深 圳 大 学 实 验 报 告

课程名称：智能网络与计算

学院：计算机与软件学院

专业：计算机科学与技术（创新班）

指导教师：车越岭

报告人：林宪亮 学号：2022150130 班级：国际班

实验时间：2024/10/09----2024/10/11

实验报告提交时间：2024 年 10 月 12 日

教务处制

实验一： RFID 原理与读写操作

实验目的与要求：

1. 理解定时器的工作原理。
2. 掌握定时器库函数的使用。
3. 完成定时器程序编写。
4. 掌握定时器程序调试以及寄存器查看。

方法、步骤：

1. STM32 硬件的准备与连接
2. STM32 代码下载与调试
3. 实验现象记录及描述

实验过程及内容：

1. STM32 硬件的准备与连接

我参考“附录 A”，进行设备连接。

图 1 连接设备

2. STM32 代码下载与调试

1）打开实验代码中 06-Timer\Project 目录下的 TIMER.eww 工程

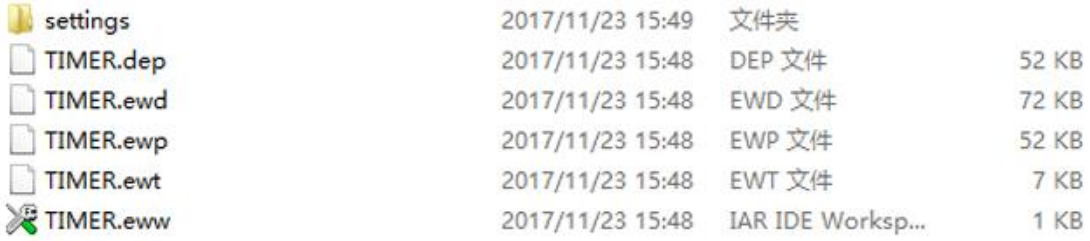


图 2 代码文件

如图 2 所示。

2）使用 IAR 开发环境打开电子时钟实验程序并阅读 readme 文件

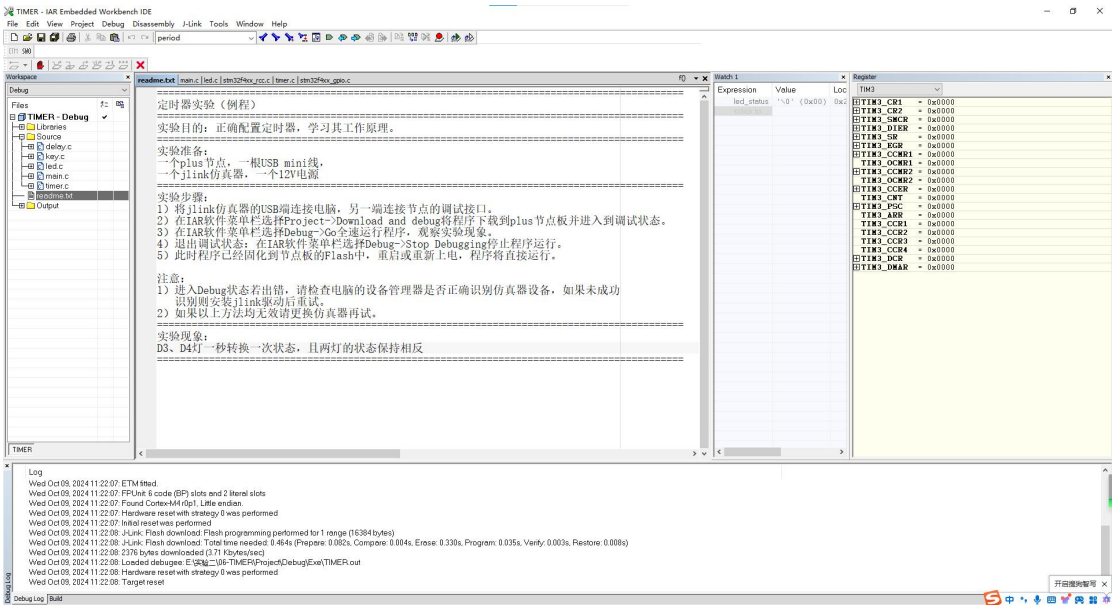


图 3 IAR

阅读 readme 文件，了解实验流程以及具体细节。

3）编译代码查看程序是否有误

图 4 编译结果

如图 4，编译无误。

4）将程序通过 J-Link 调试工具下载到 Plus 节点中，IAR 进入调试页面

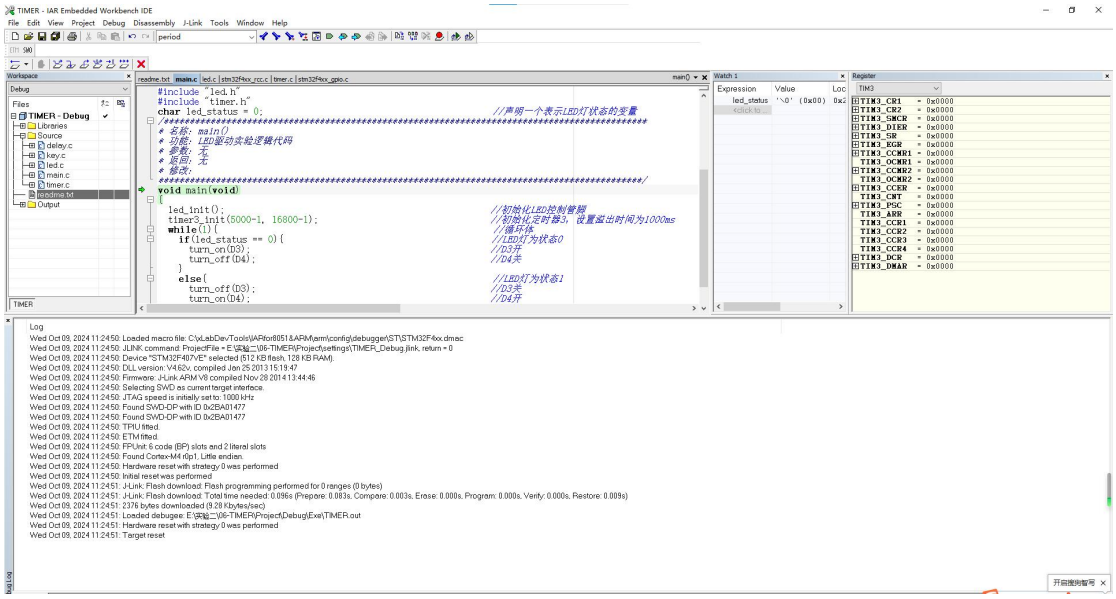


图 5 调试页面

如图所示为调试界面。

5）点击 IAR 的执行按钮（GO）执行程序，从 Plus 节点上查看实验现象

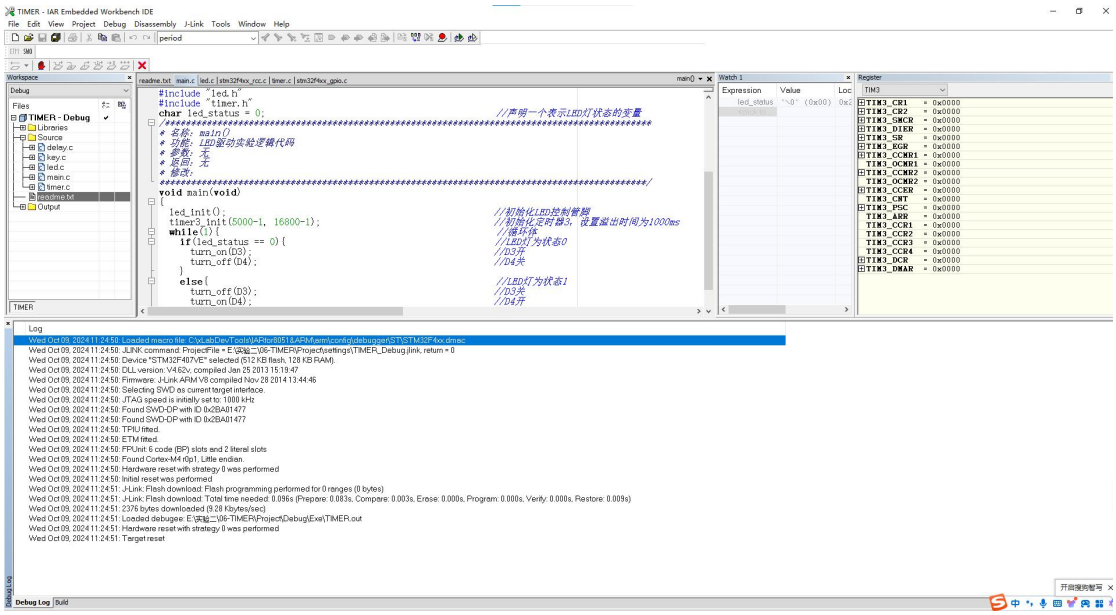


图 6 执行程序

点击执行后，我们可以看到 D3D4 交替闪烁。

6）通过 Watch 窗口 查看 LED 控制标志位 led_status 参数。

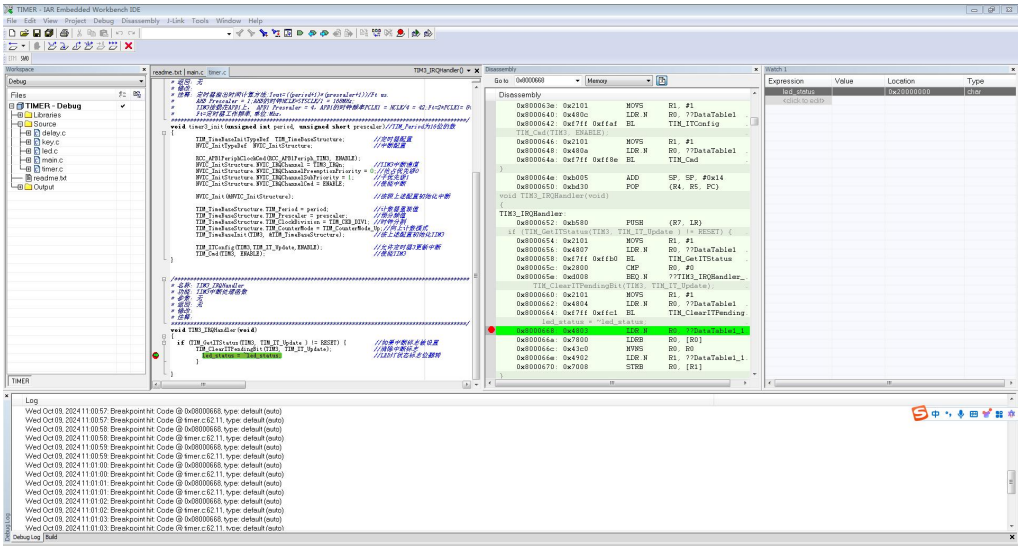


图 7 Watch 窗口

在图中位置打上断点，运行程序，经 过 1S 后程序执行到断点处，观察 led_status 状态。

7）通过 Register 窗口查看 TIM3 的计数寄存器计数值

初始 D4 亮

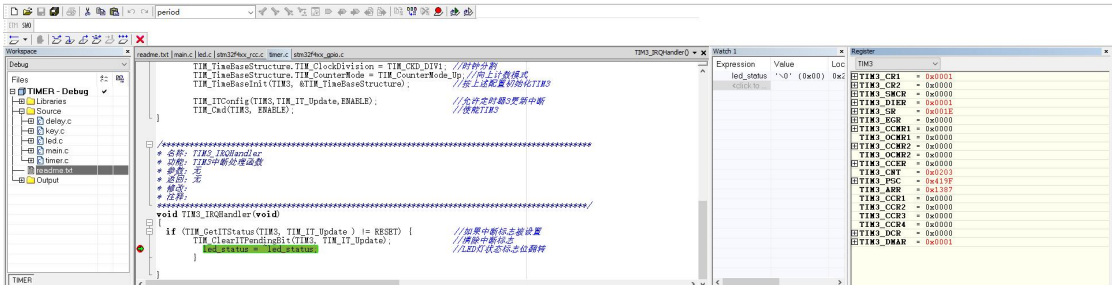


图 8 TIM3_CNT1

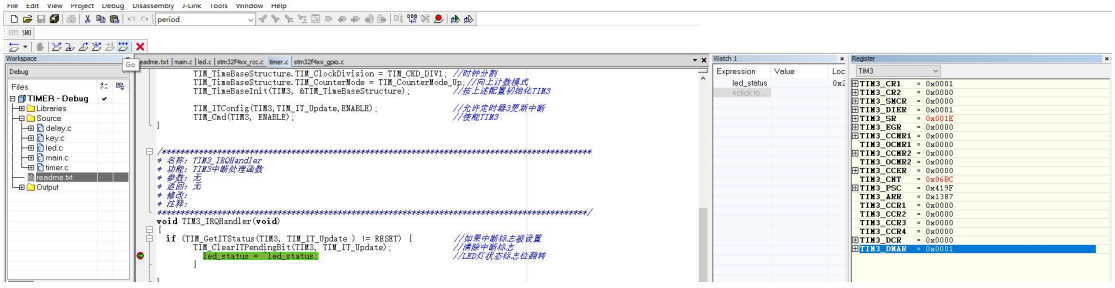


图 9 TIM3_CNT2

运行程序，执行到断点处，查看 TIM3_CNT 数值变化。
TIM3_CNT 的值持续增长。根据定时器溢出时间的计算公式：

$$T_{out} = \frac{(period + 1) \times (prescaler + 1)}{F_t}\ \mu s$$

可以得出 TIM3_CNT 每次溢出所对应的时间即为一个时钟周期。此时切换 D3 亮。

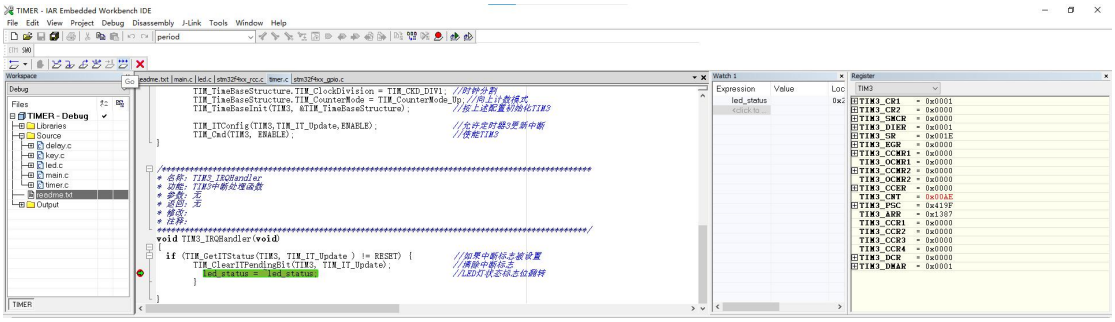


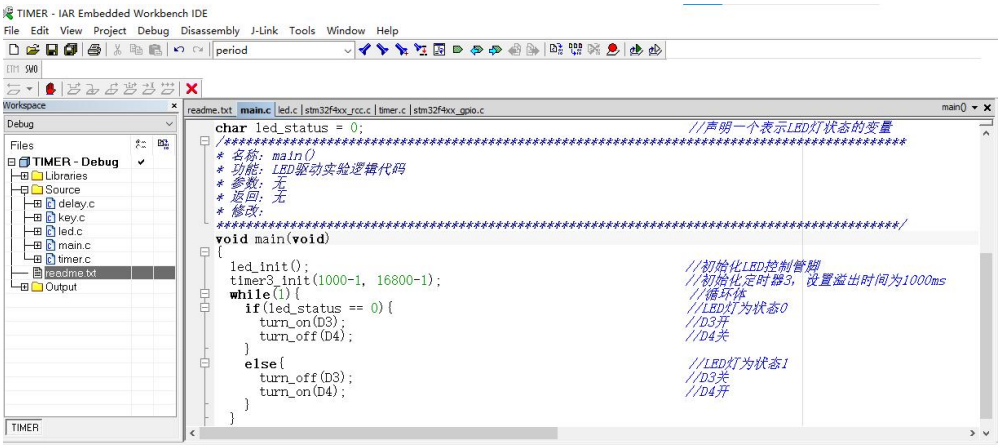
图 10　TIM3_CNT4

课程练习：改变 D3D4 闪烁频率

TIM3_CNT 的值持续增长。根据定时器溢出时间的计算公式：

$$T_{out} = \frac{(period + 1) \times (prescaler + 1)}{F_t} \mu s$$

可以看出，TIM3_CNT 每次溢出的时间对应一个时钟周期。当将 period 参数从 5000 改为 1000 时，由于 Tout_{out}out 与 period 呈线性关系，定时器溢出的时间缩短至原来的五分之一，因此频率相应增大，每 0.2 秒发生一次溢出。



```
char led_status = 0;                                    //声明一个表示LED灯状态的变量
/*********************************************************************************************
* 名称: main()
* 功能: LED驱动实验逻辑代码
* 参数: 无
* 返回: 无
* 修改:
*********************************************************************************************/
void main(void)
{
  led_init();                                           //初始化LED控制管脚
  timer3_init(1000-1, 16800-1);                         //初始化定时器3, 设置溢出时间为1000ms
  while(1){                                             //循环体
    if(led_status == 0){                                //LED灯为状态0
      turn_on(D3);                                      //D3开
      turn_off(D4);                                     //D4关
    }
    else{                                               //LED灯为状态1
      turn_off(D3);                                     //D3关
      turn_on(D4);                                      //D4开
    }
  }
```

图 11 改变 D3D4 闪烁频率

代码解释：

首先，通过调用 led_init()函数初始化 LED 控制管脚，以便设置相关 GPIO 端口为输出模式。接着，调用 timer3_init(5000-1, 16800-1)函数初始化定时器 3，并将溢出时间设置为 1000 毫秒（1 秒），其中 5000-1 和 16800-1 分别代表定时器的计数值和预分频值，确保定时器每经过 1000 毫秒就会溢出一次。

在无限循环体 while(1)中，程序持续执行 LED 控制逻辑。通过检查 led_status 变量的值，程序决定当前 LED 的状态。如果 led_status 为 0，表示当前 LED 状态为“关闭”，此时调用 turn_on(D3)将 D3 引脚置为高电平，打开 LED 灯 D3，同时调用 turn_off(D4)将 D4 引脚置为低电平，关闭 LED 灯 D4；若 led_status 不为 0，表示当前 LED 状态为“开启”，程序将 D3 引脚置为低电平关闭 LED 灯 D3，同时将 D4 引脚置为高电平，打开 LED 灯 D4。通过这种逻辑，LED 灯可以实现交替闪烁的效果，创造出直观的视觉表现。

实验结论：

本次实验围绕 STM32 定时器的基本原理、库函数使用及程序调试展开，旨在掌握定时器的编程与应用。实验过程中，通过理论学习和动手实践，成功实现了定时器的功能，并验证了其工作原理。

1. 定时器工作原理的理解

通过实验，深入理解了 STM32 定时器的核心工作原理：定时器的核心是计数器，它通过加 1（或减 1）来进行计数。当计数器溢出时，定时器会向 CPU 提出中断请求。在定时功能中，计数信号来源于周期性的内部时钟，而计数功能则可通过外部输入的非周期性信号来实现。理解定时器的计数、溢出及中断机制为后续编程奠定了基础。

2. 掌握定时器库函数的使用

通过 HAL 库或 LL 库，实验中学会了如何配置定时器的工作模式、预分频值及计数周期。库函数的使用简化了底层寄存器的配置，使得定时器的初始化与使用更加方便。例如，使用 HAL_TIM_Base_Init() 函数来配置定时器的初始参数，通过 HAL_TIM_Base_Start_IT() 启动定时器，并使其能够产生中断。

3. 定时器程序的编写与调试

实验中成功编写了定时器程序。首先，通过初始化函数配置定时器的各项参数，包括定时模式、计数周期及中断优先级等。启动定时器后，利用定时器的中断机制，程序能够在定时器溢出时执行中断服务例程，完成周期性任务。在调试过程中，使用 STM32 调试工具（如 ST-LINK）查看了定时器的寄存器状态，确保定时器按预期工作。同时，在中断服务例程中，避免了执行耗时的操作，确保了中断处理的实时性。

4. 寄存器的查看与调整

通过对定时器相关寄存器的观察和分析，实验验证了定时器的配置是否正确，并在调试过程中调整了预分频值和重载值，以达到精确的定时效果。寄存器的查看帮助理解了定时器的工作状态，尤其是在计数器溢出、中断触发等方面的验证起到了关键作用。

5. 实验现象与结论

在实验过程中，定时器能够按照设定的时间间隔触发中断，程序也能够正确响应定时器事件。这表明定时器的工作原理和编程方式得到了有效验证。通过调试预分频器和计数周期，实验进一步确认了定时器计数的精确性。最后，通过观察定时器中断的频率、响应时间等实验现象，得出了定时器工作正常、配置正确的实验结论。

通过本次实验，掌握了 STM32 定时器的工作原理与编程技巧，学会了使用库函数配置和控制定时器，积累了调试定时器程序的经验。对定时器的中断机制、寄存器查看以及如何通过调试工具检测定时器状态有了更深入的理解，为后续的开发和学习打下了坚实的基础。

心得体会：

在进行 STM32 定时器实验的过程中，我深刻感受到了嵌入式系统编程的复杂性与挑战。这次实验不仅让我加深了对定时器工作原理的理解，还使我学会了如何通过 STM32 的库函数来灵活配置和控制定时器。在实践中，我意识到预分频器和重

载值的精确设置对于计时精度至关重要，这些参数的微小调整都会直接影响到定时器的运行效果。
调试过程中，借助 ST-LINK 等工具查看寄存器状态，帮助我直观地理解定时器的工作情况。这一过程对于发现问题并确保定时器按预期运行起到了非常重要的作用。在编写中断服务程序时，我学会了如何高效地处理中断请求，并避免在中断处理函数中执行耗时操作，以维持系统的实时响应能力。
此外，我深刻体会到硬件连接的正确性对实验成功的重要性。任何连接错误都可能导致程序异常或无法正常运行。实验过程中的详细记录与现象观察，为我分析问题、调整参数提供了宝贵的依据，最终确保了定时器能够实现预期的功能。
这次实验不仅让我对定时器的基本功能有了全面的理解，还激发了我对其高级功能（如 PWM 生成和输入捕获）的兴趣，未来会继续深入学习这些内容。
同样值得一提的是，与同伴的讨论和合作在实验过程中发挥了关键作用。通过团队协作，我们不仅更快地解决了问题，还从不同的角度激发了新的思路与创意。
总体而言，这次实验不仅提升了我的嵌入式编程技能，也增强了我在解决实际问题时的能力，为未来的学习和职业发展奠定了坚实基础。

指导教师批阅意见：

成绩评定：　　分

指导教师签字：

年　　月　　日

备注：