# 计算机科学与技术学院

# 嵌入式系统实验报告

# （九）

姓　　名：　Banban

专　　业：　计 算 机 科 学 与 技 术

班　　级：

学　　号：

指 导 教 师：

2023 年 5 月 10 日

# 一、任务要求

1、 跑通并理解 rtcdemo1 项目

2、 使用 8 个数码管显示 rtc 时间（形如：23-59-59），时间校准借助串口输入，可加入整点报时

# 二、实验报告要求

1、 任务 2 中自编程序的源代码（加上注释）

2、 能说明软件仿真结果的截图、反映硬件电路连接和硬件验证结果的图片或视频

# 三、实验过程

## 一. 任务一：跑通并理解 rtcdemo1 项目

### 1. 代码

```
// main.c
// ----------------------------------------------------------
int main(void) {
    USART1_Config();    /* 配置 USART1 */
    NVIC_Configuration();    /* 配置 RTC 秒中断优先级 */
    printf("\r\n This is a RTC demo...... \r\n");
    if (BKP_ReadBackupRegister(BKP_DR1) != 0xA5A5) {
        /* 备份数据寄存器值不正确或尚未编程（第一次执行程序时） */
        printf("\r\nThis is a RTC demo!\r\n");
        printf("\r\n\n RTC not yet configured....");
        RTC_Configuration();  /* 配置 RTC */
        printf("\r\n RTC configured....");
        Time_Adjust();  /* 根据用户在超级终端上输入的值调整时间 */
        BKP_WriteBackupRegister(BKP_DR1, 0xA5A5);
    } else {
        /* 检查是否设置了上电复位标志位 */
```

```
        if (RCC_GetFlagStatus(RCC_FLAG_PORRST) != RESET) {
            printf("\r\n\n Power On Reset occurred....");
        }
        /* 检查是否设置了引脚复位标志位 */
        else if (RCC_GetFlagStatus(RCC_FLAG_PINRST) != RESET) {
            printf("\r\n\n External Reset occurred....");
        }
        printf("\r\n No need to configure RTC....");
        RTC_WaitForSynchro();/* 等待 RTC 寄存器同步 */
        RTC_ITConfig(RTC_IT_SEC, ENABLE); /* 使能 RTC 秒中断 */
        RTC_WaitForLastTask();  /* 等待 RTC 寄存器上一次写操作完成 */
    }

#ifdef RTCClockOutput_Enable
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR | RCC_APB1Periph_BKP,
ENABLE); /* 使能 PWR 和 BKP 时钟 */
    PWR_BackupAccessCmd(ENABLE);   /* 允许访问 BKP 域 */
    /* 禁用防篡改引脚 */
    BKP_TamperPinCmd(DISABLE); /* 要在防篡改引脚上输出 RTCCLK/64，必须禁用防篡
改功能 */
    /* 在防篡改引脚上使能 RTC 时钟输出 */
    BKP_RTCOutputConfig(BKP_RTCOutputSource_CalibClock);
#endif
    RCC_ClearFlag();    /* 清除复位标志位 */
    Time_Show();  /* 在无限循环中显示时间 */
    while (1) {}
}

// rtc.c
// ------------------------------------------------------------
// 返回用户在超级终端中输入的时间值，并将值储存在 RTC 计数寄存器中
uint32_t Time_Regulate(void) {
    uint32_t Tmp_HH = 0xFF, Tmp_MM = 0xFF, Tmp_SS = 0xFF;
    printf("\r\n=======================================");
    printf("\r\n============ Time Settings ============");
```

```
    printf("\r\n=======================================");
    printf("\r\n  Please Set Hours: ");

    while (Tmp_HH == 0xFF) {
        Tmp_HH = USART_Scanf(23);
    }
    printf(":  %d", Tmp_HH);
    printf("\r\n  Please Set Minutes: ");
    while (Tmp_MM == 0xFF) {
        Tmp_MM = USART_Scanf(59);
    }
    printf(":  %d", Tmp_MM);
    printf("\r\n  Please Set Seconds: ");
    while (Tmp_SS == 0xFF) {
        Tmp_SS = USART_Scanf(59);
    }
    printf(":  %d", Tmp_SS);

    /* Return the value to store in RTC counter register */
    return ((Tmp_HH * 3600 + Tmp_MM * 60 + Tmp_SS));
}
```

## 2. 图片效果

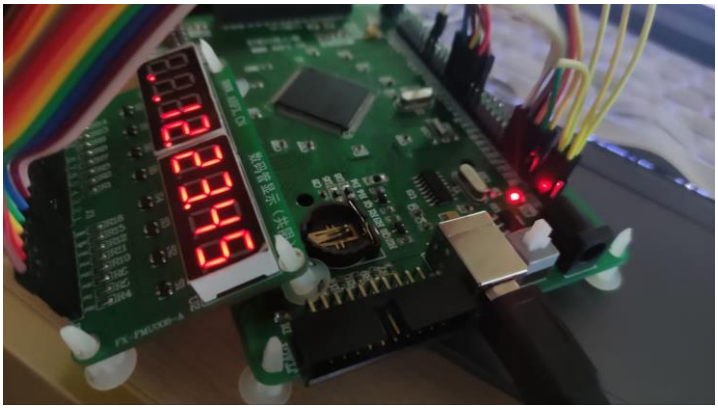
## 二. 任务二：使用 8 个数码管显示 rtc 时间（形如：23-59-59），时间校准借助串口输入，可加入整点报时

### 1. 代码

```
// rtc.c
// -----------------------------------------------------------
void Time_Show(void) {
    printf("\n\r");

    /* Infinite loop */
    while (1) {
        /* If 1s has paased */
        if (TimeDisplay == 1) {
            /* Display current time */
            Time_Display(RTC_GetCounter());  // 获取 RTC 计数器的值
            TimeDisplay = 0;
        }
    }
}

// 描述  ：串口从超级终端中获取数值
uint8_t USART_Scanf(uint32_t value) {
    uint32_t index = 0;
    uint32_t tmp[2] = {0, 0};

    while (index < 2) {
        /* Loop until RXNE = 1 */
        while (USART_GetFlagStatus(USART1, USART_FLAG_RXNE) == RESET) {
        }
        tmp[index++] = (USART_ReceiveData(USART1));
        // 从串口终端里面输进去的数是 ASCII 码值
        if ((tmp[index - 1] < 0x30) || (tmp[index - 1] > 0x39)) {
            printf("\n\rPlease enter valid number between 0 and 9");
            index--;
        }
```

```
    }
    /* Calculate the Corresponding value */
    index = (tmp[1] - 0x30) + ((tmp[0] - 0x30) * 10);
    /* Checks */
    if (index > value) {
        printf("\n\rPlease enter valid number between 0 and %d", value);
        return 0xFF;
    }
    return index;
}
```

2. 图片效果



## 四、总结与分析

本次实验我们利用 STM32 的 RTC 实现了一个简易的电子时钟，并能够通过数码管进行实时显示。在实验中，我们需要了解 RTC 的基本原理和使用方法，同时需要学会如何使用 GPIO 口和定时器控制数码管的显示。首先，在使用 RTC 之前，我们需要对 RTC 进行初始化。

我们需要配置 RTC 的时钟源，使能 RTC 时钟，以及设置 RTC 的时间和日期，其中时间和日期可以通过电脑终端进行输入和修改。通过在每次进入 while 循环时读取 RTC 的时间和日期，我们就能够实现实时显示时钟。我们将切换数码管的显示位置，并根据当前显示的数字计算出应该在数码管上显示的数字和对应的字节值，并存储到一个数组中。最后，我们再将数组中的数字写入到数码管

的寄存器中，以实现数码管的显示。

在这个实验中，我们还需要注意一些细节问题，例如数码管的显示顺序、显示格式的设置、电路连接的正确性等。我们需要仔细检查电路连线是否正确，以及数码管是否显示不正常的问题，并在需要时进行调试和修正。