



# YC31xx RTC 应用说明

V1.0

Yichip Microelectronics

©2014

**Revision History**

Version	Date	Author	Description
V1.0	2020-2-19	Duanziyang	Initial version

**Confidentiality Level:****confidential**

## 目录

1	文档说明 .....	4
1.1	编写目的 .....	4
1.2	适用范围 .....	4
1.3	文件说明 .....	4
2	库函数说明 .....	5
2.1	RTC_Config.....	5
2.2	RTC_Set_SecMax.....	5
2.3	RTC_Get_SexMax.....	5
2.4	RTC_SetRefRegister.....	6
2.5	RTC_GetRefRegister.....	6
2.6	RTC_SetWakeUpCounter.....	6
2.7	RTC_ClearITPendingBit .....	6
2.8	RTC_GetITStatus .....	7
3	Demo 函数说明 .....	8
3.1.1	RTC_Configuration .....	8
3.1.2	SEC_IRQHandler .....	8
rtccnt	.....	9

## 1 文档说明

### 1.1 编写目的

为使用 demo 中 RTC 相关示例代码及 API 提供指南

### 1.2 适用范围

3121 系列芯片

### 1.3 文件说明

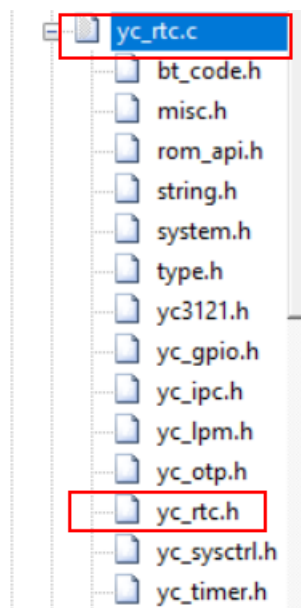
该 demo 实现的功能为使用外部 24M 晶振时钟（我们认为其是精准的）来校准 32K RTC。

RTC Demo 路径为

ModuleDemo\BPU

RTC 库文件为如下图 rtc.c 与 rtc.h,路径为

Librarier\sdk



## 2 库函数说明

### 2.1 RTC\_Config

函数原型: void RTC\_Config(FunctionalState NewState);

说明: 使能或者失能 RTC

参数	方向	说明
FunctionalState NewState	IN	ENABLE or DISABLE RTC

表格 2-1-1 RTC\_Config 形参表

返回值	说明
None	None

表格 2-1-2 RTC\_Config 返回值

### 2.2 RTC\_Set\_SecMax

函数原型: void RTC\_Set\_SecMax(uint16\_t secmax);

说明: 设置 1s 钟的 clock 数量

参数	方向	说明
uint16_t secmax	IN	32k 时钟以 secmax 为校准值, 例: secmax 为 0x8000, 则 32k 时钟每 0x8000 个周期为 1s

表格 2-2-1 RTC\_Set\_SecMax 形参表

返回值	说明
None	None

表格 2-2-2 RTC\_Set\_SecMax 返回值

### 2.3 RTC\_Get\_SexMax

函数原型: uint32\_t RTC\_Get\_SexMax(void);

说明: 通过 24M 晶体获得 RTC SecMax

参数	方向	说明
None		None

表格 2-3-1 RTC\_Get\_SexMax 形参表

返回值	说明
uint16_t secmax	参考 <a href="#">RTC_Set_SecMax</a>

表格 2-3-2 RTC\_Get\_SexMax 返回值

## 2.4 RTC\_SetRefRegister

函数原型: void RTC\_SetRefRegister(uint32\_t RefRegValue);

说明: 设置 [rtccnt](#)

参数	方向	说明
uint32_t RefRegValue	IN	<a href="#">rtccnt</a>

表格 2-4-1 RTC\_SetRefRegister 形参表

返回值	说明
None	None

表格 2-4-2 RTC\_SetRefRegister 返回值

## 2.5 RTC\_GetRefRegister

函数原型: uint32\_t RTC\_GetRefRegister(void);

说明: 获取 [rtccnt](#)

参数	方向	说明
None		None

表格 2-5-1 RTC\_GetRefRegister 形参表

返回值	说明
rtccnt	<a href="#">rtccnt</a>

表格 2-5-2 RTC\_GetRefRegister 返回值

## 2.6 RTC\_SetWakeUpCounter

函数原型: void RTC\_SetWakeUpCounter(uint32\_t wake\_RefRegValue);

说明: 设置 RTC wakeup 时间 当 RefRegValue==wake\_RefRegValue 触发中断

参数	方向	说明
uint32_t wake_RefRegValue	IN	Timeout 事件触发的阈值 (s)

表格 2-6-1 RTC\_SetWakeUpCounter 形参表

返回值	说明
None	None

表格 2-6-2 RTC\_SetWakeUpCounter 返回值

## 2.7 RTC\_ClearITPendingBit

函数原型: void RTC\_ClearITPendingBit(void);

说明: 清除 RTC 中断

参数	方向	说明
None		None

表格 2-7-1 RTC\_ClearITPendingBit 形参表

返回值	说明
None	None

表格 2-7-2 RTC\_ClearITPendingBit 返回值

## 2.8 RTC\_GetITStatus

函数原型：ITStatus RTC\_GetITStatus(void);

说明： 获取 RTC 中断状态

参数	方向	说明
None		None

表格 2-8-1 RTC\_GetITStatus 形参表

返回值	说明
RTC interrupt status	RESET/SET

表格 2-8-2 RTC\_GetITStatus 返回值

### 3 Demo 函数说明

#### 3.1.1 RTC\_Configuration

```
void RTC_Configuration(void)
```

```
{  
    uint32_t i = 0xffff;  
    uint16_t rtcCheckValue = 0;           延时，等待晶振稳定，必须有  
    while (i--);  
  
    rtcCheckValue = RTC_Get_SexMax();      获取 secmax  
    MyPrintf("32k RtcCVal: 0x%x\n", rtcCheckValue);  
  
    RTC_Set_SexMax(rtcCheckValue);  
    RTC_SetRefRegister(0);  
    RTC_SetWakeUpCounter(ONE_MINUTE);     设置校准时间，以秒为单位  
  
    RTC_Config(ENABLE);  
    RTC_ClearITPendingBit();  
}
```

#### 3.1.2 SEC\_IRQHandler

中断触发时进入该函数，进入该函数设置对下一个时间段的校准值（sexmax）

```
void SEC_IRQHandler(void)
```

```
{  
    uint16_t rtcCheckValue = 0;  
  
    rtcCheckValue = RTC_Get_SexMax();  
    MyPrintf("RTC_IRQHandler In 32k RtcCVal: 0x%x\n", rtcCheckValue);  
    MyPrintf("GET RTC RefRegister value: %d\n", RTC_GetRefRegister());  
  
    RTC_Set_SexMax(rtcCheckValue);  
    RTC_ClearITPendingBit();  
}
```



## rtcnt

rtc 计数器强制写入，clk\_lfo 以 (secmax+1) 分频，rtcnt 对分频时钟循环计数，可理解为 RTC 每 1s，rtcnt 加 1

YiCHiP