Shanghai Fudan Microelectronics Group Company Limited

复旦微电子

FM33G0XX
低功耗系列 MCU
应用笔记

RTC 应用补偿

AN001 **V1.0**

联系方式：

电表产品应用：

邢杰： xingjie@fmsh.com.cn TEL：13916427310

陈钊： chenzhao@fmsh.com.cn TEL：18616125501

水气热表及智能家居：

朱发旺：zhufawang@fmsh.com.cn TEL：17749796664

姜涛： jiangtao@fmsh.com.cn TEL：18701992908

超高频 900M 及物联网相关：

王晓腾：wangxiaoteng@fmsh.com.cn TEL：13585663727

王天纵：wangtianzong@fmsh.com.cn TEL：18221803903

资料下载及交流：

开发者论坛：http://www.fmdevelopers.com.cn

目　　录

表目录

图目录

1 说明

本文档为 FM33G0XX 系列低功耗 MCU 的应用笔记，用于说明实时时钟（RTC）原理和基本应用方法。FM33G0XX 系列是复旦微电子公司开发的低功耗 MCU 芯片，请联系复旦微电子公司提供更多相关文档支持设计开发。

2 主要特点

实时时钟(RTC)模块可长时间维持精确计时，为系统提供实时时钟和日历。该模块功耗极低，最大程度延长电池寿命。

RTC 的主要特点：

- BCD 时间格式，完整万年历
- 支持数字调校，最小步长 0.119ppm，精度可达+/-0.06ppm
- 可输出周期唤醒中断
- 闹钟功能
- RTC 计时部分寄存器不受系统复位影响
- 支持软件温度补偿/自动温度补偿

3 RTC 原理和基本应用

RTC 上电后不复位，因此正常工作前需要软件置入当前时间。走时时钟使用 32.768KHz 晶体振荡器。由于晶体振荡器有可能停振，为了保证可靠性，停振检测电路使能后不断检测 32.768KHz 振荡器输出，一旦发现停振，则产生报警中断，同时将 RTC 时钟切换到 RCLP，此时 RTC 走时有一定误差，但是并不会停止。

FM33G0xx 支持 RTC 自动温度补偿和软件温度补偿

□自动温度补偿原理是定时（256s）启动一次温度传感器，根据温度传感器的采集输出 RTC 调校值，实现每 256s 更新一次 RTC 调校值。

□软件温度补偿原理是根据实际应用中的需求（示例中设定为 10s），根据经验曲线进行误差补偿。

3.1 BCD 时间设定与读取

参考例程如图 3-1 所示

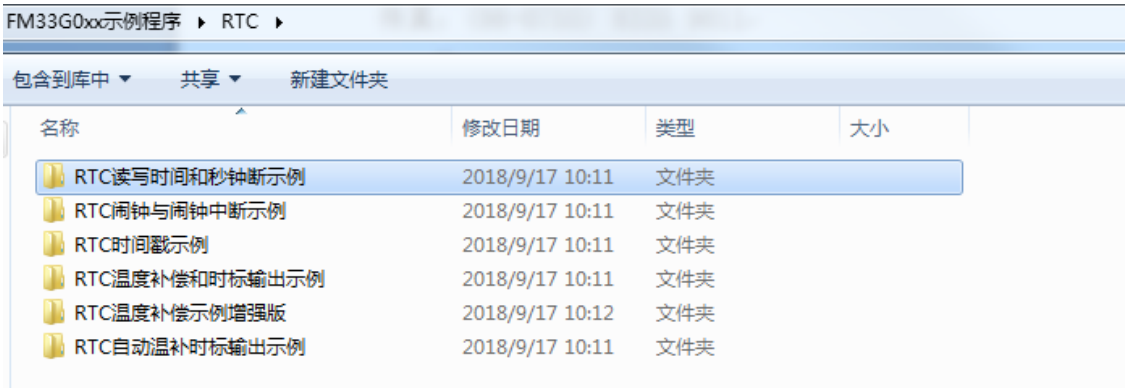

图 3-1 RTC 读写时间和秒中断示例

3.1.1 RTC 时间设定

软件可以在任意时刻直接设置 RTC 时间寄存器，通常建议在 XTLF 完成起振后再设置时间；由于设置时间寄存器的操作与 RTC 走时为异步操作关系，为了防止 RTC 时间设定过程中产生进位，例如当设置到分钟寄存器，产生秒进位信号使分钟计数器加 1，而导致设定错误，所以在设置时间后读出时间值进行校验。

FM33G0XX- RTC 读写时间和秒中断示例中有封装好的 RTC 设置时间函数，用户可直接调用。如图 3-2 所示。

```
int main (void)
{
  RTC_TimeDateTypeDef TempTime;
  const uint08 InitTime[] = {0x18, 0x01, 0x01, 0x01, 0x02, 0x03, 0x01};

  Init_System();       //系统初始化

  RTC_Init();        //RTC初始化
  memcpy((uint08*)(&TempTime), InitTime, 7);
  RTC_SetRTC(&TempTime);//设置时间

  for( ; ; )
    {
    IWDT_Clr();              //清系统看门狗

    RTC_GetRTC(&TempTime);  //读取时间 --------仿真模式下查看写入时间

    }
}
```

图 3-2 RTC 时间设定程序

RTC 设定时间函数流程

- CPU 向 RTCWE 写入 0xACACACAC 时，解除 RTC 写保护，此时 RTCWE 置 1
- 向 RTC 的 BCD 时间寄存器写入初值

- CPU 向 RTCWE 写入不为 0xACACACAC 的任意值(示例写入 0x53535353)时恢复写保护，这时 RTCWE 清 0。

具体设定时间函数如图 3-3 所示。

```
uint08 RTC_SetRTC(RTC_TimeDateTypeDef* para)
{
  uint08 n, i;
  uint08 Result;
  RTC_TimeDateTypeDef TempTime1;

  for(n=0 ;n<3; n++)
  {
    RTC_RTCWE_Write(RTC_WRITE_ENABLE);  //解除RTC写保护
    RTC_TimeDate_SetEx(para);     //设置RTC
    RTC_RTCWE_Write(RTC_WRITE_DISABLE); //打开RTC写保护

    Result = RTC_GetRTC(&TempTime1);  //读取确认设置结果
    if(Result == 0)
    {
      Result = 1;
      for(i=0; i<7; i++)//两者一致，表示设置成功
      {
        if(((uint08*)(&TempTime1))[i] != ((uint08*)(para))[i]) break;
      }
      if(i == 7)
      {
        Result = 0;
        break;
      }
    }
  }
  return Result;
}
```

图 3-3 RTC 设定时间函数

注意:硬件并不检查 BCD 时间各个寄存器值合法性，例如月计时正常计数范围为 1-12，但是 BCD 时间月寄存器可以设置为 13，软件须保证写入的 BCD 时间正确。

3.1.2 RTC 时间读取

FM33G0XX- RTC 读写时间和秒中断示例中有已经封装好的 RTC 时间读取函数，用户可直接调用即可，如图 3-4 所示。

```
int main (void)
{
  RTC_TimeDateTypeDef TempTime;
  const uint08 InitTime[] = {0x18, 0x01, 0x01, 0x01, 0x02, 0x03, 0x01};

  Init_System();      //系统初始化

  RTC_Init();       //RTC初始化
  memcpy((uint08*)(&TempTime), InitTime, 7);
  RTC_SetRTC(&TempTime);//设置时间

  for( ; ; )
    {
    IWDT_Clr();             //清系统看门狗

    RTC_GetRTC(&TempTime);  //读取时间 --------仿真模式下查看写入时间

    }
}
```

图 3-4 RTC 读取设定时间程序实例

RTC 时间读取流程，如图 3-5、3-6 所示：

- 读当前时间寄存器值
- 再次读当前时间寄存器值
- 如果 2 次读取内容一致，则为正确的当前时间；如果两次读取内容不一致，则重复前两个步骤(共三次)。

  软件在 1s 中断发生后立即读取时间寄存器，能保证读到正确的当前时间值。

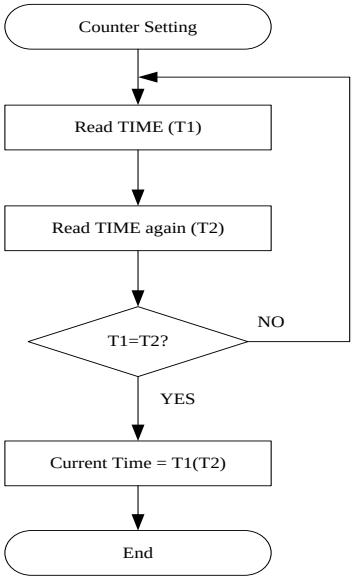

图 3-5 RTC 时间读取流程

```
uint08 RTC_GetRTC(RTC_TimeDateTypeDef* para)
{
  uint08 n, i;
  uint08 Result = 1;

  RTC_TimeDateTypeDef TempTime1,TempTime2;

  for(n=0 ;n<3; n++)
  {
    RTC_TimeDate_GetEx(&TempTime1);//读一次时间
    RTC_TimeDate_GetEx(&TempTime2);//再读一次时间

    for(i=0; i<7; i++)//两者一致，表示读取成功
    {
      if(((uint08*)(&TempTime1))[i] != ((uint08*)(&TempTime2))[i]) break;
    }
    if(i == 7)
    {
      Result = 0;
      memcpy((uint08*)(para), (uint08*)(&TempTime1), 7);//读取正确则更新新的时间
      break;
    }
  }
  return Result;
}
```

图 3-6 RTC 读取时间函数

3.2 RTC 时间戳

RTC 时间戳参考例程如图 3-7 所示

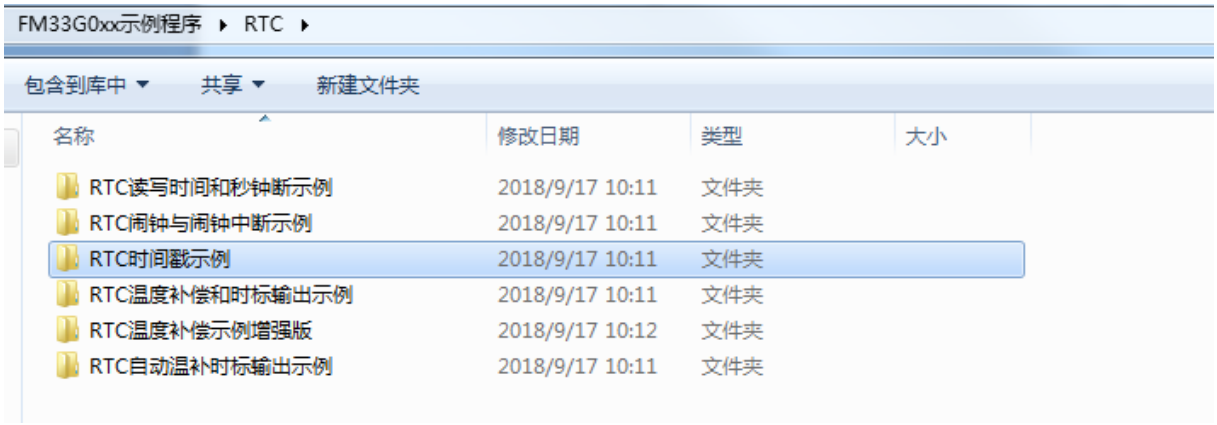

图 3-7 RTC 时间戳示例

RTC 支持外部 IO 事件触发的时间戳功能,如图 3-8 所示。外部 IO 触发源为 PB4 和 PB5 的输入电平变化，为了确保输入检测的可靠性，建议使能 PB4 和 PB5 的 IO 输入数字滤波。使用此功能时，将 PB4 和 PB5 配置为 GPIO 输入(实例中选用 PB5)，打开 RTCSTAMPEN 寄存器，当 PB4 和 PB5 上出现任何滤波后的上升沿或下降沿时，RTC 会自动记录当前时间到 STAMP 寄存器组中，同时产生相应的标志，可用于产生中断或者供软件查询。

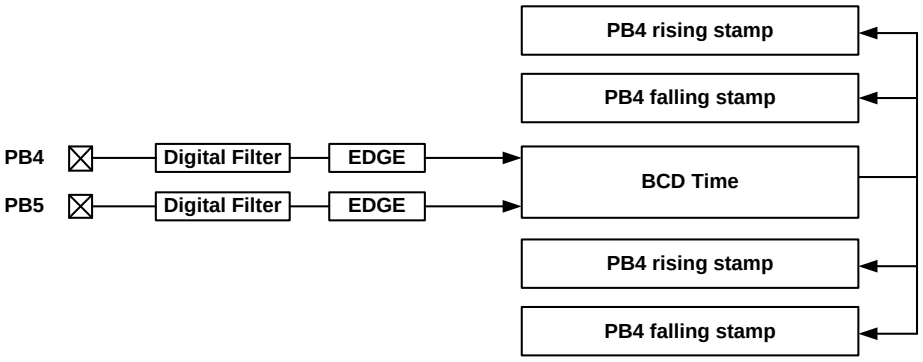

图 3-8 外部 IO 触发源

FM33G0XX- RTC 时间戳示例中有已经封装好的 RTC 时间戳测试函数，用户根据自己实际情况修改即可，如图 3-9、3-10 所示。

```
int main (void)
{
  RTC_TimeDateTypeDef TempTime;
  const uint08 InitTime[] = {0x18, 0x01, 0x01, 0x01, 0x02, 0x50, 0x01};

  Init_System();      //系统初始化
  IWDT_IWDTCFG_IWDTSLP4096S_Setable(ENABLE);

  RTC_Init();       //RTC初始化
  memcpy((uint08*)(&TempTime), InitTime, 7);
  RTC_SetRTC(&TempTime);  //设置时间

  Test_RTC_STAMP();   //注意时间戳功能仅在休眠模式下有效

  for( ; ; )
    {
    IWDT_Clr();             //清系统看门狗

    RTC_GetRTC(&TempTime);  //读取时间

    LED0_TOG;             //LED0闪烁
    TicksDelayMs( 50, NULL );//软件延时
    }
}
```

图 3-9 时间戳示例

```
void Test_RTC_STAMP(void)//注意时间戳功能仅在休眠模式下有效
{
  RTC_TimeDateTypeDef TempTime1, TempTime2;

  memset((uint08*)&TempTime1, 0x00, 7);

  InputtIO( GPIOB, GPIO_Pin_5, IN_NORMAL );    //PB5;配置为输入IO
  RTC_STAMPEN_STAMP1EN_Setable(ENABLE); //PB5触发的时间戳功能使能

  RTC_RTCIE_SetableEx(ENABLE, RTC_RTCIE_MIN_IE_Msk);//打开RTC分钟中断 作为定时唤醒源
  RTC_RTCIF_ClrEx(RTC_RTCIE_MIN_IE_Msk);  //清除中断标志
  NVIC_DisableIRQ(RTC_IRQn);          //NVIC中断控制器配置
  NVIC_SetPriority(RTC_IRQn,2);
  NVIC_EnableIRQ(RTC_IRQn);

  //进入休眠 等待分钟中断唤醒
  Test_Sleep();

  RTC_CLKSTAMPxx_GetEx(STAMP1FALL, &TempTime1);                    //记录时间发生的时间，通过仿真状态查看
  RTC_CLKSTAMPxx_GetEx(STAMP1RISE, &TempTime2);

  TicksDelayMs( 1000, NULL );//软件延时
}
```

图 3-10 时间戳函数

注意时间戳功能仅在休眠模式下有效，ACTIVE 和 LPRUN 模式下时间戳功能不起作用。时间戳仅在相应标志寄存器为 0 的情况下记录事件发生时间，如果对应标志已经为 1，则忽略相应事件。因此如果有多次事件发生，时间戳仅记录第一次事件发生的时间，除非软件在事件发生后清除了标志寄存器。

3.3 闹钟功能

RTC 时间戳参考例程如图 3-11 所示。

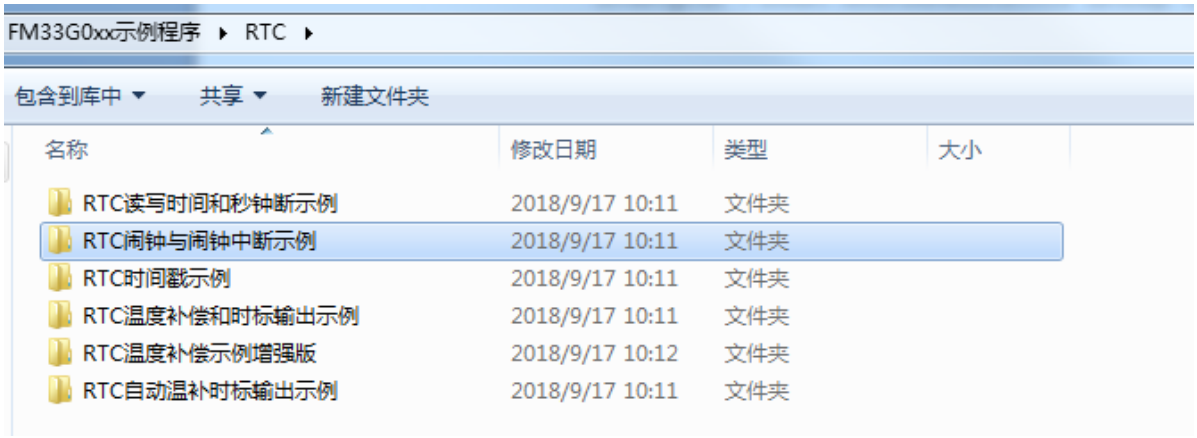

图 3-11 RTC 闹钟与闹钟中断示例

FM33G0XX 闹钟寄存器 ALARM,可以设置秒、分和时三个段，并在 RTC 中断使能寄存器中开启闹钟中断 ALARM_IE 即可（RTCIF---bit11），示例程序中有闹钟示例，用户根据自己实际情况修改即可，如图 3-12、3-13 所示。

```
int main (void)
{
  RTC_TimeDateTypeDef TempTime;
  const uint08 InitTime[] = {0x18, 0x01, 0x01, 0x01, 0x02, 0x03, 0x01};

  Init_System();      //系统初始化

  RTC_Init();       //RTC初始化
  memcpy((uint08*)(&TempTime), InitTime, 7);
  RTC_SetRTC(&TempTime);//设置时间

  Test_RTC_Alarm();   //闹钟测试

  for( ; ; )
    {
    IWDT_Clr();             //清系统看门狗

    RTC_GetRTC(&TempTime);  //读取时间

    }
}
```

图 3-12 RTC-ALARM 程序示例

```
void Test_RTC_Alarm(void)
{
  RTC_AlarmTmieTypeDef TempAlarmTime;

  TempAlarmTime.Hour = 0x01;
  TempAlarmTime.Minute = 0x02;
  TempAlarmTime.Second = 0x013;

  RTC_AlarmTime_SetEx(&TempAlarmTime);  //设置闹钟时间

  RTC_RTCIE_SetableEx(ENABLE, RTC_RTCIE_ALARM_IE_Msk);//打开RTC闹钟中断
  RTC_RTCIF_ClrEx(RTC_RTCIE_ALARM_IE_Msk);  //清除闹钟中断标志
  NVIC_DisableIRQ(RTC_IRQn);          //NVIC中断控制器配置
  NVIC_SetPriority(RTC_IRQn,2);
  NVIC_EnableIRQ(RTC_IRQn);
}
```

图 3-13 RTC-ALARM 测试函数

实验结果：当达到定时时间(10s)后，LED1 点亮。

3.3 LTBC 调校

数字调校的目的是使 RTC 能够在较长周期内获得平均准确的计时。由于 RTC 的时钟源是 32768Hz，因此数字调校的最小步长是 30.5us，如果在 1 秒内调整一次，则最高精度只能达到 30.517ppm。为了得到更高精度，必须在更长时间周期内进行调整。FM33G0XX 以 256s 为一个调校周期，由于 ADJUST[11:0],每个周期内可以调整 0~+/-4095 个 32768Hz 时钟周期，因此最小步长为 30.5us/256s=0.119ppm，最大调校范围为 +/-(4095*30.517us/256s)=+/-488.1ppm，调校后平均最小时钟误差为+/-0.06ppm。

数字调校主要由 LTBC 数值调整寄存器、LTBC 数值调整方向寄存器、LTBC 虚拟调校使能寄存器和 RTC 调校步长选择寄存器组成。

RTC 最小调校步长可以选择 0.119 或者 0.238PPM。当调教值为 256 时理论变化如表 3-1 所示

表 3-1 调教值为 256 时理论变化

调校精度 PPM	调教值	理论变化 PPM
0.119	256	30.464
0.238	256	60.928

步长	ADSIGN	ADJUST	CALSTEP	FOUT（Hz）
0.238 ppm	0x01	0XFF	0X00	1.000037
	0x01	0X00	0X00	0.999976
0.119ppm	0x01	0XFF	0X01	1.000006
	0x01	0X00	0X01	0.999976

	0x00	0XFF	0X01	0. 999946

例如，步长 0.238，ADJUST=0X00,说明未进行补偿，即 FOUT=0.999976 为原始误差。ADJUST=0XFF,说明步长 0.238*256=60.928，即 FOUT 补偿为 0.999976+0.000060928 约为 1.000037

4 RTC 软件温度补偿

FM33G0XX 系列 MCU 内置支持万年历功能的低功耗实时时钟，并且拥有走时误差数字调校功能，通过软硬件配合可实现高精度计时。

4.1 补偿原理

RTC 电路采用外接 32768Hz 石英晶振作为计数源，石英晶体的频率误差可分为工艺偏差和温度偏差。工艺偏差来自晶体的制造过程，一般在±20ppm 左右，这部分误差可以看作固定偏差，我们称之为顶点误差，温度误差则是石英晶体自身的物理特性。

石英晶振的典型温度特性如下图 4-1 所示，常温附近频率最高，我们称之为顶点，一般出现在 25℃附近。

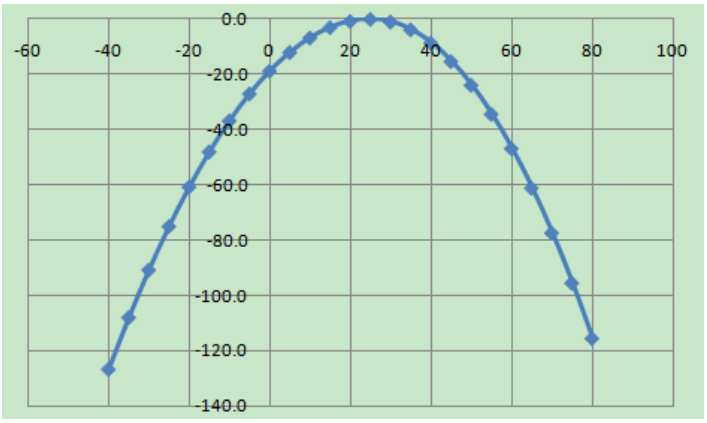

图 4-1 晶体温度(° C)-偏差(ppm)曲线

随着温度偏离顶点，无论温度上升或下降，晶振的频率都会变慢，偏差变大，其温度特性曲线近似于抛物线。品质优良的石英晶体的温度偏差一致性好，生产应用时可使用统一的曲线来计算温度偏差。

常用 RTC 补偿方法一般为数字调校和模拟电容调校或者两者结合使用。以 G 系列为例，其 RTC 电路支持数据调校，调整补偿±0.119/0.238ppm，，调整范围±488ppm，能够在全温区实现高精度数字补偿。

4.2 补偿方法

石英晶体的温度特性曲线近似于抛物线，但并不完全符合抛物线，为了使误差计算结果更接近实际情况，我们采用分段的方法以顶点温度为分界点，用两条抛物线拼接的方法来模拟石英晶体的温度曲线。两条抛物线分别使用两个独立的系数 KL 和 KH，与使用一条抛物线作为温度特性曲线相比较，按顶点分两部分的计算方法可以提高补偿精度，如图 4-2 所示。

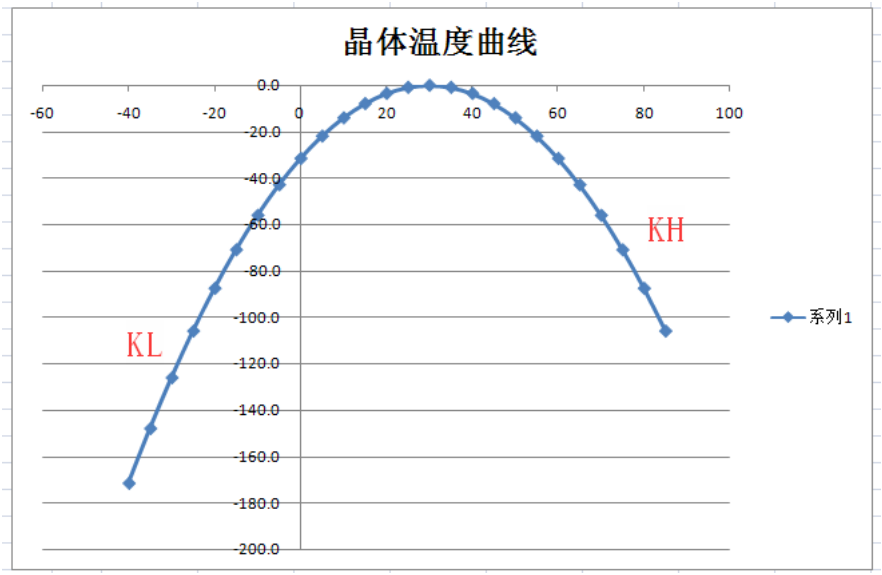

图 4-2 KL 和 KH 温度(°C)-偏差(ppm)曲线

例如，顶点温度为 25℃，KL= -0.0300，KH= -0.0382，可以计算出温度误差如下表 4-1 所示。

表 4-1 不同温度下偏差数据

温度（°C）	误差(ppm)
-40	-126.8
-20	-60.8
0	-18.8
25	0.0
50	-23.9
70	-77.4
85	-137.5

25℃时，晶体处于顶点，计算得温度偏差为 0，由于石英晶体固定偏差的存在，此时不能直接按 0 来进行补偿。例如当顶点误差等于 6ppm 时，RTC 的实际误差数据如下表所示。6ppm 的顶点误差需要叠加到所有温度偏差上，所以我们实际需要补偿的曲线是下图 4-3 所示中的红色曲线，在 25℃时需要补偿 6ppm 的频率偏差，如下表 4-2 所示。再将 6ppm 的误差转换成补偿寄存器的值并写入寄存器即可完成 25℃温度点的 RTC 顶点温度补偿。

表 4-2 存在顶点误差时，不同温度的偏差数据

温度	误差 ppm	误差 ppm（含 6ppm 顶点误差）
-40	-126.8	-120.8
-20	-60.8	-54.8
0	-18.8	-12.8
25	0.0	6.0
50	-23.9	-17.9
70	-77.4	-71.4
85	-137.5	-131.5

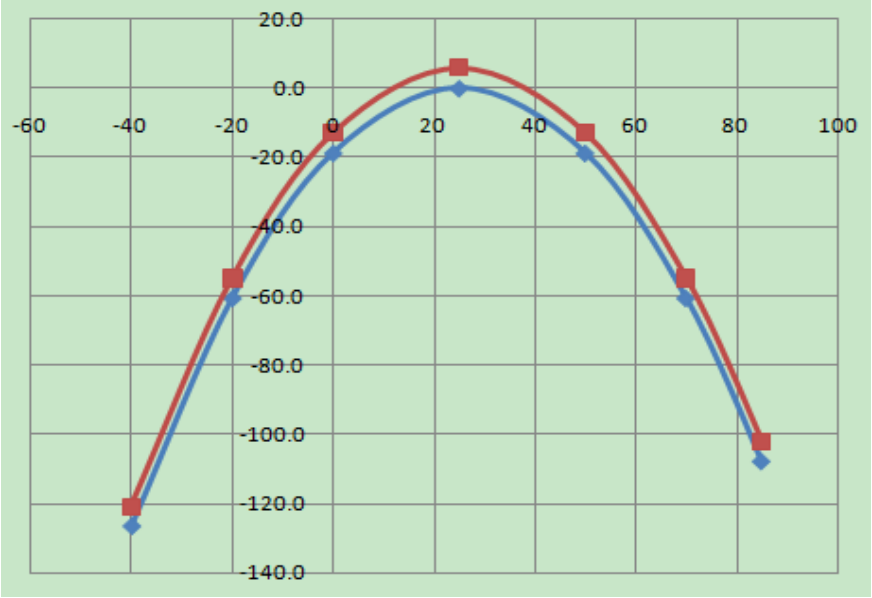

图 4-3 存在顶点误差温度(° C)-偏差(ppm)曲线

◎补偿计算方法：

□温度低于顶点温度

误差 = KL * (当前温度 – 顶点温度)^2 + 顶点误差

□温度高于顶点温度

误差 = KH * (当前温度 – 顶点温度)^2 + 顶点误差

□补偿值计算

补偿值 = 误差/ 0.119（0.238）

RTC->ADJUST = 补偿值（绝对值）

RTC->ADSIGN = 0/1（补偿值大于等于 0 时写 0，补偿值小于 0 时写 1）

◎补偿示例：

□25℃

误差 = KL * (25 - 25)^2 + 6 = 6

补偿值 = 6 / 0.119=50.4 = 50

写入寄存器 RTC->ADJUST = 50； RTC->ADSIGN = 0

□-20℃

误差 = KL * (-20 - 25)^2 + 6 = -54.8

补偿值 = -54.8 / 0.119 = -460.5 = -460

写入寄存器 RTC->ADJUST = 460； RTC->ADSIGN = 1

□70℃

误差 = KH * (70 - 25)^2 + 6 = -71.4

补偿值 = -71.4 / 0.119= -600 = -600

写入寄存器 RTC->ADJUST = 600； RTC->ADSIGN = 1

由于晶体振荡器的特性还会受到电容以及老化的影响，此外不同的 PCB 布线或不同的 PCB 供应商都会带来寄生电容的改变，所以示例中的典型情况并不适用于所有用户。用户在做完样机后需要自行测试一组老化后的样机的温度频率数据，并以此修正 KH，KL 是的温度系数适应自家产品，获得最佳补偿效果。

4.3 RTC 温度补偿硬件设计

本章需要用到的硬件如图 4-4 所示

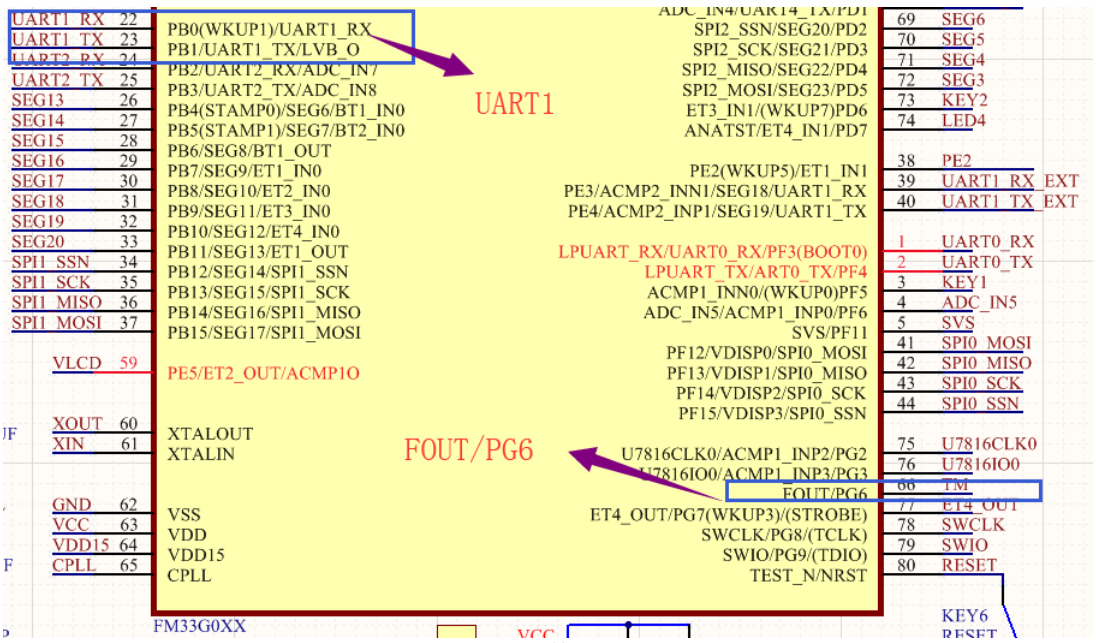

图 4-4 硬件原理图设计

1、UART1——下发外部测量的顶点误差；

2、FOUT/PG6——使用频率计测量配置的精确秒时标

对应的 PCB 位置如图 4-5 所示

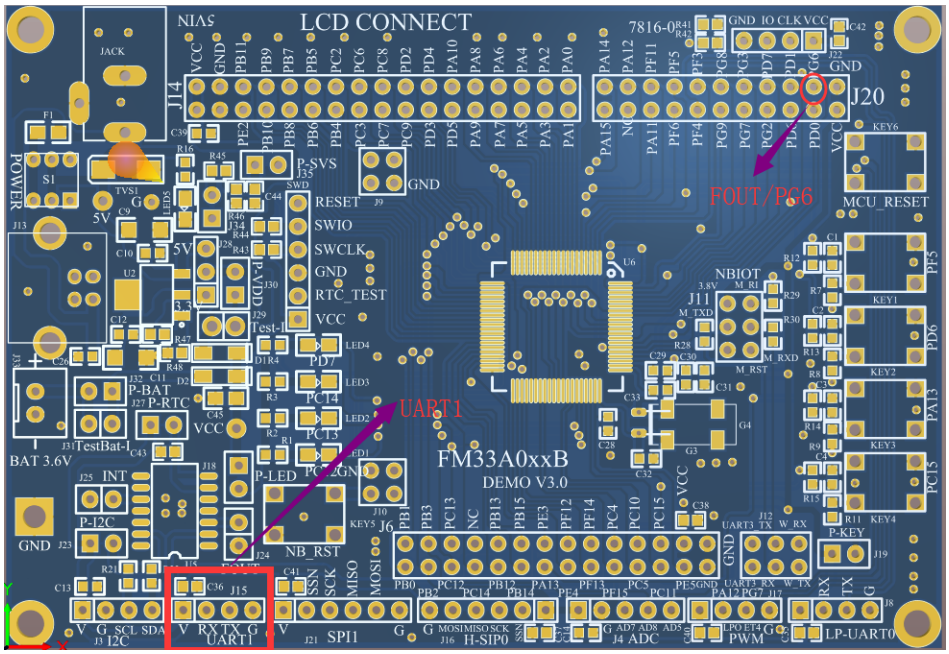

图 4-5 硬件 PCB 版图

4.4 RTC 温度补偿软件实现

4.4.1 参考例程

参考例程如图 4-6 所示

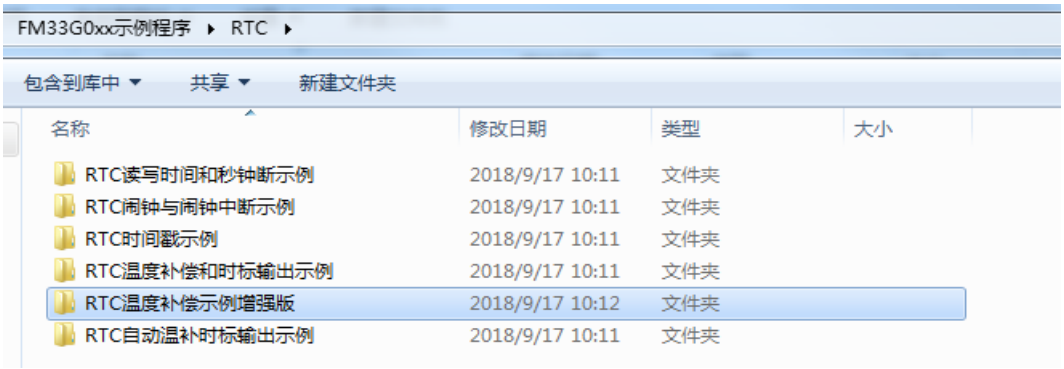

图 4-6 RTC 软件温度补偿示例

4.4.2 误差测量与计算

如图 4-7 所示为引起 RTC 误差的因素以及测量方法

□误差 = 顶点误差+温度误差

□温度误差可根据温度数据配合晶体曲线计算得出

□顶点误差可通过调校仪调校或外部手段得到，（建议使用外部手段测量，通过串口下发顶点误差）

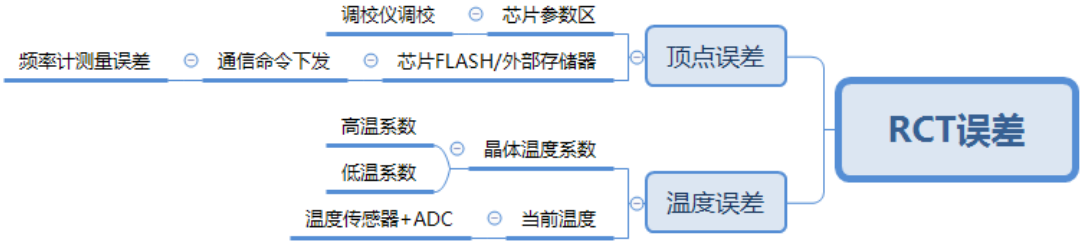

图 4-7 RTC 误差因素

□RTC 补偿流程

- 可定时根据温度对 RTC 进行误差修正，示例程序中每 10 秒进行一次误差补偿，实际应用中可根据需求调整。
- 休眠后 RTC 按最后写入的补偿值运行，所以需要定时唤醒进行 RTC 误差修正。
- 补偿流程如图 4-8 所示：

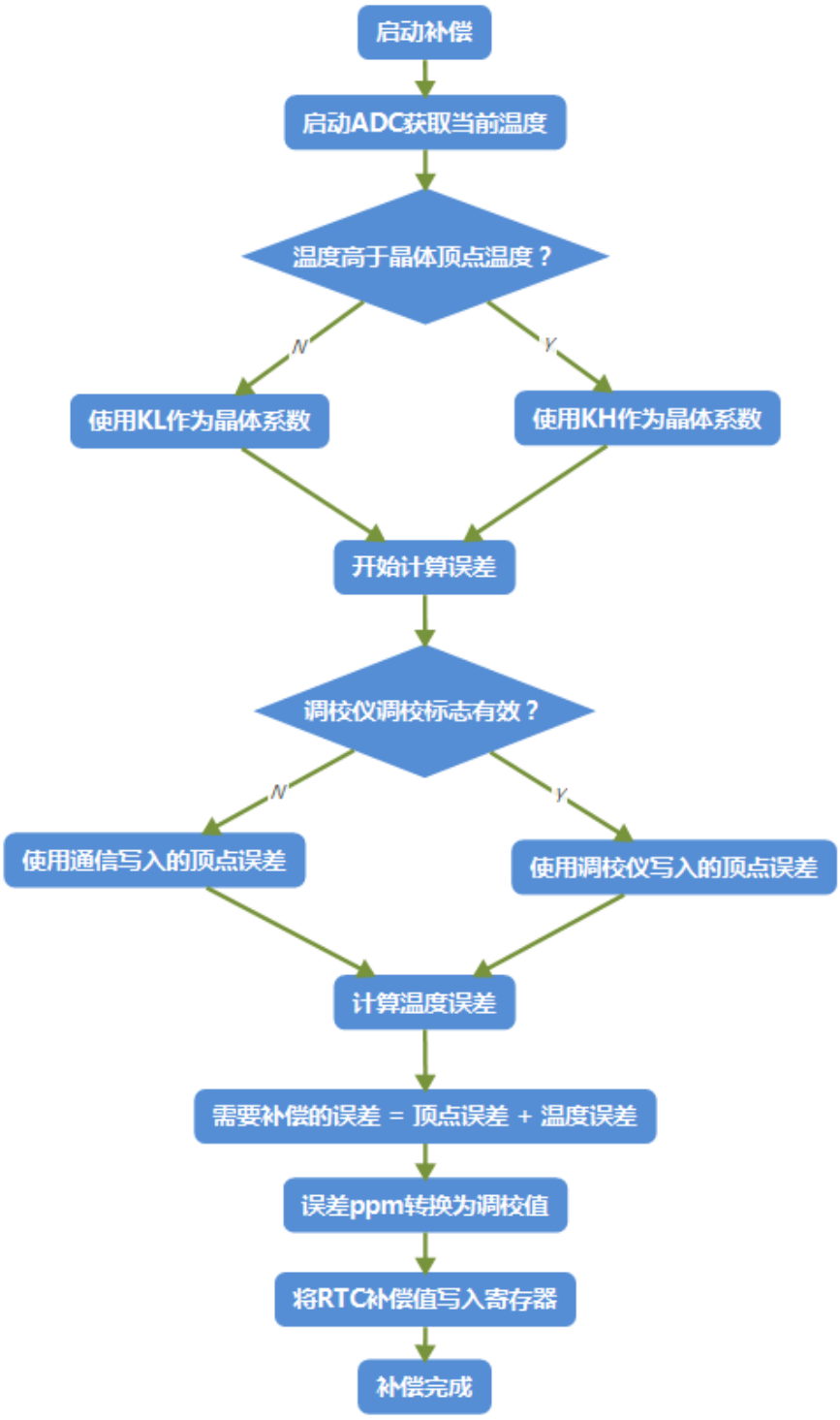

图 4-8 RTC 软件补偿流程

4.4.3 调校仪编程调校补偿

使用复旦微电子提供的专用脱机编程器对目标模块进行编程调校，即可完成对 RTC 顶点误差的测量与存储。

调校信息存储在芯片参数区，地址如下：

调校标志：0x1ffffa20（16 位无符号整数，0x3cc3 代表编程调校成功）

顶点误差：0x1ffffa36（16 位有符号整数，单位 0.01ppm）

使用时可通过指针访问，如图 4-9 所示：

```
main.c | fm33G0xx_btim.h | fm33g0xx_btim.c | FM33G0XX.h | fm33G0xx_rtc.h
64 #define const_tx_flag      (*((uint16_t *)(0x1ffffa20))) //调校标志(0x3cc3为编程调校)
65 #define const_rtc_adj_top   (*((int16_t *)(0x1ffffa36))) //顶点误差ppm,0.01ppm
66 #define const_rtc_backup  0x1FFFF800  //RTC备份寄存器首地址
67
68
```

图 4-9 调校信息存储 NVR 中

```
/*  误差计算
  输入：温度Temperature ， 晶体温度系数K(正数)
  输出：当前温度对应的误差ppm  */
float RTC_TrimData_Calculate(float Temperature, float K)
{
  float TopF_Rtc_Adj;   //顶点误差
  float Rtcadj_Offset;  //温度变换带来的误差变化

  if( const_tx_flag == 0x3cc3 )//调校仪校准时钟误差
  {
    TopF_Rtc_Adj = const_rtc_adj_top/100.0;//缩小100倍
  }
  else      //其他方法输入顶点误差ppm
  {
    TopF_Rtc_Adj = const_rtc_TopOffset/100.0;//缩小100倍
  }

  Rtcadj_Offset = K*(Temperature - const_xtl_top)*(Temperature - const_xtl_top);//顶点误差计算，ppm

  return (TopF_Rtc_Adj + Rtcadj_Offset);//需要补偿的误差ppm = 顶点误差 + 温度误差
}
```

当选择调校仪编程调校，0x3cc3 代表调校成功，使用 NVR 中的顶点误差调校值。

4.4.4 外部串口通信下发误差数据

对于需要先对芯片进行编程，再进行焊接的用户，则可以通过通信下发误差数据的方式对 RTC 进行调校（用户程序需添加相关支持代码）。

使用通信命令调校时，先下发命令关闭 RTC 温度补偿，然后通过频率计、日计时误差仪等设备测得 RTC 的走时误差后下发给芯片。测量误差时需要先配置芯片输出“精确秒时标”。如图 4-10 所示

```
//RTC输出精确1Hz配置函数
void RTC_CAL1HZ_Out(void)
{
  RCC_PLL_InitTypeDef PLL_InitStruct;

  //启动PLL，输出精确秒时标需要配置pll为16.384MHz
  PLL_InitStruct.PLLDB = 499; //pll倍频数 = PLLDB + 1
  PLL_InitStruct.PLLINSEL = RCC_PLLCON_PLLINSEL_XTLF; //PLL时钟源选择XTLF
  PLL_InitStruct.PLLOSEL = RCC_PLLCON_PLLOSEL_MUL1; //默认选择1倍输出，当超出PLLDB的1023时，可使用2倍输出
  PLL_InitStruct.PLLEN = DISABLE; //默认关闭PLL

  RCC_PLL_Init(&PLL_InitStruct);
  RCC_PLLCON_PLLEN_Setable(ENABLE);//启动PLL

  //配置IO口为FOUT并输出RTCTM
  GPIO_FOUTSEL_FOUTSEL_Set(GPIO_FOUTSEL_FOUTSEL_RTCTM);
  AltFunIO( GPIOG, GPIO_Pin_6, ALTFUN_NORMAL );//PG6;//TM输出

  //打开虚拟调校功能
  TicksDelayMs( 10, NULL );//软件延时//PR1SEN置位前需先使能PLL,并等待足够时间（>2ms）确保PLL锁定
  RTC_FSEL_FSEL_Set(RTC_FSEL_FSEL_PLL1HZ);//频率输出选择信号
  RTC_PR1SEN_PR1SEN_Setable(ENABLE);    //虚拟调校使能
}
```

图 4-10 精准秒时标

FM33G0XX 系列芯片的 RTC 支持从 FOUT（PG6）脚输出精确秒时标用于测量 RTC 走时精度，具体方法如下。

1. 配置 PLL 为 XTLF 信号的 500 倍并启动 PLL
2. 配置 PG6 为 FOUT 功能（IO 口数字特殊功能）并输出 RTCTM 信号
3. 待 PLL 锁定后（2ms）打开虚拟调校功能并配置 FSEL 输出 PLL1HZ

由于测量误差时的温度不一定是顶点温度，所以对于下发的误差，还要进行折算（Get_RTCTop_Proc 函数），如图 4-11 所示，将其换算为对应顶点温度的顶点误差。（注意：测量误差时需要保持环境温度稳定，并且低频晶体和 MCU 芯片温度平衡）

```
//RTC常温校准折算顶点误差函数
int16 Get_RTCTop_Proc( int16 Temp16 ) //顶点调校值折算
{
  int16 i;
  float Rtcadj, fTemperature, fK, Rtcadj_Offset2;

  fTemperature = Get_ADC_Temperature();//获取当前温度

  if(fTemperature > const_xtl_top) fK = const_KH;   //当前温度高于顶点温度
  else fK = const_KL;

  Rtcadj_Offset2 = fK*(fTemperature - const_xtl_top)*(fTemperature - const_xtl_top);//顶点误差计算，ppm

//  Rtcadj = Temp16 * 5.0 / 432.0 + Rtcadj_Offset2;   //输入日计时误差转换为ppm
  Rtcadj = Temp16 / 100.0 + Rtcadj_Offset2;   //输入0.01ppm，缩小为ppm

  Rtcadj = Rtcadj * 100.0;

  if( Rtcadj >= 0 )
  {
    i = Rtcadj + 0.5;
  }
  else
  {
    i = Rtcadj - 0.5;
  }
  return i;
}
```

图 4-11 转换顶点误差函数 Get_RTCTop_Proc

示例采用了一个简易通讯协议来传输误差数据。示例程序支持两条命令，一条为关闭补偿命令，一条为误差下发命令。通过串口下发三个字节，前两个字节为 RTC 误差数据（int16 型，低字节在前），单位 0.01ppm，第三个字节为前两个字节的校验和。收到三个字节后检查第三个字节是否为前两个字节的校验和，如果是，则将换算为顶点误差后的误差数据写入 FLASH 中，如图 4-12 所示。

```
/*
串口通信处理函数
通过串口下发三个字节，前两个字节为RTC误差数据（int16型，低字节在前），单位0.01ppm，第三个字节为前两个字节的校验和
收到三个字节后检查第三个字节是否为前两个字节的校验和，如果是，则将换算为顶点误差后的误差数据写入FLASH中
例如下发：BC 02 BE，代表RTC误差为7.00ppm
下发成功则返回00，失败返回FF
*/
void Uart1_RxProc(uint08 RxData)
{
  static uint08 RxState;
  static uint08 RxBuf[2];
  uint32 FlashOpID;
  int16 Temp16;
  uint08 Temp08 = 0xFF;

  switch(RxState)
  {
    case 0:
      RxBuf[0] = RxData;  //误差ppm数据低字节
      break;

    case 1:
      RxBuf[1] = RxData;  //误差ppm数据高字节
      break;

    case 2:
      if(RxData == (uint08)(RxBuf[0]+RxBuf[1])) //CS判断正确
      {
        //一般情况下顶点误差都在±30.00ppm以内，超过范围则认为异常，作为扩展命令使用
        Temp16 = (RxBuf[1]<<8)|RxBuf[0];
        if((Temp16<=3000)&&(Temp16>=-3000))
        {
          //将误差数据写入flash
          Temp16 = Get_RTCTop_Proc(Temp16);
          //存储在ee或者flash中，操作flash前读取赋值给FlashOpID
          FlashOpID = 0x12ABF00F;//FLASHOPKEY
          Flash_Erase_Sector(const_rtc_TopOffset_ADDR/512, FlashOpID);
          FlashOpID = 0;
```

命令校验

转换为顶点误差

图 4-12 串口下发顶点误差示例

正常情况下，RTC 初始误差都会在±30.00ppm 范围内，超过范围则认为异常，这部分命令作为扩展命令使用。

关闭补偿命令：30 75 A5，代表关闭 RTC 补偿（300.00ppm 命令）

误差下发命令：64 00 64，代表 RTC 误差为 1.00ppm

下发成功则返回 00，失败返回，如图 4-13 所示

图 4-13 串口助手下发数据指令

示例程序中 RTC 顶点误差存放在 flash 的 0x000001FE00 地址，实际应用中可根据需求调整存储位置，或者存储在外部 ee 或 flash 中，如图 4-14 所示。

```
main.c | fm33G0xx_btim.h | fm33g0xx_btim.c | FM33G0XX.h | fm33G0xx_rtc.h
1  #include "define_all.h"
2  #include <string.h>
3
4
5  //示例程序中RTC顶点误差存放在flash的0x000001FE00地址，实际应用中可根据需求调整位置，或者存储在外部ee或flash中
6  #define const_rtc_TopOffset_ADDR  0x000001FE00
7  #define const_rtc_TopOffset    (*((int16_t *)(const_rtc_TopOffset_ADDR)))  //rtc顶点误差，16位有符号数，单位0.01ppm
```

图 4-14 外部存储顶点误差地址

4.5 软件补偿结果

1、软件温度补偿之前通过频率计测量 FOUT/PG6 结果如图 4-15 所示

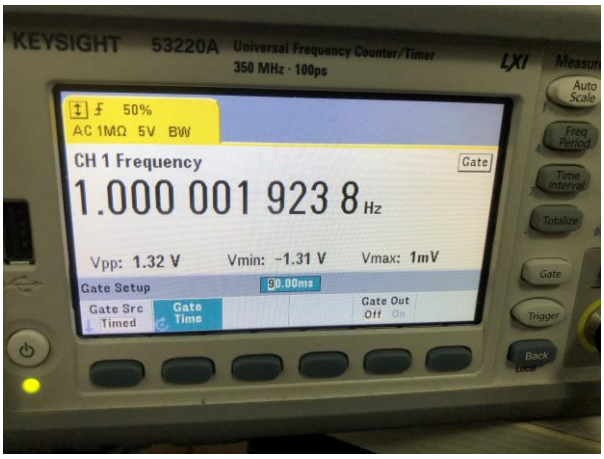

图 4-15 未顶点误差补偿 1HZ 输出

2、通过串口助手下发 1ppm 顶点误差到 flash 的 0x000001FE00 处，通过频率计测量温度补偿结果如图 4-16 所示.

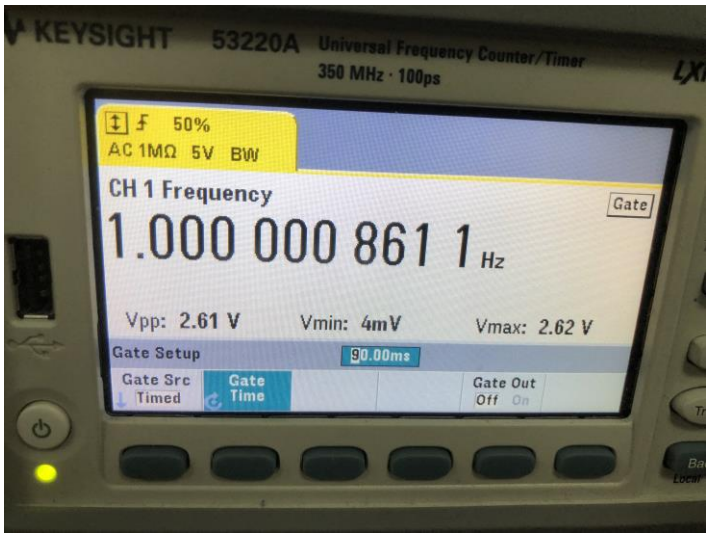

图 4-16 采用顶点误差补偿 1HZ 输出

实验结果说明：采用串口下发 1ppm 的顶点误差补偿，输出精准 1HZ 信号。

5 RTC 自动温度补偿

5.1 自动补偿实现原理

芯片预先在 Flash 的 NVR 扇区保存了 512 字节的温补参数，由于每个调校值需要 13bit，因此实际保存 256 个温度点，即-42~+85.5℃每 0.5℃保存一个点。芯片上电后软件需要从 NVR 中读取数据并载入自动温补相关 RAM 中（RTC 备份寄存器组，地址从 0x40011200 开始）。这部分 RAM 在软件不启动自动温补的情况下，可以作为普通 RAM 使用，在启动自动温补的情况下，软件无法访问。在自动温度补偿的情况下，当 XTLF 停振时，将禁止自动温补。

启动自动温补时，需要定时开启 ADC 和 PTAT，同时需要使能 RCLP 512KHz 提供 ADC 工作时钟。当 XTLF 停振时，将禁止自动温补；当 XTLF 正常时，在自动温补启动时，硬件自动开启 RCLP 并输出 512KHz。

注：自动温补电路在启动时将独占 ADC，应当确保在启动自动温补前，ADC 未被使能

芯片温度定标在 30℃，在此温度下记 4LSB/℃斜率下的 ADC 输出。假设 4LSB/℃斜率下 ADC 输出 1400，则可以推算 85℃下输出应为 1620，-42℃下输出应为 1112；将推算-42℃下 ADC 写入到 ADOFFSET 寄存器中，作为自动温补零点修正值。

5.2 30℃温度定标与 ADOFFSETBCD

FM33G0xx ，30℃温度定标值存在 NVR 扇区中,如图 5-1 所示：

0x1FFFFC96 为 30℃温度定标值 PTAT 的 ADC 值低位

0x1FFFFC97 为 30℃温度定标值 PTAT 的 ADC 值高位

DR	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
C00	55	AA	AA	55	55	AA	AA	55	AA	55	FF	FF	02	00	04	04
C10	11	58	66	0A	14	00	22	00	36	00	FF	FF	E2	07	06	07
C20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00
C30	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C40	42	33	A0	48	31	38	15	16	07	18	80	1D	D4	05	D6	05
C50	65	05	73	F8	72	08	8E	F7	DE	80	D3	0B	44	06	3E	0B
C60	3E	0B	D1	04	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	21	03
C70	FF	FF	94	03	C7	80	20	93	FF	FF	FF	FF	34	00	00	00
C80	33	CC	CC	33	FF	FF	02	05	09	0C	12	13	17	1A	1E	23
C90	1E	00	E1	05	09	0C	AF	04	40	1C	7B	39	6C	57	BA	6A
CA0	FE	FA	F5	F1	ED	E9	E5	E0	DC	D7	01	05	09	0E	12	16

图 5-1 NVR 扇区中 30℃温度定标值

如图 5-1 中所示，该芯片的 30℃温度定标值为 0X4AF

说明：NVR 中 30℃温度定标值为晶圆温度，变为芯片后相同温度下定标值会高一些，批量测试数据表明 30℃温度下芯片 ADC 值比定标值高 7 个值。

例如某芯片的 30℃温度定标值为 0X4AF，那么实测温度对应的 ADC 值为 0X4B6。

```
WH= *(uint32_t*)0x1FFFFC94>>24;    //30度定标值高位
WL= *(uint32_t*)0x1FFFFC94>>16;    //30度定标值低位
DB30= ((WH<<8)|WL)-0x113;        //0x120为4LSB    0x11A为3.92LSB
*(uint32_t*)0x40011068=DB30;  //ADOFFSET   -42度 ADC值
*(uint32_t*)0x40000248=0x01;  //RCLF关闭
*(uint32_t*)0x40011064=0x01;  //使能自动温补
```

-42℃ ADOFFSET 寄存器地址为 0X40011068 ,根据实测成品芯片 ADC 斜率为 3.92/℃， 即（30℃-（-42℃））*3.92/℃=283=0X11A

ADOFFSET=DB30+0X07-0X11A （公式 1）

DB30：为 NVR 中 30℃晶圆温度定标值

0X07：为相同温度下芯片和晶圆定标差值

软件需要根据温度定标结果，计算零点修正值 ADOFFSET，并将此修正值写入 RTC 寄存器；自动温补时，硬件将 ADC 结果减去零点修正值后，截取 bit1~8 形成 8bit 查表地址

5.3 温度补偿数据组织结构

Flash 的 NVR 扇区中保存 256 个温度点的补偿数据，每个点 16bit，如图 5-2 所示，其中有效数据为 13bit，包含 1bit 符号位和 12bit 调校值。其余 3bit 中 2bit 用于保存校验位，1bit 为无关数据。

x	P2	P1	SIGN	CAL[11:8]	CAL[7:0]

图 5-2 补偿数据格式

其中 CAL[11:0]为调校值，SIGN 为符号位，P1=XOR{CAL[7:0]}，P2=XOR{SIGN,CAL[11:8]}

举例：35℃下晶体原始误差为 0.9999800　调校步长选择 0.119ppm，则补偿值 =20/0.119=0xA8　　SIGN=1　　P1=1　　P2=1

则温度补偿数据组织形式为：0X70A8

5.4 Flash 的 NVR 扇区写入

根据精工 VT200 晶体在不同温度点原始误差，形成全温区晶体经验曲线：

打开“FM33XX 编程器上位机” >> “配置” >>“温度补偿配置”如图 5-3 所示

图 5-3 FM33XX 编程器上位机

- 顶点温度设置为：25
- 二次曲线系数 KH 设置为：-0.0382
- 二次曲线系数 KL 设置为：-0.03

补偿值偏移：常温 1Hz 时的误差（顶点误差）（不同晶体误差不一样，需要手动写入）。例如某个晶体常温 1Hz 输出为 0.99999772，(0.99999772-1)*1000000 = -2.28 PPM，此时补偿值偏移写成 2.28

点击“下发”按钮，完成自动温补补偿数据 NVR 扇区写入。如图 5-4 所示。

图 5-4 下发完成标志

点击“读取”按钮就能看到各个温度点的补偿值，软件自动形成一个 TXT 文本，文本内记录各个温度点的补偿值，如图 5-5 所示。

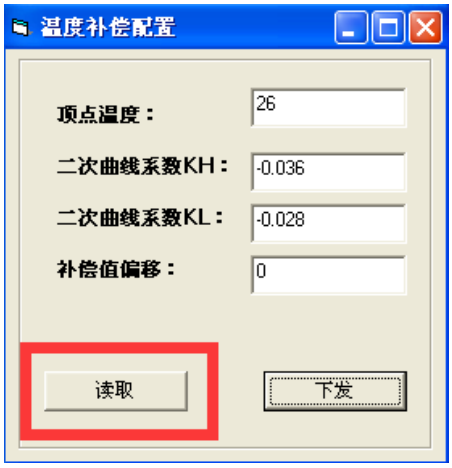

图 5-5 温度补偿配置

5.5 ADC 输出与 RF 地址映射

存放温补调校数据的是 512 字节 RF，包含 128word 地址，256 half-words，下面的讨论我们以 0~255 为完整寻址空间。ADC 输出对 RF 地址查表的设计思路是，将 30℃定标点固定指向 RF 的 144 地址，寻址 step 是 0.5℃，全地址空间代表的温度范围是 -42~+85.5℃，如表 5-1 所示：

表 5-1 全地址温度范围

addr	温度	addr	温度	addr	温度	addr	温度
0	-42	64	-10	128	22	192	54
1	-41.5	65	-9.5	129	22.5	193	54.5
2	-41	66	-9	130	23	194	55
3	-40.5	67	-8.5	131	23.5	195	55.5
4	-40	68	-8	132	24	196	56
5	-39.5	69	-7.5	133	24.5	197	56.5
6	-39	70	-7	134	25	198	57
7	-38.5	71	-6.5	135	25.5	199	57.5
8	-38	72	-6	136	26	200	58
9	-37.5	73	-5.5	137	26.5	201	58.5
10	-37	74	-5	138	27	202	59
11	-36.5	75	-4.5	139	27.5	203	59.5
12	-36	76	-4	140	28	204	60
13	-35.5	77	-3.5	141	28.5	205	60.5
14	-35	78	-3	142	29	206	61
15	-34.5	79	-2.5	143	29.5	207	61.5
16	-34	80	-2	144	30	208	62
17	-33.5	81	-1.5	145	30.5	209	62.5
18	-33	82	-1	146	31	210	63
19	-32.5	83	-0.5	147	31.5	211	63.5
20	-32	84	0	148	32	212	64
21	-31.5	85	0.5	149	32.5	213	64.5
22	-31	86	1	150	33	214	65
23	-30.5	87	1.5	151	33.5	215	65.5
24	-30	88	2	152	34	216	66

25	-29.5	89	2.5	153	34.5	217	66.5
26	-29	90	3	154	35	218	67
27	-28.5	91	3.5	155	35.5	219	67.5
28	-28	92	4	156	36	220	68
29	-27.5	93	4.5	157	36.5	221	68.5
30	-27	94	5	158	37	222	69
31	-26.5	95	5.5	159	37.5	223	69.5
32	-26	96	6	160	38	224	70
33	-25.5	97	6.5	161	38.5	225	70.5
34	-25	98	7	162	39	226	71
35	-24.5	99	7.5	163	39.5	227	71.5
36	-24	100	8	164	40	228	72
37	-23.5	101	8.5	165	40.5	229	72.5
38	-23	102	9	166	41	230	73
39	-22.5	103	9.5	167	41.5	231	73.5
40	-22	104	10	168	42	232	74
41	-21.5	105	10.5	169	42.5	233	74.5
42	-21	106	11	170	43	234	75
43	-20.5	107	11.5	171	43.5	235	75.5
44	-20	108	12	172	44	236	76
45	-19.5	109	12.5	173	44.5	237	76.5
46	-19	110	13	174	45	238	77
47	-18.5	111	13.5	175	45.5	239	77.5
48	-18	112	14	176	46	240	78
49	-17.5	113	14.5	177	46.5	241	78.5
50	-17	114	15	178	47	242	79
51	-16.5	115	15.5	179	47.5	243	79.5
52	-16	116	16	180	48	244	80
53	-15.5	117	16.5	181	48.5	245	80.5
54	-15	118	17	182	49	246	81
55	-14.5	119	17.5	183	49.5	247	81.5
56	-14	120	18	184	50	248	82
57	-13.5	121	18.5	185	50.5	249	82.5
58	-13	122	19	186	51	250	83

59	-12.5	123	19.5	187	51.5	251	83.5
60	-12	124	20	188	52	252	84
61	-11.5	125	20.5	189	52.5	253	84.5
62	-11	126	21	190	53	254	85
63	-10.5	127	21.5	191	53.5	255	85.5

需要注意，如果上述运算后结果超出了 0~255 的范围，则将小于 0 的地址固定为 0，将大于 255 的地址固定为 255，避免严重的校正错误。

5.6 参考例程使用说明

参考例程如图 5-6 所示

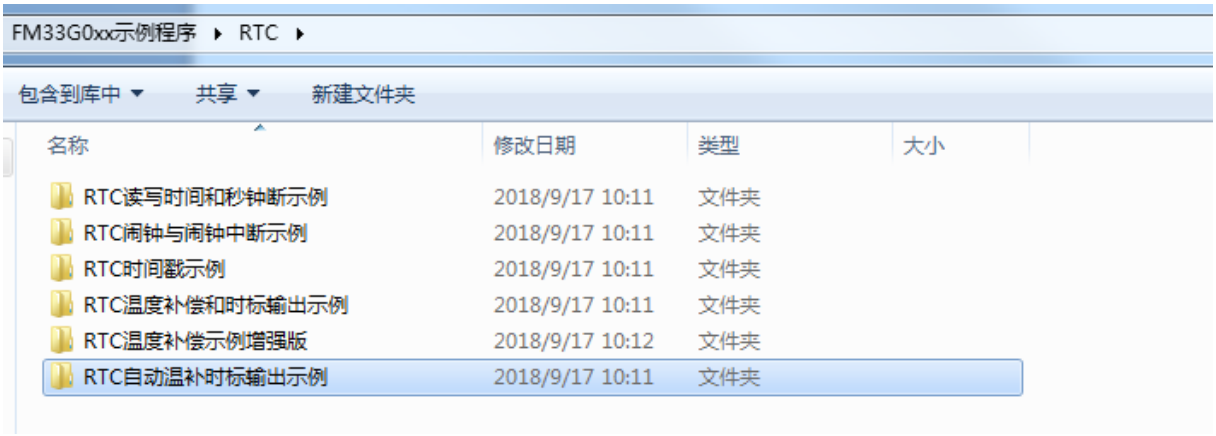

图 5-6 RTC 自动温补时标示例

需要注意，在配置 ADC 时，需要将 ADC 始终关闭掉，（自动温补采用硬件使能）中断禁止。如图 5-7 所示。

```
void ADC_Init_TsetTemperature(void)
{
  //使用简单函数配置
//  RCC_PERCLK_SetableEx(ANACCLK, ENABLE);     //模拟电路总线时钟使能
//  RCC_PERCLK_SetableEx(ADCCLK, ENABLE);    //ADC时钟使能
//  RCC_PERCLKCON2_ADCCKSEL_Set(RCC_PERCLKCON2_ADCCKSEL_RCHFDIV16); //ADC工作时钟配置，不可高于1M

  ANAC_ADC_Channel_SetEx(CH_PTAT);       //ADC输入通道选择温度传感器
  ANAC_ADCCON_ADC_IE_Setable(DISABLE);     //中断禁止
  ANAC_ADCCON_ADC_EN_Setable(DISABLE);     //ADC关闭
//  /*NVIC中断配置*/
//  NVIC_DisableIRQ(ADC_IRQn);
//  NVIC_SetPriority(ADC_IRQn,2);//中断优先级配置
//  NVIC_EnableIRQ(ADC_IRQn);
}
```

图 5-7 RTC-ADC 温度测试

程序应用步骤

- 根据晶体曲线通过上位机软件将补偿值写入 Flash 的 NVR 扇区
- 将 ADC 通道设置为 PTAT 模式
- 将 ADC trim 值写为 0x4FB，调整温度传感器斜率为 3.928LSB/℃
- LTBC 步长选择，根据温度补偿 NVR 扇区计算的补偿数据，选择 0.119 或者 0.238
- 从 NVR 读取 512bytes 温度-频率调校值，按顺序写入 RTC 备份寄存器组
- 从 NVR 读取 30C 下的温度定标值（4LSB/C 条件下），推算出-42 度下 ADC 值，并将该值写入 ADOFFSET 寄存器
- 置位 AUTOCAL.ACALEN 寄存器，使能自动温补

版本信息

版本号	发布日期	更改说明
1.0	2018.9	首次发布

附录

芯片寄存器介绍

附 1-1：RTC 控制寄存器

地址	名称	符号
0x40011000	RTC 写使能寄存器	RTCWE
0x40011004	RTC 中断使能寄存器	RTCIE
0x40011008	RTC 中断标志寄存器	RTCIF
0x4001100C	BCD 时间秒寄存器	BCDSEC
0x40011010	BCD 时间分钟寄存器	BCDMIN
0x40011014	BCD 时间小时寄存器	BCDHOUR
0x40011018	BCD 时间天寄存器	BCDDATE
0x4001101C	BCD 时间星期寄存器	BCDWEEK
0x40011020	BCD 时间月寄存器	BCDMONTH
0x40011024	BCD 时间年寄存器	BCDYEAR
0x40011028	闹钟设置寄存器	ALARM
0x4001102C	时钟信号输出控制寄存器	FSEL
0x40011030	LTBC 数值调整寄存器	ADJUST
0x40011034	LTBC 数值调整方向寄存器	ADSIGN
0x40011038	LTBC 虚拟调校使能寄存器	PR1SEN
0x4001103C	毫秒计数寄存器	MSECCNT
0x40011040	RTC 时间戳使能寄存器	STAMPEN
0x40011044	RTC 上升沿时间戳 0	CLKSTAMP0R
0x40011048	RTC 上升沿日历戳 0	CALSTAMP0R
0x4001104C	RTC 下降沿时间戳 0	CLKSTAMP0F
0x40011050	RTC 下降沿日历戳 0	CALSTAMP0F
0x40011054	RTC 上升沿时间戳 1	CLKSTAMP1R
0x40011058	RTC 上升沿日历戳 1	CALSTAMP1R
0x4001105C	RTC 下降沿时间戳 1	CLKSTAMP1F
0x40011060	RTC 下降沿日历戳 1	CALSTAMP1F
0x40011064	RTC 自动温补控制寄存器	AUTOCAL
0x40011068	自动温补零点修正值寄存器	ADOFFSET
0x4001106C	RTC 调校步长选择寄存器	CALSTEP
0x40011070	RTC 自动温补状态寄存器	CALBUSY

0x40011074	RTC 补偿周期计数器	ADJCNT
0x40011078	RTC 自动温补调校值寄存器	ACALADJ
0x40011200~0x400113FC	RTC 备份寄存器组	RTCBKPREG

RTC 写使能寄存器

名称	RTCWE							
地址	0x40011000							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							RTCWE
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	RTCWE	RTC 写使能寄存器， 当 CPU 向 RTCWE 写入 0xACACACAC 时，允许 CPU 向 RTC 的 BCD 时间寄存器写入初值，这时 RTCWE 置 1；当 CPU 向 RTCWE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 RTCWE 清 0。

RTC 中断使能寄存器

名称	RTCIE							
地址	0x40011004							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							PB5R_IE
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PB5F_IE	PB4R_IE	PB4F_IE	ADJ_IE	ALARM_IE	1KHZ_IE	256HZ_IE	64_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IE	8HZ_IE	4HZ_IE	2HZ_IE	SEC_IE	MIN_IE	HOUR_IE	DATE_IE
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:17	--	RFU：未实现，读为 0
16	PB5R_IE	PB5 上升沿中断使能。 1：中断使能打开 0：中断使能禁止
15	PB5F_IE	PB5 下降沿中断使能。 1：中断使能打开 0：中断使能禁止
14	PB4R_IE	PB4 上升沿中断使能。 1：中断使能打开 0：中断使能禁止
13	PB4F_IE	PB4 下降沿中断使能。 1：中断使能打开 0：中断使能禁止
12	ADJ_IE	调校周期中断使能。 1：中断使能打开 0：中断使能禁止
11	ALARM_IE	闹钟中断使能。 1：中断使能打开

		0：中断使能禁止
10	1KHZ_IE	1khz 中断使能。 1：中断使能打开 0：中断使能禁止
9	256HZ_IE	256hz 中断使能。 1：中断使能打开 0：中断使能禁止
8	64HZ_IE	64hz 中断使能。 1：中断使能打开 0：中断使能禁止
7	16HZ_IE	16hz 中断使能。 1：中断使能打开 0：中断使能禁止
6	8HZ_IE	8hz 中断使能。 1：中断使能打开 0：中断使能禁止
5	4HZ_IE	4hz 中断使能。 1：中断使能打开 0：中断使能禁止
4	2HZ_IE	2hz 中断使能。 1：中断使能打开 0：中断使能禁止
3	SEC_IE	秒中断使能。 1：中断使能打开 0：中断使能禁止
2	MIN_IE	分中断使能。 1：中断使能打开 0：中断使能禁止
1	HOUR_IE	小时中断使能。 1：中断使能打开 0：中断使能禁止
0	DATE_IE	天中断使能。 1：中断使能打开 0：中断使能禁止

RTC 中断标志寄存器

名称	RTCIF							
地址	0x40011008							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							PB5R_IF
位权限	U-0							R/W-0
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	PB5F_IF	PB4R_IF	PB4F_IF	ADJ_IF	ALARM_IF	1KHZ_IF	256HZ_IF	64_IF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
位名	16HZ_IF	8HZ_IF	4HZ_IF	2HZ_IF	SEC_IF	MIN_IF	HOUR_IF	DATE_IF
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

Bit	助记符	功能描述
31:17	--	RFU：未实现，读为 0
16	PB5R_IF	PB5 上升沿中断标志。写 1 清零 1：中断置位 0：无中断产
15	PB5F_IF	PB5 下降沿中断标志。写 1 清零 1：中断置位 0：无中断产
14	PB4R_IF	PB4 上升沿中断标志。写 1 清零 1：中断置位 0：无中断产
13	PB4F_IF	PB4 下降沿中断标志。写 1 清零 1：中断置位 0：无中断产
12	ADJ_IF	128 秒中断标志。写 1 清零 1：中断置位 0：无中断产生
11	ALARM_IF	闹钟中断标志。写 1 清零 1：中断置位 0：无中断产生

10	1KHZ_IF	1khz 中断标志。写 1 清零 1：中断置位 0：无中断产生
9	256HZ_IF	256hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
8	64HZ_IF	64hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
7	16HZ_IF	16hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
6	8HZ_IF	8hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
5	4HZ_IF	4hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
4	2HZ_IF	2hz 中断标志。写 1 清零 1：中断置位 0：无中断产生
3	SEC_IF	秒中断标志。写 1 清零 1：中断置位 0：无中断产生
2	MIN_IF	分中断标志。写 1 清零 1：中断置位 0：无中断产生
1	HOUR_IF	小时中断标志。写 1 清零 1：中断置位 0：无中断产生
0	DATE_IF	天中断标志。写 1 清零 1：中断置位 0：无中断产生

BCD 时间秒寄存器

名称	BCDSEC							
地址	0x4001100C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	BCDSEC						
位权限	U-0	R/W-X						
Bit	助记符	功能描述						
31:7	--	RFU：未实现，读为 0						
6:0	BCDSEC	秒时间数值，BCD 格式。						

BCD 时间分钟寄存器

名称	BCDMIN							
地址	0x40011010							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	BCDMIN						
位权限	U-0	R/W-X						

Bit	助记符	功能描述
31:7	--	RFU：未实现，读为 0
6:0	BCDMIN	分钟时间数值，BCD 格式。

BCD 时间小时寄存器

名称	BCDHOUR							
地址	0x40011014							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BCDHOUR					
位权限	U-0		R/W-X					

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:0	BCDHOUR	小时数值，BCD 格式。

BCD 时间天寄存器

名称	BCDDATE							
地址	0x40011018							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							

位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		BCDDATE					
位权限	U-0		R/W-X					

Bit	助记符	功能描述
31:6	--	RFU：未实现，读为 0
5:0	BCDDATE	天数值，BCD 格式。

BCD 时间星期寄存器

名称	BCDWEEK							
地址	0x4001101C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-					BCDWEEK		
位权限	U-0					R/W-X		

Bit	助记符	功能描述
31:3	--	RFU：未实现，读为 0
2:0	BCDWEEK	周数值，BCD 格式。

BCD 时间月寄存器

名称	BCDMONTH							
地址	0x40011020							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-			BCDMONTH				
位权限	U-0			R/W-X				

Bit	助记符	功能描述
31:5	--	RFU：未实现，读为 0
4:0	BCDMONTH	月数值，BCD 格式。

BCD 时间年寄存器

名称	BCDYEAR							
地址	0x40011024							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	BCDYEAR							
位权限	R/W-X							

Bit	助记符	功能描述

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	BCDYEAR	年数值，BCD 格式。

闹钟设置寄存器

名称	ALARM							
地址	0x40011028							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		ALARMHOUR					
位权限	U-0		R/W-00					
位	Bit15	Bit14	BIT13	BIT12	BIT11	BIT10	Bit9	Bit8
位名	-	ALARMMIN						
位权限	U-0	R/W-00						
位	Bit7	Bit6	BIT5	BIT4	BIT3	BIT2	Bit1	Bit0
位名	-	ALARMSEC						
位权限	U-0	R/W-00						

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
21:16	ALARMHOUR	闹钟的小时数值。
15	--	RFU：未实现，读为 0
14:8	ALARMMIN	闹钟的分数值。
7	--	RFU：未实现，读为 0
6:0	ALARMSEC	闹钟的秒数值。

时钟信号输出控制寄存器

名称	FSEL							
地址	0x4001102C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-				FSEL			
位权限	U-0				R/W-00			

Bit	助记符	功能描述
31:4	--	RFU：未实现，读为 0
3:0	FSEL	频率输出选择信号： 4’b0000：输出 PLL 分频得到的精确 1 秒方波 4’b0001：输出 PLL 分频的高电平宽度 80ms 的秒时标 4’b0010：输出秒计数器进位信号，高电平宽度 1s 4’b0011：输出分计数器进位信号，高电平宽度 1s 4’b0100：输出小时计数器进位信号，高电平宽度 1s 4’b0101：输出天计数器进位信号，高电平宽度 1s 4’b0110：输出闹钟匹配信号 4’b0111：输出 128 秒方波信号 4’b1000：反向输出 PLL 分频的高电平宽度 80ms 的秒时标 4’b1001：反向输出秒计数器进位信号 4’b1010：反向输出分计数器进位信号 4’b1011：反向输出小时计数器进位信号 4’b1100：反向输出天计数器进位信号

Bit	助记符	功能描述
		4’b1101：反向输出闹钟匹配信号 4’b1110：反向输出 PLL 分频的精确 1s 方波信号 4’b1111：输出 RTC 内部秒时标方波

LTBC 数值调整寄存器

名称	ADJUST							
地址	0x40011030							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					ADJUST[10:8]		
位权限	U-0					R/W-xxx		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJUST[7:0]							
位权限	R/W-xxxxxxxx							

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
10:0	ADJUST	LTBC 补偿调整数值

LTBC 数值调整方向寄存器

名称	ADSIGN							
地址	0x40011034							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							ADSIGN
位权限	U-0							R/W-x

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	ADSIGN	LTBC 补偿方向 0：表示增加计数初值 1：表示减少计数初值

LTBC 虚拟调校使能寄存器

名称	PR1SEN							
地址	0x40011038							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							PR1SEN

位权限	U-0	R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	PR1SEN	虚拟调校使能信号 0：表示禁止虚拟调校功能 1：表示使能虚拟调校功能

毫秒计数寄存器

名称	MSECCNT							
地址	0x4001103C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	MSCNT							
位权限	R/W-xxxxxxxx							

Bit	助记符	功能描述
31:8	--	RFU：未实现，读为 0
7:0	MSCNT	毫秒计数器值。以 256Hz 为周期计数，精度 3.9ms。

RTC 时间戳使能寄存器

名称	STAMPEN							
地址	0x40011040							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							

位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-						STAMP1EN	STAMP0EN
位权限	U-0						RW-X	RW-X

Bit	助记符	功能描述
31:2	--	RFU：未实现，读为 0
1	STAMP1EN	PB5 触发的时间戳功能使能位。无复位值，建议软件上电后进行初始化。 1：打开时间戳 0：关闭时间戳
0	STAMP0EN	PB4 触发的时间戳功能使能位。无复位值，建议软件上电后进行初始化。 1：打开时间戳 0：关闭时间戳

RTC 上升沿时间戳 0

名称	CLKSTAMP0R							
地址	0x40011044							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HRSTP0R					
位权限	U-0		RW-X					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	MINSTP0R						
位权限	U-0	RW-X						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SECSTP0R						
位权限	U-0	RW-X						

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	HRSTP0R	检测到 PB4 上升沿后存储 BCD 小时寄存器的值。
15	--	RFU：未实现，读为 0
14:8	MINSTP0R	检测到 PB4 上升沿后存储 BCD 分寄存器的值。
7	--	RFU：未实现，读为 0
6:0	SECSTP0R	检测到 PB4 上升沿后存储 BCD 秒寄存器的值。

RTC 上升沿日历戳 0

名称	CALSTAMP0R							
地址	0x40011048							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	YRSTP0R							
位权限	RW-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名				MONSTP0R				
位权限				RW-X				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					WKSTP0R		
位权限	U-0					RW-X		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAYSTP0R					
位权限	U-0		RW-X					

Bit	助记符	功能描述
31:24	YRSTP0R	检测到 PB4 上升沿后存储 BCD 年寄存器的值。
23:21	--	RFU：未实现，读为 0
20:16	MONSTP0R	检测到 PB4 上升沿后存储 BCD 月寄存器的值。
15:11	--	RFU：未实现，读为 0
10:8	WKSTP0R	检测到 PB4 上升沿后存储 BCD 周寄存器的值。
7:6	--	RFU：未实现，读为 0
5:0	DAYSTP0R	检测到 PB4 上升沿后存储 BCD 天寄存器的值。

RTC 下降沿时间戳 0

名称	CLKSTAMP0F							
地址	0x4001104C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HRSTP0F					
位权限	U-0		RW-X					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	MINSTP0F						
位权限	U-0	RW-X						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SECSTP0F						
位权限	U-0	RW-X						

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	HRSTP0F	检测到 PB4 下降沿后存储 BCD 小时寄存器的值。
15	--	RFU：未实现，读为 0
14:8	MINSTP0F	检测到 PB4 下降沿后存储 BCD 分寄存器的值。
7	--	RFU：未实现，读为 0
6:0	SECSTP0F	检测到 PB4 下降沿后存储 BCD 秒寄存器的值。

RTC 下降沿日历戳 0

名称	CALSTAMP0F							
地址	0x40011050							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	YRSTP0F							
位权限	RW-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名				MONSTP0F				
位权限				RW-X				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					WKSTP0F		

位权限	U-0					RW-X		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAYSTP0F					
位权限	U-0		RW-X					

Bit	助记符	功能描述
31:24	YRSTP0F	检测到 PB4 下降沿后存储 BCD 年寄存器的值。
23:21	--	RFU：未实现，读为 0
20:16	MONSTP0F	检测到 PB4 下降沿后存储 BCD 月寄存器的值。
15:11	--	RFU：未实现，读为 0
10:8	WKSTP0F	检测到 PB4 下降沿后存储 BCD 周寄存器的值。
7:6	--	RFU：未实现，读为 0
5:0	DAYSTP0F	检测到 PB4 下降沿后存储 BCD 天寄存器的值。

RTC 上升沿时间戳 1

名称	CLKSTAMP1R							
地址	0x40011054							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HRSTP1R					
位权限	U-0		RW-X					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	MINSTP1R						
位权限	U-0	RW-X						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SECSTP1R						
位权限	U-0	RW-X						

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	HRSTP1R	检测到 PB5 上升沿后存储 BCD 小时寄存器的值。
15	--	RFU：未实现，读为 0
14:8	MINSTP1R	检测到 PB5 上升沿后存储 BCD 分寄存器的值。
7	--	RFU：未实现，读为 0
6:0	SECSTP1R	检测到 PB5 上升沿后存储 BCD 秒寄存器的值。

RTC 上升沿日历戳 1

名称	CALSTAMP1R							
地址	0x40011058							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	YRSTP1R							
位权限	RW-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名				MONSTP1R				
位权限				RW-X				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					WKSTP1R		
位权限	U-0					RW-X		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAYSTP1R					
位权限	U-0		RW-X					

Bit	助记符	功能描述
31:24	YRSTP0R	检测到 PB5 上升沿后存储 BCD 年寄存器的值。
23:21	--	RFU：未实现，读为 0
20:16	MONSTP0R	检测到 PB5 上升沿后存储 BCD 月寄存器的值。
15:11	--	RFU：未实现，读为 0
10:8	WKSTP0R	检测到 PB5 上升沿后存储 BCD 周寄存器的值。
7:6	--	RFU：未实现，读为 0
5:0	DAYSTP0R	检测到 PB5 上升沿后存储 BCD 天寄存器的值。

RTC 下降沿时间戳 1

名称	CLKSTAMP1F							
地址	0x4001105C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-		HRSTP1F					
位权限	U-0		RW-X					
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-	MINSTP1F						
位权限	U-0	RW-X						
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-	SECSTP1F						
位权限	U-0	RW-X						

Bit	助记符	功能描述
31:22	--	RFU：未实现，读为 0
21:16	HRSTP1F	检测到 PB5 下降沿后存储 BCD 小时寄存器的值。
15	--	RFU：未实现，读为 0
14:8	MINSTP1F	检测到 PB5 下降沿后存储 BCD 分寄存器的值。
7	--	RFU：未实现，读为 0
6:0	SECSTP1F	检测到 PB5 下降沿后存储 BCD 秒寄存器的值。

RTC 下降沿日历戳 1

名称	CALSTAMP1F							
地址	0x40011060							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	YRSTP1F							
位权限	RW-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名				MONSTP1F				
位权限				RW-X				
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					WKSTP1F		

位权限	U-0					RW-X		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-		DAYSTP1F					
位权限	U-0		RW-X					

Bit	助记符	功能描述
31:24	YRSTP1F	检测到 PB5 下降沿后存储 BCD 年寄存器的值。
23:21	--	RFU：未实现，读为 0
20:16	MONSTP1F	检测到 PB5 下降沿后存储 BCD 月寄存器的值。
15:11	--	RFU：未实现，读为 0
10:8	WKSTP1F	检测到 PB5 下降沿后存储 BCD 周寄存器的值。
7:6	--	RFU：未实现，读为 0
5:0	DAYSTP1F	检测到 PB5 下降沿后存储 BCD 天寄存器的值。

RTC 自动温补控制寄存器

名称	AUTOCAL							
地址	0x40011064							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							ACALEN
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	ACALEN	自动温度补偿使能 1：使能自动温补 0：禁止自动温补

自动温补零点修正值寄存器

名称	ADOFFSET							
地址	0x40011068							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-					ADOFFSET[10:8]		
位权限	U-0					R/W-000		
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADOFFSET[7:0]							
位权限	R/W-00000000							

Bit	助记符	功能描述
31:11	--	RFU：未实现，读为 0
10:0	ADOFFSET	零点修正值；软件在启动温补前需要将 NVR1 中的修正值写入此寄存器

RTC 调校步长选择寄存器

名称	CALSTEP							
地址	0x4001106C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							CALSTEP
位权限	U-0							R/W-0

Bit	助记符	功能描述
31:1	--	RFU：未实现，读为 0
0	CALSTEP	LTBC 最小调校步长选择 1：0.119ppm（精度+/-0.06ppm） 0：0.238ppm（精度+/-0.119ppm）

RTC 自动温补状态寄存器

名称	CALBUSY							
地址	0x40011070							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	-							AUCALBUSY
位权限	U-0							R-0

位	位名	功能描述
31:1	--	RFU：未实现，读为 0
0	AUCALBUSY	RTC 自动温补 BUSY 位 1：RTC 占用 ADC 进行自动温补 0：RTC 未占用 ADC 进行自动温补

RTC 补偿周期计数器

名称	ADJCNT							
地址	0x40011074							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							

位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADJCNT							
位权限	R-X							

位	位名	功能描述
31:8	--	RFU：未实现，读为 0
7:0	ADJCNT	RTC 补偿周期计数器值 计数范围 0~255，当 BCD 秒寄存器有写入行为时，此计数器被清零。

RTC 自动温度补偿调校值寄存器

名称	**ACALADJ**							
地址	0x40011078							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-			AUADJUST				
位权限	U-0			R-X				
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	AUADJUST							
位权限	R-X							

位	位名	功能描述
31:13	--	RFU：未实现，读为 0
12:0	AUADJUST	RTC 自动温度补偿调校值寄存器 表示自动温补场景下，当前正在起作用的温度补偿值

RTC 备份寄存器

名称	RTCBKREG							
地址	0x40011200~0x400113FC							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	RTCBKREGx[31:24]							
位权限	R/W-X							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	RTCBKREGx[23:16]							
位权限	R/W-X							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	RTCBKREGx[15:8]							
位权限	R/W-X							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	RTCBKREGx[7:0]							
位权限	R/W-X							

位	位名	功能描述
31:0	RTCBKREGx	RTC 备份寄存器，总共 128words（512 字节），在 RTCBKP 模式下保持数据

上海复旦微电子集团股份有限公司销售及服务网点

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼
邮编：200433
电话：(86-021) 6565 5050
传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司
地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室
电话：(852) 2116 3288　2116 3338
传真：(852) 2116 0882

北京办事处
地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室
邮编：100007
电话：（86-10）8418 6608
传真：（86-10）8418 6211

深圳办事处
地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室
邮编：518028
电话：(86-0755) 8335 0911　8335 1011　8335 2011 8335 0611
传真：(86-0755) 8335 9011

台湾办事处
地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室
电话：(886-2) 7721 1889
传真：(886-2) 7722 3888

新加坡办事处
地址：237, Alexandra Road, #07-01, The Alexcier, Singapore 159929
电话：(65) 6472 3688
传真：(65) 6472 3669

北美办事处
地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA
电话：(480) 857-6500 ext 18
公司网址：http://www.fmsh.com/