

工作日志

标题	0914 工作日志																																																																																																																																																										
姓名	汪能志								学号				U201713082																																																																																																																																														
日期	2020年9月 14 日								天气				晴																																																																																																																																														
日记撰写时间																																																																																																																																																											
内容																																																																																																																																																											
<p>一、数码管显示原理及译码 本次实验所使用的是 4 位 8 段，共阴极数码管。由于其设计，在显示数据时，需要采用扫描刷新的形式，即：在 1-4 位依次显示数据，在不显示时数码管关闭。在扫描速度足够快时，人眼看到的就是不变的图像。</p> <p>二、调试 在下午对中断服务程序进行调试时，由于其逻辑较为复杂，在开发板上调试相对较大。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 等方法输出进行检查。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。</p> <p>三、数码管显示流程图</p> <pre>graph TD; Start([中断开始]) --> ResetTMR[重置 TMR]; ResetTMR --> IncAB[A计数器++ B计数器=A%4]; IncAB --> SelectSegment[B计数器选择帧内四个数码管之一]; SelectSegment --> BMod4{B%4==0}; BMod4 -- N --> End1([中断结束]); BMod4 -- Y --> IncC[帧数 C 计数器 ++]; IncC --> CLimit{C到达指定值?}; CLimit -- N --> End1; CLimit -- Y --> ChangeData[显示内容改变]; ChangeData --> LoadNext[载入下一组显示数据]; LoadNext --> End2([中断结束]);</pre> <p>The flowchart illustrates the logic of the interrupt service routine for displaying data on a 4-digit 8-segment common cathode LED display. It starts at "Interrupt Start", followed by resetting the timer (TMR). Then, it increments counter A and calculates B = A % 4. Counter B selects one of four digits from the frame. If B mod 4 equals 0, the loop ends. Otherwise, it proceeds to increment the frame count C. When C reaches its set value, the displayed content changes, and the next group of data is loaded before returning to the start.</p> <table border="1"> <thead> <tr> <th></th> <th colspan="4">帧</th> <th colspan="4">帧</th> <th></th> <th colspan="4">帧</th> <th colspan="4">帧</th> <th></th> <th></th> </tr> <tr> <th>触发中断计数</th> <td>1</td><td>2</td><td>3</td><td>4</td> <td>5</td><td>6</td><td>7</td><td>8</td> <td>.....</td> <td>4N - 3</td> <td>4N - 2</td> <td>4N - 1</td> <td>4N</td> <td>4N + 1</td> <td>4N + 2</td> <td>4N + 3</td> <td>4N + 4</td> <td></td> <td></td> </tr> </thead> <tbody> <tr> <td>扫描信号模 4 循环</td> <td>0</td><td>1</td><td>2</td><td>3</td> <td>0</td><td>1</td><td>2</td><td>3</td> <td>.....</td> <td>0</td><td>1</td><td>2</td><td>3</td> <td>0</td><td>1</td><td>2</td><td>3</td> <td>.....</td> <td>0</td> </tr> <tr> <td>帧重复计数</td> <td colspan="4">0</td> <td colspan="4">1</td> <td>.....</td> <td colspan="4">N-1</td> <td colspan="4">0</td> <td>.....</td> <td>0</td> </tr> <tr> <td>显示序号</td> <td colspan="8"></td> <td>.....</td> <td colspan="4">0</td> <td colspan="4">1</td> <td>.....</td> <td>0</td> </tr> <tr> <td>计数</td> <td colspan="8"></td> <td>.....</td> <td colspan="4"></td> <td colspan="4"></td> <td>.....</td> <td></td> </tr> <tr> <td></td> <td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td> <td></td> <td></td><td></td><td></td> <td>载入下一组信号</td> <td></td><td></td><td></td><td></td> <td></td> <td></td> </tr> </tbody> </table> <p>定时器中断间隔</p>																	帧				帧					帧				帧						触发中断计数	1	2	3	4	5	6	7	8	4N - 3	4N - 2	4N - 1	4N	4N + 1	4N + 2	4N + 3	4N + 4			扫描信号模 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	显示序号									0				1				0	计数																							载入下一组信号						
	帧				帧					帧				帧																																																																																																																																													
触发中断计数	1	2	3	4	5	6	7	8	4N - 3	4N - 2	4N - 1	4N	4N + 1	4N + 2	4N + 3	4N + 4																																																																																																																																										
扫描信号模 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																																																																																																																																								
显示序号									0				1				0																																																																																																																																								
计数																																																																																																																																																	
													载入下一组信号																																																																																																																																														

```
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
Target halted. Stopwatch cycle count = 5000 (5 ms)
```

显示内容刷新间隔

```
Target halted. Stopwatch cycle count = 1000000 (1 s)
Target halted. Stopwatch cycle count = 1000000 (1 s)
Target halted. Stopwatch cycle count = 1000000 (1 s)
Target halted. Stopwatch cycle count = 1000000 (1 s)
Target halted. Stopwatch cycle count = 1000000 (1 s)
```

四、定时器中断服务程序

内容显示

在每次触发中断时，对中断数量进行模 4 循环计数。通过计数信号，选通对应的数码管。通过计数信号，将 4byte 显示信号寄存器对应 byte 的值输出给管脚。

在中断数量（模 4）计数器完成一次循环时，数码管 0-3 完成一次扫描显示，称为一帧。如果不改变 4byte 显示信号寄存器中的数据，那么数码管将循环显示同样的内容。

循环显示

在完成一帧信号的扫描显示后，帧重复计数器++。

重复内容显示的操作，直到帧重复计数器达到预设值。这一帧的内容完成显示，帧序号计数器递增循环计数。为了改变显示内容，需要提取出下一组待显示的信号，用于下一个显示周期。此时根据帧序号计数器和其他显示参数，从显示数据缓存中取出对应位置的 4byte 数据，载入显示信号寄存器。在下一个周期中将显示新的内容。通过改写存储循环显示内容的内存和帧序号计数器，可以实现对循环显示内容的修改。

五、实验迭代版本

v1

循环显示固定内容：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

问题 输出信号无法更改

v2

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

问题：改写时间难以确定。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

v3

保证改写时间：加入显示数据缓存，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序号计数器。在下一个显示周期中，就会显示修改过的信号。

六、实验结果

数码管动态显示

走马灯动态显示

开机动画显示，随后改写显示内容缓存，显示走马灯