

华中科技大学 实验报告 电子器件与电路（二） 课程个人总结

学院

工程科学学院

班级

工程科学学院（生医）1701 班

姓名及学号

汪能志

U201713082

指导老师

钟国辉

2020 年 9 月 19 日

目录

1.	实验过程梳理.....	1
1.1.	实验一.....	1
1.2.	实验二.....	2
1.3.	实验三.....	2
1.4.	实验四.....	3
1.5.	实验五.....	4
2.	掌握的知识和方法.....	6
3.	实验总结与感受.....	7
4.	小组评分.....	7
5.	对课程的意见和建议.....	7

1. 实验过程梳理

1.1. 实验一

实验一开始遇到的第一个问题就是 MPLAB 集成开发环境和 XC8 编译器出现了各种各样的问题，例如 5.40 版本没有汇编器，5.35 版本无法自动补全缺失文件，XC8 找不到头文件等待。面对这些问题，首先需要确定自己在安装程序时，正确执行了每一步，特别是添加环境变量等容易出现问题的地方需要特别注意。如果还是有问题，也需要注意相关论坛的反馈，回避某些问题较多的版本。在选择版本之前，也需要留意官方给出的更新内容，确认是否对功能做出了重大更新。

之后开始进行闪灯实验。对于单片机开发板来说，闪灯是最简单的实验，这表明了开发板连接正常，可以执行指令并输出结果。但是，对于一个陌生的开发板和一个陌生的语言，这一步在开始时并不简单。首先就是需要正确设置引脚的数据方向，模数选择。一开始我选择了自己尝试，但是很快就遇到了不小的问题。之后才在 datasheet 中看到了一段用于初始化引脚的代码，通过照抄这段实例程序，才得以完成对引脚的驱动。

延时部分一开始我们组是在网上找到了一段 PIC 单片机的程序，并且可以在我们的开发板上工作，随后还在论坛上找到了诸如 PIC 单片机汇编延时程序计算小工具等。这些地方得到的代码都可以正常工作，但是对于其原理还不甚了解。

在进一步的了解 PIC 单片机相应的功能，并分析循环操作可能对应的汇编指令后，我们组才完成了通过三层循环嵌套完成定时，并对各循环计数变量的初值对定时周期进行了定量计算。



图 1 PIC 延时程序计算器

```
PS D:\WNZ\Embedded_PIC_SES_2020> python -u "d:\WNZ\Embedded_PIC_SES_2020\delay_cal.py"
Required Instruction Num 500000
n0= 0x4f
n1= 0x19
n2= 0x53
Accuracy Instruction 500000
```

图 2 PIC 延时初值计算程序输出结果

```

Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)

```

图3 循环嵌套闪灯调试结果

1.2. 实验二

实验二开始应用定时器,由于有了实验一的经验,一开始我就详细查阅了 datasheet 中关于 Timer0 的部分,并且结合老师上课所说的 8bits 定时器定时周期公式进行了分析。得到的结果和理论值差异不大,但是很快在调试中就出现了问题。首先是时间周期性小幅波动,这个问题很好解决,通过一些简单的分析,就可以确定其问题来源是查询指令消耗的周期。第二个问题就是程序周期性重启。经过讨论,我们认为这是典型的看门狗被触发,但是在开发板上并没有出现相应的问题。第二天在报告后,我们才确认到是由于 bootloader 环境下有一些额外的设置,调试时需要运行在 default 环境下。

```

Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499709 (499.709 ms)
Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499713 (499.713 ms)
Target halted. Stopwatch cycle count = 499709 (499.709 ms)

```

图4 8bits 定时器查询闪灯调试结果

1.3. 实验三

实验三的入门很简单,由于之前在 Arduino 单片机中我已经接触过中断的相关概念,对于实验二的程序进行简单改写后,就完成了 8bits 定时器中断的闪灯实验。但是由于 8bits 定时器的精度不够,我们小组又开始研究 16bits 定时器。16bits 定时器的设定时间更加精确,但是发生了我们不知道原因的周期+10,阴差阳错地正好凑齐了 500ms。随后和老师讨论中,我们得知原因是在重置定时器 TMR 时会清空分频器,因此会出现几个附加周期。

```

59  PIRObits.TMR0IF = 0x00; // clear TMR0 register
60  PORTC = PORTC;
    PORTC = ~PORTC; // flip PORTC
    isr
55  void __interrupt() isr(void)
56  {
57      TMR0H = 0x2F; //
58      TMR0L = 0x75; // (TMR0H TMR0L)=12149 65536*
59      PIRObits.TMR0IF = 0x00;
    PORTC = ~PORTC;
61  }

```

Output window shows:

```

Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
Target halted. Stopwatch cycle count = 499712 (499.712 ms)
跑表周期计数 = 499712 (499.712 ms)

```

Output window shows:

```

Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)
Target halted. Stopwatch cycle count = 500000 (500 ms)

```

图5 8bits 和 16bits 定时器中断闪灯调试结果

1.4. 实验四

实验四我设计了一个可能有点复杂的显示驱动，包括五种显示模式和多级缓冲结构。其中遇到的最大的问题就是数组索引号的计算，在开发板上进行调试相对较难。

在一开始测试时，还出现了显示乱码等问题。因此，我使用了在电脑上使用 `cpp` 工程进行调试的方法进行调试。在电脑上新建一个 `cpp` 工程，并将单片机上的中断服务程序和相关的全局变量进行适当的改动后写入该工程，需要注意的是变量的位数等细节需要保持一致。通过 `while(1)` 循环调用中断服务函数，并且将关键结果通过 `cout` 等方法输出进行查看。在编译时，也可以选择 `debug` 模式等限制更为严格的模式，将数组越界等可能的 `bug` 尽可能扼杀掉。通过调试，我将完成了对中断服务程序的软件计数的调试，同时发现之前出现的乱码问题是由于数组越界访问，导致向数码管推入了不明地址的数据。

	帧 ($5ms \times 4$)				帧					帧					帧					
触发中断计数	0	1	2	3	4	5	6	7	...	$4N-4$	$4N-3$	$4N-2$	$4N-1$		$4N$	$4N+1$	$4N+2$	$4N+3$		
扫描序号模4循环	0	1	2	3	0	1	2	3	...	0	1	2	3		0	1	2	3	...	0
帧重复计数	0				1				...	$N-1$					0				...	0
													载入下一组信号							

图 6 显示信号刷新周期

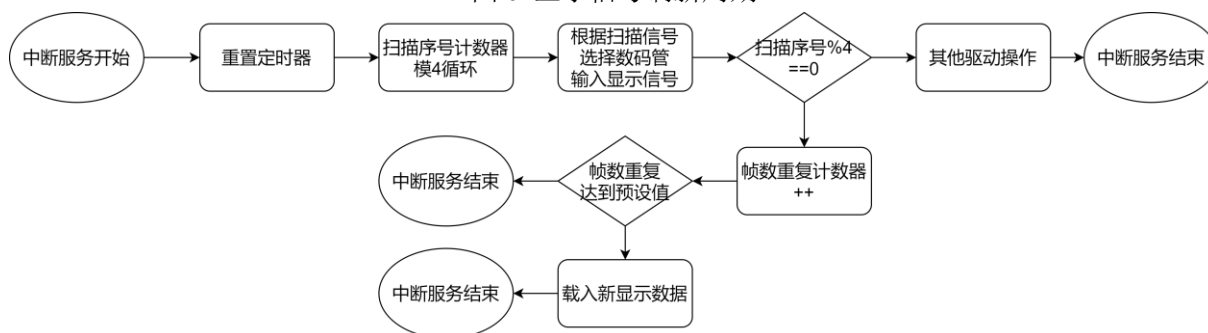


图 7 中断服务流程图

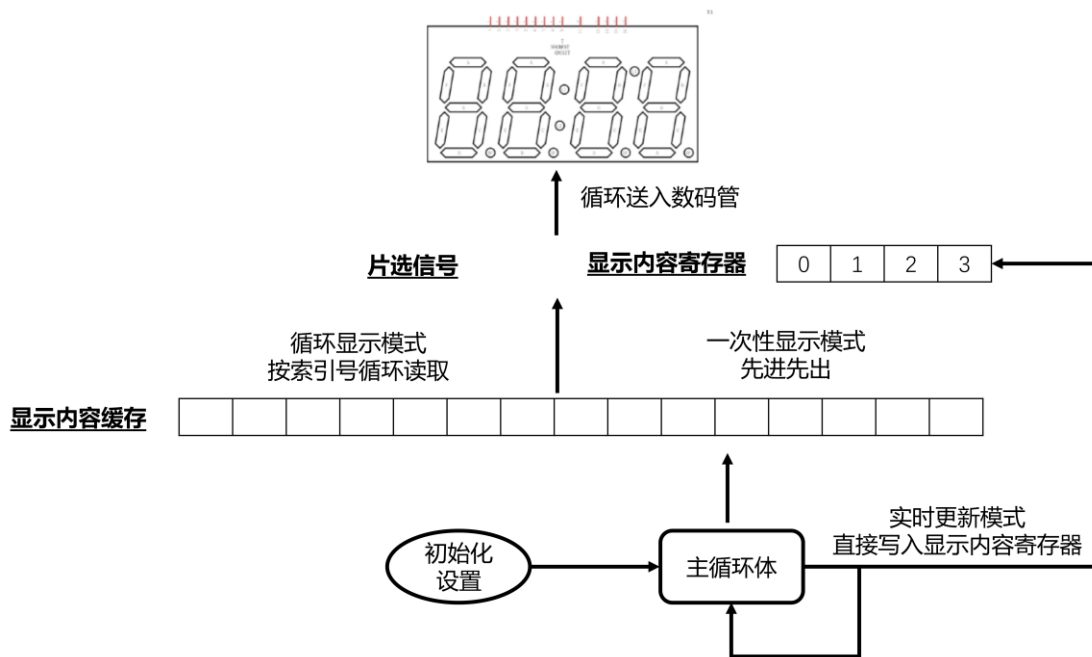


图 8 数码管驱动的缓存结构

实验中，我先后完成了三个版本的显示模块：

第一版通过向数码管输出信号时从存储显示内容的数组中循环读取数据，以显示固定的一组信号。但是输出信号无法更改，在编译时就已经固定。

第二版加入 4byte 的显示信号寄存器，向数码管输出信号时从显示信号寄存器中读取数据，通过改写显示信号寄存器实现对输出内容的更改。但是存在改写时间难以确定的问题。如果直接改写则会打断原有信号的正常输出；如果在每一组数据写完后改写，则只能使用轮询的方法，效率不足。

第三版加入了显示数据缓存，显示信号寄存器从显示数据缓存中读取数据。在需要更改输出信号时，直接修改显示数据缓存和帧索引。在下一个显示周期中，就会显示修改过的信号。

最终实现了开机时先显示开机动画，随后改为循环显示走马灯的效果。实验五中沿用了该显示模块。

1.5. 实验五

这次实验我们小组通过两种不同的方法完成了实现。（虽然我实现的方案连续和陈子康组撞方案两次）。

首先是按键扫描部分，由于和我同组的周政宏先完成了这一部分，因此我在我的程序中直接调用了这一段代码。随后我选择了降低扫描频率，同时对比多次结果的方法来判断按键动作；通过按键动作和计数器的组合，判断复合动作。但是一开始没有明确需求，导致出现了不松手就不知道是否触发长按；双击第一次被误判为单击这两个大问题。

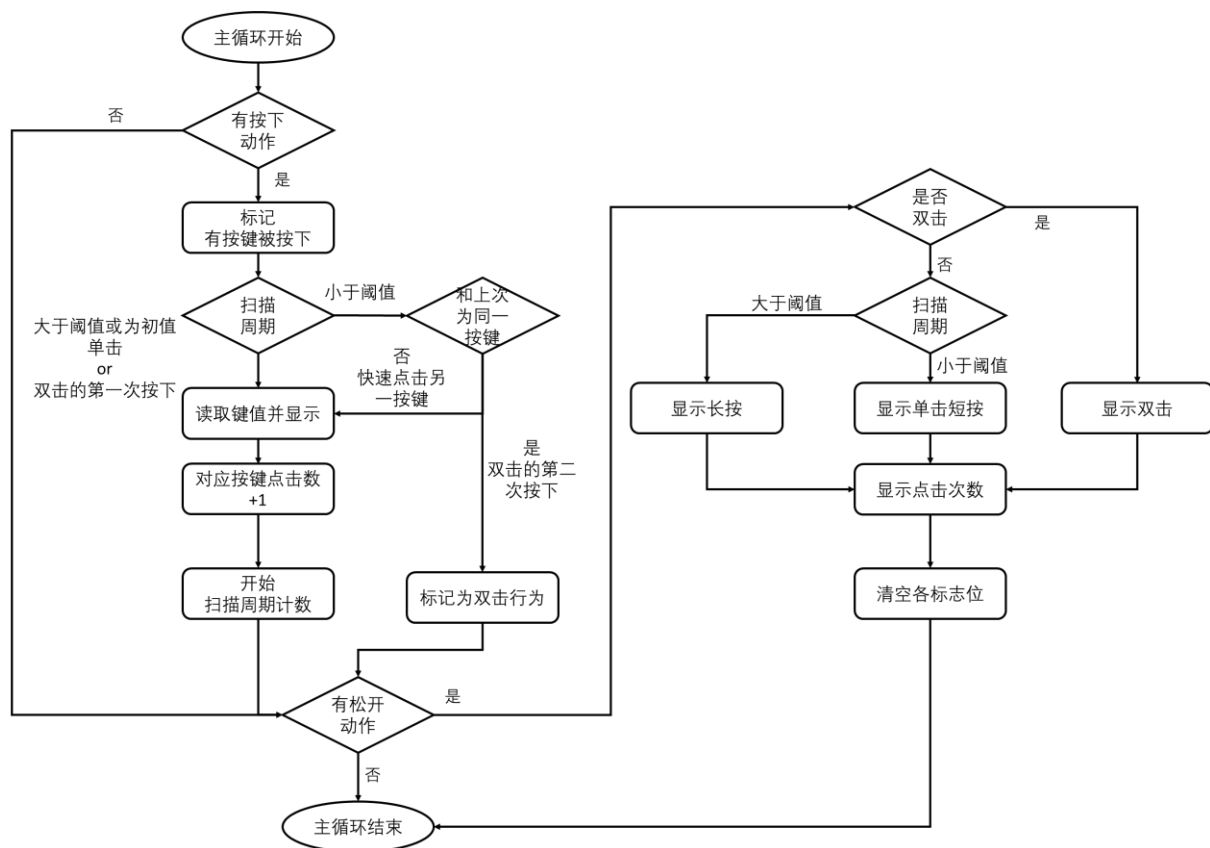


图 9 流程图 1

之后，我对于流程图进行了升级，通过增加计数器和优化判断流程，改进后的判断如下：

- 松开后一定时间后判定单击；
- 短时间内第二次按下时，判定双击；
- 按下后一定时间就判定长按；
- 做出判定后立刻清零变量，准备下一次输入。

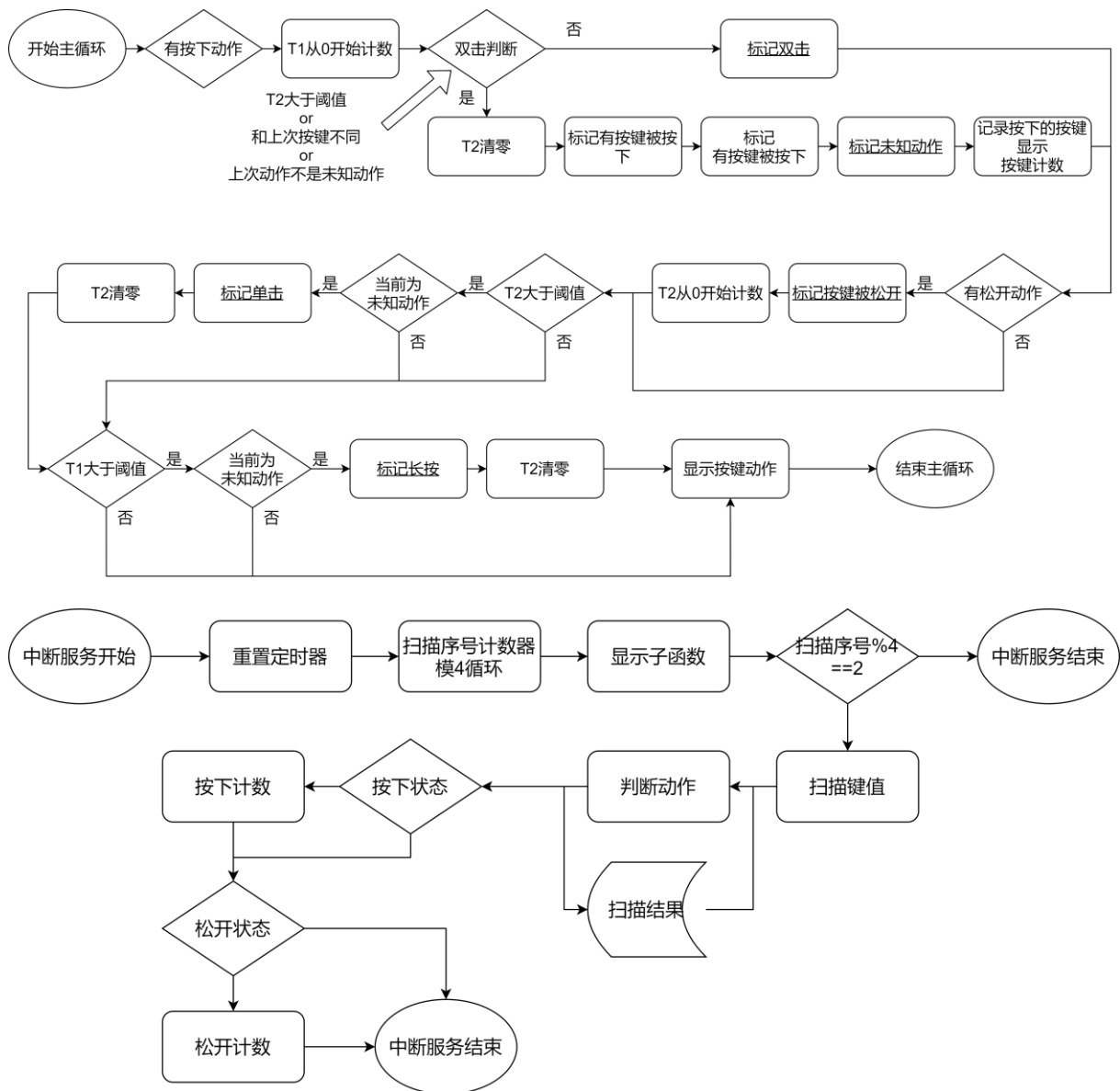


图 10 流程图 2

最终实现的结果为：

短按：计数 1 次，在松手后完成动作判断；
 长按：计数 1 次，在松手前完成动作判断；
 双击：计数 1 次，在第二次点击时完成动作判断；
 双击，第二下长按：判断为双击；
 长按后迅速短按：判断为长按+短按；
 在不同键值双击两次：计数两次，均为短按；
 快速点击三次：判断为一次双击，一次单击；
 快速点击四次：判断为两次双击。

2. 掌握的知识和方法

处理器执行指令的顺序和处理器的存储结构；
 基本的汇编语言；

多位数码管的显示方法；
按键的接线与扫描检测方法；
如何快速入门新的单片机和开发框架；
单片机程序调试方法；
汇报问题的基本方法和要点。

3. 实验总结与感受

其实本次嵌入式实验并不是我第一次接触到单片机，之前的项目实践课程和课外活动中我已经对 Arduino 单片机有了一些初步的了解。但是 Arduino 是一个已经被高度封装的平台，对其进行操作时，不需要过多地注意一些底层的内容。端口初始化，延时，屏幕驱动，键盘扫描等都有了可以直接调用的函数或库。特别是在刚接触 PIC 单片机时，就使用了汇编语言，这一直接面向处理器的低级语言。一开始其实还是很不适应的。不过，虽然不同单片机及其 IDE 之间有所差异，但是也需要注意到其中的相似部分。例如对端口数据方向的设置，对端口进行模数选择等内容，在不同的单片机上都是相通的。在完成实验一后，我对于 PIC 单片机和汇编语言有了初步的认识。在实验二中，我对定时器也有了一定的了解。后面三次实验就没有遇到太多的实现问题，主要的问题也从怎么把任务流程转换为程序语言变成了怎么设计一个满足要求的任务流程。随着实验逐步复杂，同时引入了外部输入，流程图的重要性也日益凸显。特别是在实验五的预汇报时，被老师和同学指出流程图中出现了很多问题，有的很好解决，有的只能推倒重建时。只有先完成流程图，并且尽可能在其中分析可能的执行路径，才能避免在发现问题时拆东墙补西墙和程序因为未定义输入而跑死的情况出现。

4. 小组评分

同组搭档评分，周政宏：100

在实验中，我们充分交流了对实验实现思路的意见，同时及时指出了对方程序流程中可能出现的问题并提出了检测方法。合作中也没有出现任何矛盾（技术思路矛盾除外）。

自我评分：100

独立完成了五次实验，同时对其中部分内容进行了拓展。

5. 对课程的意见和建议

由于本学期实验课安排情况特殊，其实嵌入式部分的实验时间还是非常紧张的，如果可能希望可以给更长的实验时间，方便我们完善每一次的实验。

对于按键行为检测实验，可以先鼓励同学按照先流程图，后实现的方法多设计一些方案。

在完成实验一、二后，就对数码管和按键的基本结构加以说明，以免实验三完成后没事干（也可能是因为我们的实验三正好是周末完成，时间相对充裕）。