

EI209 Final Project

王正 518021910079

2020 年 5 月 12 日

目录

1 实验要求	1
2 思路、困难及解决方法	1
2.1 整体框架及中断函数	1
2.2 第一问	2
2.3 第二问	2
2.4 第三问	2
2.5 第四问	2
2.6 其他问题	3
3 结果和现象	3
3.1 第一问	3
3.2 第二问	3
3.3 第三问	3
3.4 第四问	4
4 总结	4

1 实验要求

1. 展示当前的日期。如：今天是 4 月 28 日，则需要在数码管上显示 “0428”
2. 展示当前的时间。如：现在是 14:15，则需要在数码管上显示 “1415”。你可以设定任意的时间初始值，但时间需要每过 1 分钟变化 1 次 (这里的 1 分钟指仿真软件中的 1 分钟)。
3. 实现 1000 秒的倒计时。即初始值为 1000，每过 1 秒数字变化 1 次。
4. 通过 BT2 按钮实现上述 1) -3) 中显示内容的切换

2 思路、困难及解决方法

2.1 整体框架及中断函数

通过 8253 的 Timer0, 令每毫秒执行一次中断服务函数, 每执行一次中断服务程序, Timer-CountDisplay 和 msec 均自加 1, 当 msec=1000 时, 令 sec 和 _sec 加 1, 同时置零 msec。

在主函数中，程序通过一个无限循环来不断检查变量 TimerCountDisplay 的数值。当 DisplayCount 等于 5 时，说明已经产生了 5 次中断，时间已过去了 5ms。这时程序进入到模式选择 (CheckMode) 部分，选中模式后，跳转到相应的模式下运行代码，在数码管上显示需要显示的内容。

2.2 第一问

第一问较为简单，在选入 Mode1 后，每 5ms 刷新一位数码管，DisplayIndex 表示数码管位数 0,1,2,3，DisplayVal 表示数码管显示内容；我们通过 CMP 指令来判断数码管位数，若不符合则跳转到下一位，选中数码管位数后赋值 DisplayVal，之后调用 PLAY 函数来控制 L8255PA 和 L8255PB 使内容显示在数码管的制定位上。

因此我们分别为每一位赋值 0,5,1,3，即可显示出设定的初始日期“0513”

2.3 第二问

选入 Mode2 后，首先进行分钟数 min，小时数 hour 的计算 (setmin, sethour)。根据中断服务程序中得到的 sec 值，当 sec=60 时，令 min+1，同时 sec 置零；当 min=60 时，令 hour+1，同时置零 min 的值。这样就得到了分钟数和小时数。值得注意的是，当小时数 hour=24 时，要将 hour 置零。这一操作在 setday 中实现。

得到 hour 和 min 后，和第一问一样，要逐位显示。第 0 位和第 1 位显示小时数：在第 0 位，对 hour 进行除法操作，令 hour 除以 10，所得的商为 hour 十位上的数字，所得的余数入栈，在第 1 位时出栈并显示，即为 hour 个位上的数字。

第 2 位和第 3 位显示小时数：在第 2 位，对 min 进行除法操作，令 min 除以 10，所得的商为 min 十位上的数字，所得的余数入栈，在第 3 位时出栈并显示，即为 min 个位上的数字。

2.4 第三问

选入 Mode3 后，根据总秒数 _sec (从程序开始运行以来经过的总秒数)，得到 1000-__，该值即为要在数码管显示的数值。在第 0 位将该值除以 1000，商为该数值的千位，赋值给 DisplayVal，将余数入栈；到第 1 位，令余数出栈，成为新的被除数，除以 100，则可得到数值的百位，赋值给 DisplayVal，余数依旧入栈；到第 2 位，余数出栈，除以 10，得到数值的十位，赋值给 DisplayVal，余数入栈，在第 3 位出栈，赋值给 DisplayVal，即数值的个位。

综上，通过除法操作，将倒计时数值逐位显示在数码管上。

2.5 第四问

第 4 问对我来说是难度最大的疑问，也是调试和修改时间最长的一问。一开始我的思路是利用 74LS74 连接 8255 的 PortC，点击按钮改变 PC0 的状态，同时通过改变 PC4 来对 74LS74 来进行 reset (如图 1)，然后通过每 0.1s 检测 PortC 来实现模式的转换。但是在程序运行时一直出错，在按下第一次按钮后 PC0 就一直保持高电平状态无法复位，而我也没有检查出代码和连线的错误。

在一直 debug 无果后，我采用了另一种思路：直接将 BT2 接在 PC0 上，另一端接地 (如图 2)。每按一次按钮 PC0 被置 1，每 0.1s 检测 PC0 状态，若 PC0=1 则 Mode+1，当然还要利用 CMP 指令保证 Mode 不可以超过 3。这个简单粗暴的方法没有使用 74LS74，但实现了功能切换的效果。

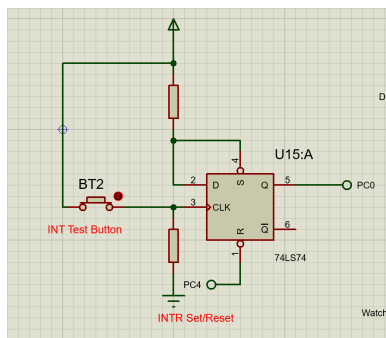


图 1: 思路一的连线

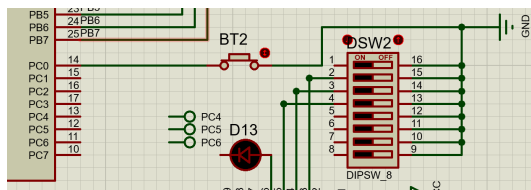


图 2: 思路二的连线

2.6 其他问题

一开始数码管的闪烁问题比较严重，于是我先调用了 delay 函数来解决，让每一位显示有一定的显示延时。然后我修改了 CPU 的频率，将原先的 500K 修改为 1M，闪烁现象大有改观。然后我又优化、精简了部分代码，减少了一些冗余的操作，尽量减少乘除而多用加减，以减少代码运行时间。最后，闪烁问题在很大程度上被解决。

3 结果和现象

3.1 第一问

数码管显示预设日期“0513”（图 3）

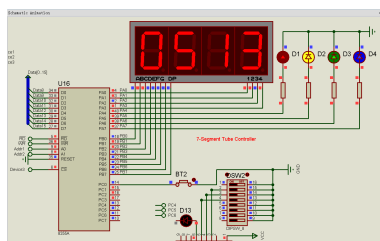


图 3: 日期显示“0513”

3.2 第二问

数码管显示初始时间“2359”，并在仿真时间一分钟后同步更新时间（图 4）。（偶尔显示更新会有几秒钟的延迟）

3.3 第三问

数码管显示 1000s 倒计时，每过 1s 显示数字减一（图 5）。

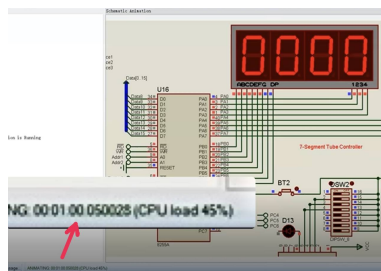


图 4: 一分钟后由“2359”变为“0000”

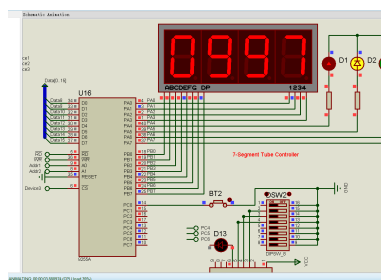


图 5: 倒计时 1000s

3.4 第四问

每按一次按钮，述 1) -3) 中显示内容，具体结果见视频。

4 总结

这次的 final project 让我学习和巩固了很多知识，如 Proteus 软件的使用，中断服务程序的使用，8255,8253 和七段数码管的工作原理，能够熟练使用一些之前并不太了解的指令，如 DIV, CMP, JNZ, 大大加强了使用汇编语言的熟练程度。