



机械与能源工程系

SUSTech Department of  
Mechanical and Energy  
Engineering

# 实 验 报 告

课程名称： 机电一体化

课程编号： ME3333

实验题目： RTC 时钟实验

学 号： 12313215

姓 名： 梁家源

专 业： 机器人工程

指导教师： 柯文德 曾千里

实验成绩：

实验日期： 2025 年 4 月 28 日

# RTC 时钟实验

## 一、实验目的

本实验旨在使用 STM32F767 的 RTC（实时时钟）模块结合 LCD 显示屏，实现一个基本的实时时钟功能。通过三个按键控制时间跳变速度（模拟 1 秒、0.5 秒、2 秒等节奏更新秒数），掌握 RTC 初始化与时间获取方法，理解 RTC 分频配置及其唤醒机制。

## 二、实验仪器和用具

主要仪器设备：

1. 计算机
2. Keil uVision 软件

## 三、实验方法与步骤

### 1 时钟原理

STM32F767 内部集成了一个低功耗实时时钟（RTC），可在系统掉电或进入待机模式时仍能保持时间运行。RTC 的主要特点包括：提供年、月、日、时、分、秒等信息；支持两个可编程闹钟（ALARM A 和 B）；周期性自动唤醒功能（WakeUp Timer）；使用外部 32.768KHz 晶振（LSE）作为时钟源，以保证高精度和低功耗。

RTC 内部使用两个 32 位寄存器（TR、DR）以 BCD 格式记录时间和日期。通过 PREDIV\_S（256）和 PREDIV\_A（128）的配置，实现 1Hz 的更新频率。

### 2. 时间获取

初次运行时从 RTC 获取当前时间，存储为模拟时钟变量 hour, min, sec。

```
// 获取当前RTC时间，作为模拟时间起点
HAL_RTC_GetTime(&RTC_Handler, &RTC_TimeStruct, RTC_FORMAT_BIN);
hour = RTC_TimeStruct.Hours;
min = RTC_TimeStruct.Minutes;
sec = RTC_TimeStruct.Seconds;
```

### 3. 控制跳变速度

通过更改 tick\_interval 来修改跳变速率

```
key = KEY_Scan(0);

if (key == KEY0_PRES)    tick_interval = 200; // 2秒加1秒
else if (key == KEY1_PRES) tick_interval = 50;  // 0.5秒加1秒
else if (key == KEY2_PRES) tick_interval = 100; // 1秒加1秒

tick++;
```

#### 4.时间显示

用 LCD\_ShowString 语句来显示时间，并通过时间进位算法准确表示时间

```
if (tick >= tick_interval)
{
    tick = 0;
    // 模拟时间增加1秒
    sec++;
    if (sec >= 60) { sec = 0; min++; }
    if (min >= 60) { min = 0; hour++; }
    if (hour >= 24) hour = 0;

    sprintf((char *)tbuf, "Time:%02d:%02d:%02d", hour, min, sec);
    LCD_ShowString(30, 140, 210, 16, 16, tbuf);

    LED0_Toggle;
}
```

### 四、实验分析及结论

#### 1. 实验结果

LCD 显示如下内容：

Apollo STM32F4/F7

RTC TEST

ATOM@ALIENTEK

2025/04/23

Time:12:45:30

按 KEY0 后，时间每 2 秒跳 1 秒；

按 KEY1 后，时间每 0.5 秒跳 1 秒；

按 KEY2 后，恢复为 1 秒跳 1 秒；

#### 2. 实验总结

本实验通过 RTC 模块和 LCD 显示实现了一个基本的数字时钟功能，重点掌握了：RTC 初始化与分频设置；时间获取与显示；利用按键控制时间更新节奏；LCD 字符串显示方法。虽然实验中模拟了时间跳变，但已经较好地展现了 RTC 的实时性和可编程性