中南大學
CENTRAL SOUTH UNIVERSITY

嵌入式系统 实验五
实 验 报 告

指导老师: 贺建彪 戴训华

学　　院: 计算机学院

专　　业: 物联网工程

班　　级: 物联网 1802

学　　号: 8208181125 8213180228

姓　　名: 王灏洋　王云鹏

一、 实验目的

掌握Cortex-M7 定时器的工作原理；

掌握Cortex-M7 定时器的配置及初始化方法；

通过实验掌握定时器中断的设置和使用方法；

通过实验掌握定时器中断的响应流程。

二、 实验内容

编写程序，对指定TIMER 进行初始化，完成相关寄存器的配置，完成定时器中断的设置和初始化，完成串口数据的发送与接收。实验中通过串口通信完成对定时器初值的设定，实现串口对程序的控制，最终实现LED 以不同的频率闪烁。在实验过程中学习Cortex-M7 中定时器以及定时器中断相关寄存器的设置、初始化，以及定时器中断的响应过程，进一步掌握使用串口对程序进行调试的方法。

三、 实验方法

⑴ 定时器

Cortex-M7 具有2 个高级控制定时器、10 个通用定时器、2 个基本定时器和2 个看门狗定时器。

下面以实验例程中所用的通用定时器为例，介绍Cortex-M7中定时器的主要功能和用法。

⑵ 通用定时器

通用定时器包含一个16位或32位自动重载计数器，该计数器由可编程预分频器驱动。它们可用于多种用途,包括测量输入信号的脉冲宽度(输入捕获)或生成输出波形(输出比较和PWM)。使用定时器预分频器和RCC时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。这些定时器彼此完全独立，不共享任何资源。

通用 TIMx 定时器具有以下特性：

16 位（TIM3 和 TIM4）或 32 位（TIM2 和 TIM5）递增、递减和递增/递减自动重载计数器。

16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65535 之间。

多达 4 个独立通道，可用于：

- 输入捕获

- 输出比较
- PWM 生成（边沿和中心对齐模式）
- 单脉冲模式输出

使用外部信号控制定时器且可实现多个定时器互连的同步电路。

发生如下事件时生成中断/DMA 请求：

- 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
- 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
- 输入捕获
- 输出比较

支持定位用增量（正交）编码器和霍尔传感器电路

触发输入作为外部时钟或者逐周期电流管理

本实验例程中使用定时器的时基单元功能，即定时功能。

可编程定时器的主要模块由一个 16 位/32 位计数器及其相关的自动重装寄存器组成。计数器可递增计数、递减计数或同时递增和递减计数。计数器的时钟可通过预分频器进行分频。计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包括：

计数器寄存器 (TIMx_CNT)

预分频器寄存器 (TIMx_PSC)

自动重载寄存器 (TIMx_ARR)

本实验中采用递增计数模式，计数器从0计数到自动重载值（TIMx_ARR寄存器的内容），然后重新从0开始计数并生成计数器上溢事件，即定时器中断。若采用递减计数模式，计数器从自动重载值（TIMx_ARR寄存器的内容）开始递减计数到0，然后重新从自动重载值开始计数并生成计数器下溢事件。计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位(CEN)置1时，才会启动计数器。

预分频器说明：

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

定时器频率计算方法及定时器中断原理定时器3频率计算方法如下：

TIMER3 时钟频率（TIM3CLK）为APB1 总线时钟频率（PCLK1）的2 倍，

APB1 总线时钟频率为系统时钟的1/4，

即 TIM3CLK = PCLK1 * 2，

PCLK1 = SystemCoreClock / 4，

所以 TIM3CLK = SystemCoreClock / 2。

本实验例程中需要将TIMER3 时钟频率配置为10KHz，由于采用递增计数模式，预分频计算方式如下：

Prescaler = (TIM3CLK / TIM3 counter clock) — 1

Prescaler = ((SystemCoreClock / 2) / (10KHz) — 1

所以在对定时器3 初始化时，需要按照以上方法计算预分频数值。

四、 实验步骤

⑴ 准备实验环境

使用 ULINK2 USB-JTAG 仿真器连接 ARM Cortex-M7 实验板与 PC，实验板一侧接右下方的 P1 接口。使用串口线，连接实验板右侧的串口 J3 和 PC 机的串口。

⑵ 串口接收设置

在 PC 机上运行 windows 自带的超级终端串口通信程序（波特率 115200 、1 位停止位、无校验位、无硬件流控制）；或者使用其它串口通信程序。

⑶ 打开实验例程

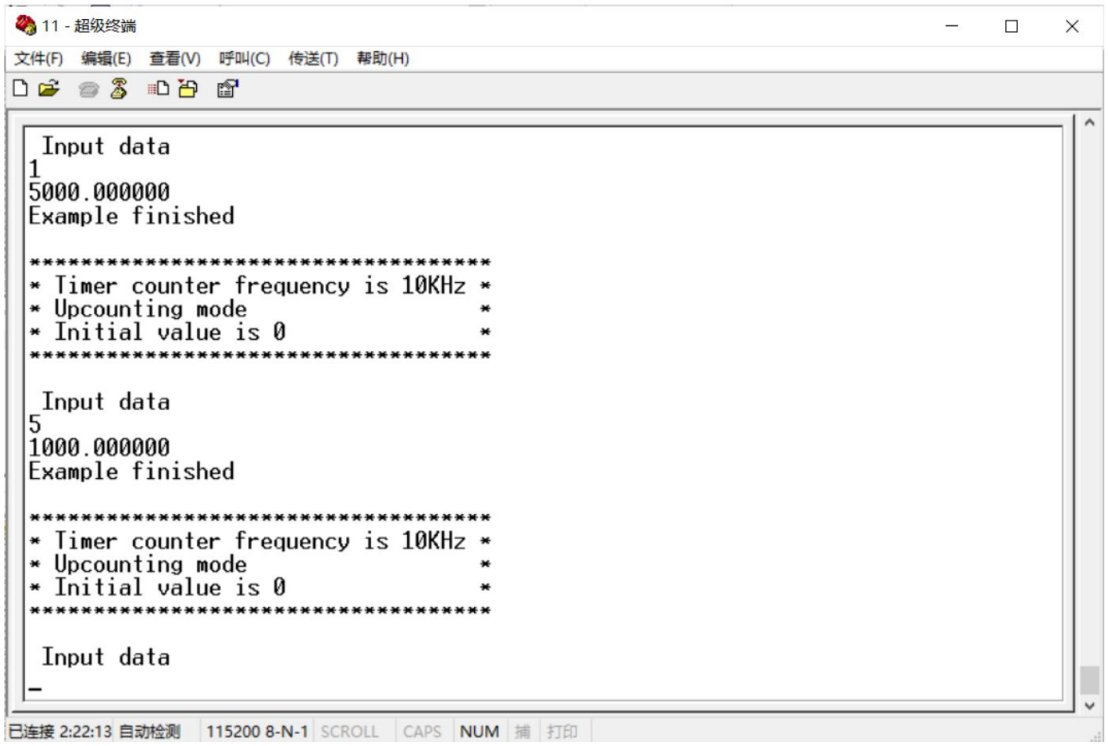
拷贝实验平台附带程序“05_TIMER”，使用μVision IDE for ARM 通过 ULINK2 USB-JTAG 仿真器连接实验板，打开工程文件，编译链接工程，根据本实验指导书中 2.3.2 小节中“编译配置”部分对工程进行配置（工程默认已经配置正确），点击 MDK 的 Project 菜单，选择 Rebuild all target files 进行编译，编译成功后，点击 Debug 菜单，选择 Start/Stop Debug Session 项或点击工具栏中的图标，下载工程生成的.axf 文件到目标板的 RAM 中调试运行。

⑷ 观察实验结果

结合实验内容和相关资料，使用一些调试命令，观察程序运行。注意观察PC 中超级终端显示信息,根据提示使用键盘输入数据,MCU 接收到数据后对TIMER3 的 TIM3_ARR 寄存器进行相应配置并完成定时器中断的初始化。初始化结束后开启 TIMER3 开始计时，产生定时器中断后对LED 执行跳变操作，即观察到LED 以高

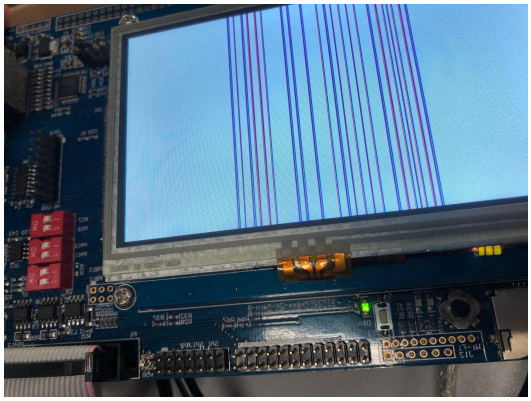
或低频率闪烁。

五、 实验结果



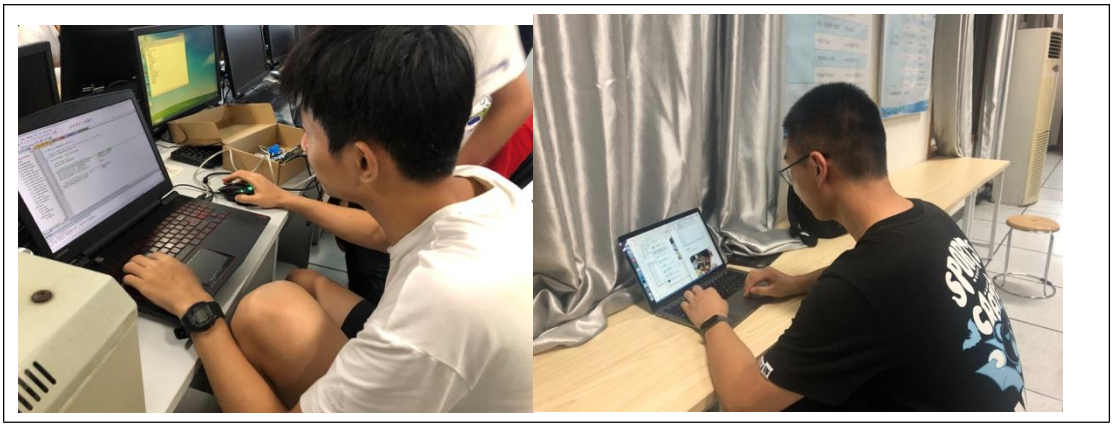
我们可以输入想实现的频率值，运行程序后小灯按输入频率进行闪烁。

如输入0.5则等2s内闪烁一次，输入1则每秒钟闪烁1次。



六、 实验感想

首先证明王灏洋和王云鹏同学来上课了。



通过本次实验，我们得知，触发的方式有上升沿和下降沿，而这两种方式的触发会有不同的效果。以前只是在课本中学习到这些知识，没有更为深入的了解，通过本次的实验，我对两种触发方式有了更为深刻的认识，同时，我们在讨论中前进，在互相帮助中一步一步完成了这个实验，有一说一，收获还是巨大的。

如果这次实验我能够完成，那么需要感谢我的老师，贺建飚老师和戴训华，他对我们的教诲如同春风化雨，润物细无声。我们不知不觉就学会了很多关于嵌入式的知识，更了解了许多嵌入式技术实际应用的生动例子。相信经过一学期的学习，我肯定学到了嵌入式的基本要领与精髓，更是能在以后的人生中披荆斩棘，所向披靡。

七、 源代码