s;//降频变量，用于把1ms时间降为1s。
void timer0_it_callback(void){//定时器0中断回调函数。
    div_1ms_1s++;//降频变量累加。
    if(div_1ms_1s>=1000){//加到1000的时候，就到了1秒。
        div_1ms_1s=0;//清零。
        one_sec_flag=1;//设置1秒标志。
    }
}
}

```

## 注意事项

1. 本函数用于设置日期，对于本机跑时间来说即时生效，对于外置RTC硬件来说，需要将那几个时间变量存到外置RTC硬件里才生效。具体的保存方法会在对应的RTC库里提到，比如DS1302库。
2. 本库不提供外置RTC库的联动，如使用了外置RTC芯片，请自行完善保存读取操作。
3. 使用本函数之前先确认函数使能打开，如图所示：

Option	Value
SOFT-RTC运行函数使能	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SOFT-RTC时间更新函数使能	<input checked="" type="checkbox"/>
SOFT-RTC时间设置函数使能	<input checked="" type="checkbox"/>

## soft\_rtc\_set\_time

函数原型：u8 soft\_rtc\_set\_time(u8 hour,u8 minute,u8 second);

## 描述

设置时间函数。

## 输入

- hour：小时，可输入0~23。
- minute：分钟，可输入0~59。
- second：秒钟，可输入0~59。

## 输出

无

## 返回值

- SOFTRTC\_OK：设置正确。
- SOFTRTC\_HOUR\_ERR：时设置错误。
- SOFTRTC\_MIN\_ERR：分设置错误。
- SOFTRTC\_SEC\_ERR：秒设置错误。

## 调用例程

```

#include "ecbm_core.h" //加载库函数的头文件。
#include "soft_rtc.h" //加载软件RTC库函数的头文件。
bit one_sec_flag=0; //1秒标志。
void main(void){ //main函数，必须的。

```

```

system_init();           //系统初始化函数，也是必须的。
timer_init();            //定时器初始化。
timer_set_timer_mode(0,1000); //设置定时器0为1ms。
timer_start(0);          //启动定时器0。
soft_rtc_set_date(2024,9,14,6); //初始值设置为2024年9月14日星期六。
soft_rtc_set_time(10,26,0); //初始值设置为10点26份0秒。
while(1){
    if(one_sec_flag){ //1秒标志为1时，执行。
        one_sec_flag=0; //清除标志。
        soft_rtc_run(); //运行软件RTC。
    }
}
}
u16 div_1ms_1s; //降频变量，用于把1ms时间降为1s。
void timer0_it_callback(void){ //定时器0中断回调函数。
    div_1ms_1s++; //降频变量累加。
    if(div_1ms_1s>=1000){ //加到1000的时候，就到了1秒。
        div_1ms_1s=0; //清零。
        one_sec_flag=1; //设置1秒标志。
    }
}
}

```

## 注意事项

1. 本函数用于设置时间，对于本机跑时间来说即时生效，对于外置RTC硬件来说，需要将那几个时间变量存到外置RTC硬件里才生效。具体的保存方法会在对应的RTC库里提到，比如DS1302库。
2. 本库不提供外置RTC库的联动，如使用了外置RTC芯片，请自行完善保存读取操作。
3. 使用本函数之前先确认函数使能打开，如图所示：

Option	Value
SOFT-RTC运行函数使能	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> SOFT-RTC时间更新函数使能	<input checked="" type="checkbox"/>
SOFT-RTC时间设置函数使能	<input checked="" type="checkbox"/>

## soft\_rtc\_update

函数原型：u8 soft\_rtc\_update(u8 dat);

### 描述

设置时间函数。

### 输入

- dat：串口接收到的数据。

### 输出

无

## 返回值

- 0: 数据不完整或正在接收中。
- 1: 数据完整。

## 调用例程

```
#include "ecbm_core.h" //加载库函数的头文件。
#include "soft_rtc.h" //加载软件RTC库函数的头文件。
bit one_sec_flag=0; //1秒标志。
bit time_save_flag=0; //时间保存标志位。
void main(void){ //main函数，必须的。
    system_init(); //系统初始化函数，也是必须的。
    uart_init(); //串口初始化。
    timer_init(); //定时器初始化。
    timer_set_timer_mode(0,1000); //设置定时器0为1ms。
    timer_start(0); //启动定时器0。
    while(1){
        if(one_sec_flag){ //1秒标志为1时，执行。
            one_sec_flag=0; //清除标志。
            soft_rtc_run(); //运行软件RTC。
            uart_printf(1,"%04u-%02bu-%02bu %02bu:%02bu:%02bu
week:%bu\r\n",soft_rtc_year,soft_rtc_month,soft_rtc_day,soft_rtc_hour,soft_rtc_m
inute,soft_rtc_second,soft_rtc_week);
        }
        if(time_save_flag){
            time_save_flag=0;
            ...//做点保存措施。
            uart_printf(1,"保存成功! ");
        }
    }
}
u16 div_1ms_1s; //降频变量，用于把1ms时间降为1s。
void timer0_it_callback(void){ //定时器0中断回调函数。
    div_1ms_1s++; //降频变量累加。
    if(div_1ms_1s>=1000){ //加到1000的时候，就到了1秒。
        div_1ms_1s=0; //清零。
        one_sec_flag=1; //设置1秒标志。
    }
}
void uart1_receive_callback(void){ //串口1接收中断回调函数。
    if(soft_rtc_update(SBUF)){ //解析时间戳，更新软件RTC。同时判断是否接受完整。
        time_save_flag=1; //接收完整的话，就通知主程序可以保存了。
    }
}
```

## 注意事项

1. 本函数用于上位机通过串口来设置时间，对于本机跑时间来说即时生效，对于外置RTC硬件来说，需要将那几个时间变量存到外置RTC硬件里才生效。具体的保存方法会在对应的RTC库里提到，比如DS1302库。
2. 本库不提供外置RTC库的联动，如使用了外置RTC芯片，请自行完善保存读取操作。

3. 使用本函数之前先确认函数使能打开，如图所示：

Option	Value
SOFT-RTC运行函数使能	<input checked="" type="checkbox"/>
SOFT-RTC时间更新函数使能	<input checked="" type="checkbox"/>
命令头	SOFT-RTC
SOFT-RTC时间设置函数使能	<input checked="" type="checkbox"/>

4. 通信头可以自由定义，但是不能留空！同时在上位机也要设置一模一样的通信头，否则就会更新失败！

起始地址

0x0000

☒清除代码缓冲区

打开程序文件

0x0000

☒清除EEPROM缓冲区

打开EEPROM文件

RS485控制

自动增量

用户接口

RTC校对

波特率 115200

校验位 无校验

停止位 1位

命令头 SOFT-RTC

☐Hex

☒获取系统时间

2024-09-14

11:01:05

时间误差补偿 0

ms

设置RTC

5. 校时上位机就在STC-ISP工具中，根据截图可以找到其位置，设置好之后点击【设置RTC】就能把当前电脑的时间发送给单片机进行校时。因此电脑的时间要是不准的话，校时的结果也必然是不准的。截图即是demo代码出来的效果，本来上电没有初始化，日期和时间都是从0开始，校时之后变

接收缓冲区

☒文本模式

☐HEX模式

清空接收区

保存接收数据

复制接收数据

成了当前时间：

0000-00-00 00:02:29 week:0

0000-00-00 00:02:30 week:0

保存成功! 2024-09-14 11:05:05 week:6

2024-09-14 11:05:06 week:6

2024-09-14 11:05:07 week:6

## 优化建议

根据实际需求控制函数的使能就行了。