

北京交通大学汇编与接口技术实验报告

姓名：程维森

学号：21231264

实验二、利用8255A实现LED的流水点亮实验

1. 实验目的

该实验的目的在于让学生掌握8255A和微机接口的连接方法，了解8255A的基本的工作原理和编程方法。

1. 实验内容

PA口接8个拨动开关K1-K8，PB口接8个LED。初始由开关K1-K8设定8位不同的值，当执行程序后LED按K1-K8初始设定的值点亮，并向右流动（8255A工作在0方式）。选择完成在数码管上显示“8255-A”。

内容分析：

LED流水灯：  
 PA口（端口 280H）连接了8个拨动开关（K1-K8），PB口（端口 281H）连接了8个LED灯。在程序开始执行时，首先等待拨动开关K1-K8设置8位不同的值。然后，LED按照K1-K8初始设定的值点亮，并且LED灯的状态向右流动，即从K1-K8依次亮起。这个实验基于Intel 8255A芯片，该芯片提供了8位的并行输入/输出端口。

8255-A显示：  
 与LED流水灯相似，但是显示的为8255-A字样

1. 实验接线图

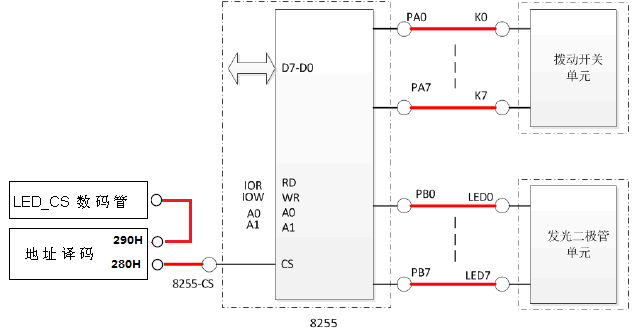
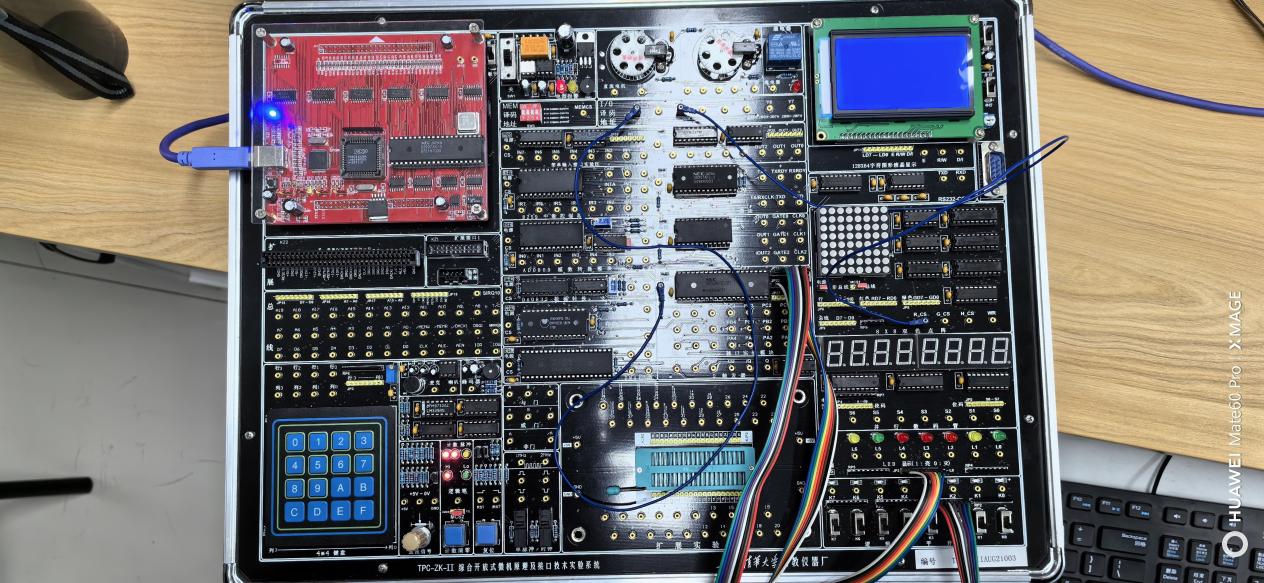
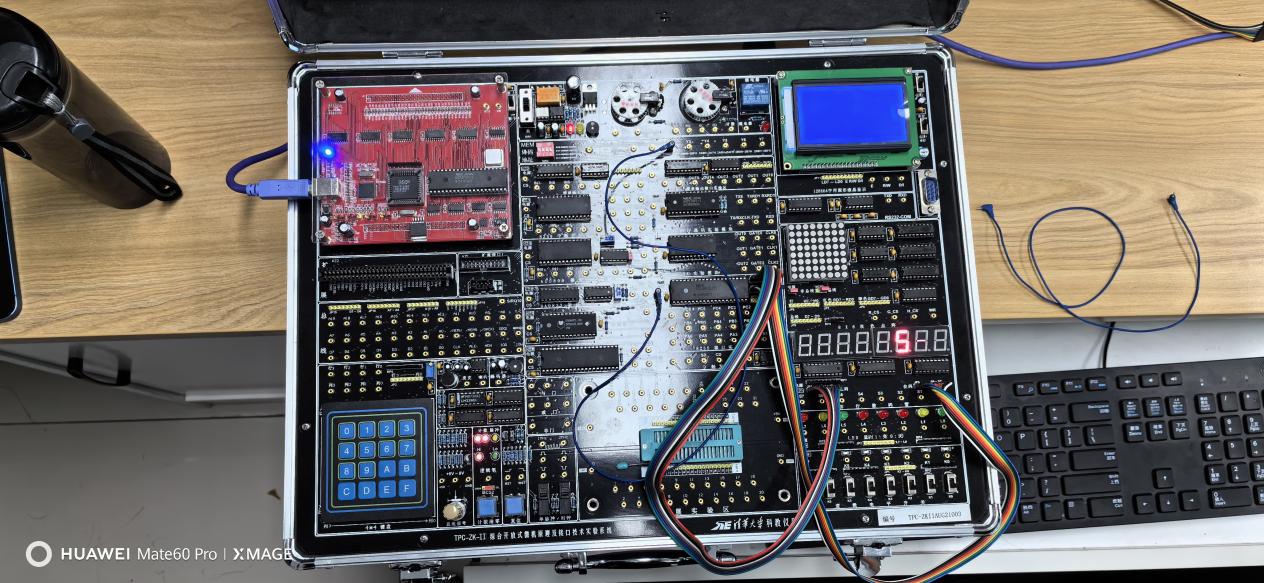


图1-3 实验连线图



LED流水灯接线图

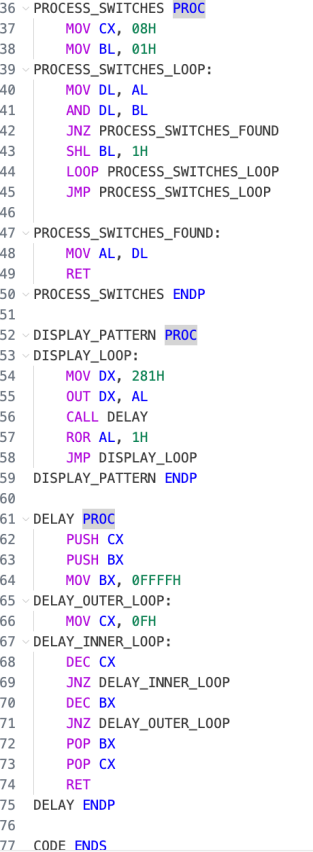
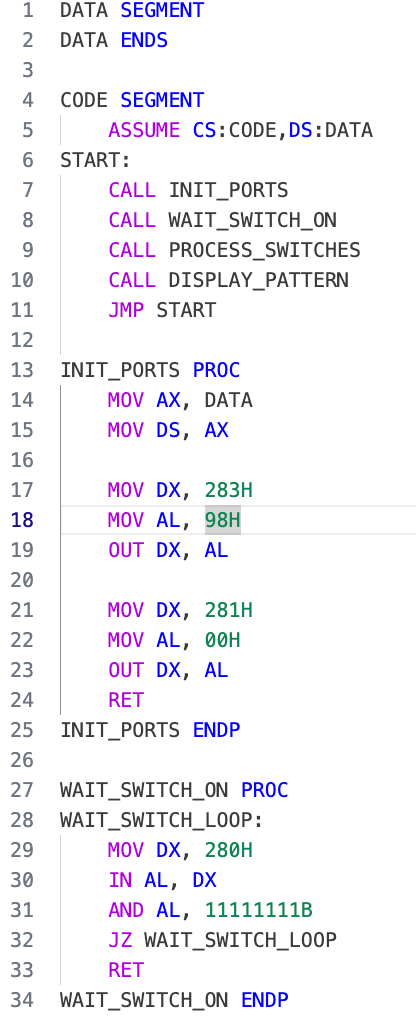


8255—A数显接线图

1. 实验编程代码

LED右流水仅仅取决于第1次开关的位置，一旦LED流水开始， LED流水的次序将不再理睬开关位置的重新变化。

提示：为了使流水显示明显，每个位置应加延时程序。



LED流水灯代码

代码分析：

INIT\_PORTS 初始化了端口，将数据加载到端口283H和281H上，用来配置8255芯片的功能。

WAIT\_SWITCH\_ON 循环等待开关被打开。在端口280H读取数据，如果某个位为1（表示开关打开），则跳出循环。

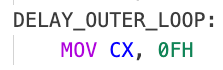
PROCESS\_SWITCHES 处理开关的状态。通过循环检查各个位，找到第一个打开的开关，然后返回这个开关的状态。

DISPLAY\_PATTERN 控制LED指示灯的显示。将找到的开关状态输出到端口281H上，然后通过DELAY过程实现延时，再将数据右移一个位，实现流水灯效果。

DELAY 实现一个简单的延时功能，通过嵌套的循环来进行延时。

代码拓展：

更改流水灯速度：

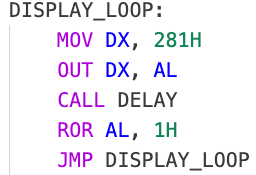


修改0FH的值即可

原理：  
在这段代码中，`DELAY\_OUTER\_LOOP` 的内部循环 `MOV CX, 0FH` 控制了外部循环的迭代次数，从而影响了延时的时长。具体地说，`MOV CX, 0FH` 指令将 CX 寄存器的值设置为 15（0FH 的十六进制表示），这就是外部循环的迭代次数。

在这个上下文中，外部循环的目的是实现一个较长的延时，用于控制LED指示灯的亮灭速度。修改这个值，即修改 `MOV CX, 0FH` 中的 `0FH`，会直接影响延时的时长。如果你想要减小延时，可以将这个值改小（例如，设置为 `MOV CX, 07H`，对应十进制的 7）。如果想要增加延时，可以将这个值增大（例如，设置为 `MOV CX, 1EH`，对应十进制的 30）。

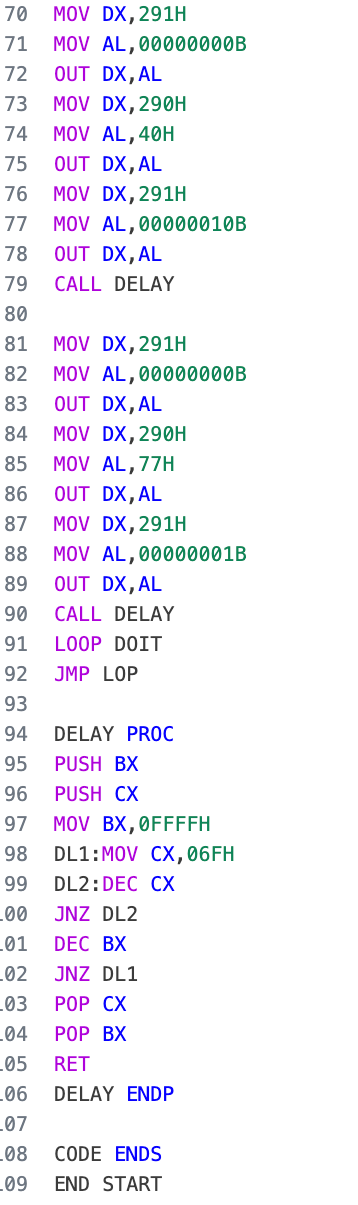
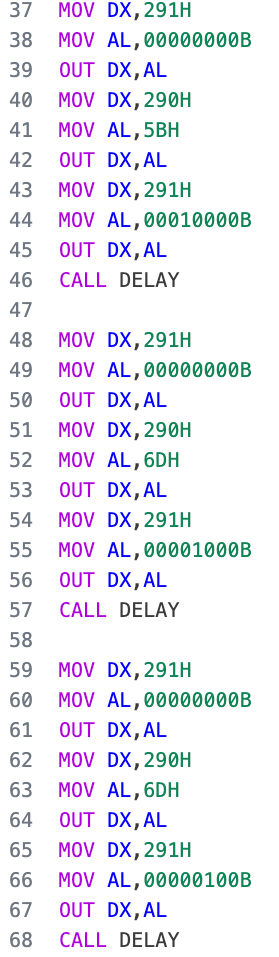
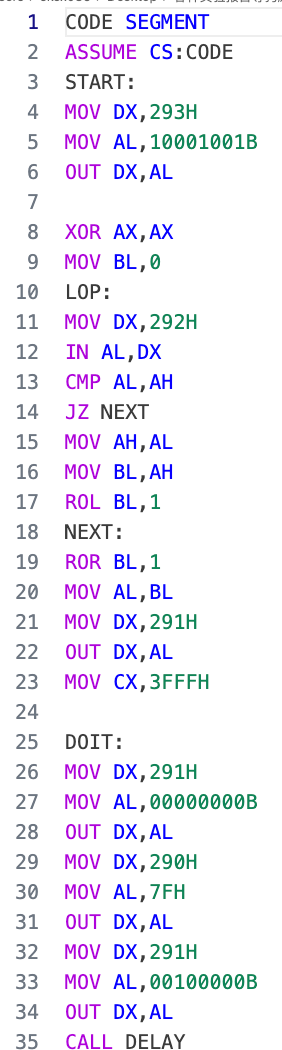
通过修改这个值，可以控制LED流水灯的速度，使其亮灭的间隔时间更短或更长。

更改流水灯方向：  


修改ROR为ROL即可

原理：  
这段代码的主要目的是控制LED的流水灯效果。其中，`DISPLAY\_PATTERN` 过程通过 `DISPLAY\_LOOP` 循环来控制LED的亮灭，而 `ROR AL, 1H` 指令是将 `AL` 寄存器的内容向右循环右移1位。在这个特定的场景下，将 `ROR` 指令改为 `ROL` 指令，即将 `AL` 寄存器的内容向左循环左移1位，将改变LED流水灯的方向。

原来的 `ROR AL, 1H` 将 `AL` 寄存器的最低位（最右边的位）移到最高位（最左边的位），导致LED流水灯向右移动。而如果将 `ROR` 改为 `ROL`，则将 `AL` 寄存器的最高位（最左边的位）移到最低位（最右边的位），导致LED流水灯向左移动。所以，修改 `ROR` 为 `ROL` 会改变流水灯的方向。



1. A显示代码

原理几乎同LED流水灯，不予阐述

1. 实验步骤
   1. 根据原理图正确连接实验线路（需要连接红线）。
   2. 正确理解实验原理。
   3. 编写实验程序，并上机调试，观察实验结果。

请见上实验报告

1. 思考题

在本实验的硬件电路中，能使用C口对LED指示灯控制码？如果可以写出2种控制方法（编程方法）。

在8255芯片中，A、B、C三个端口分别对应8255的端口A、端口B、端口C。对于控制LED指示灯，我们可以将数据发送到C端口。

以下是两种不同的编程方法，可以通过C口控制LED指示灯的控制码：

方法一：

在原始代码中，将数据发送到C端口（端口C的地址是 282H）：

MOV DX, 282H

MOV AL, 01H

OUT DX, AL

在这种方法中，将控制码数据直接发送到C端口。

方法二：

如果需要更多的控制，可以将控制码数据合并到C端口的原有数据中。在这种方法中，我们首先从C端口读取数据，然后修改其中的特定位，最后将修改后的数据发送回C端口。

MOV DX, 282H

IN AL, DX

OR AL, 01H

OUT DX, AL

在这种方法中，我们通过OR操作将特定位设置为1，保持其他位不变。这种方式可以实现更加灵活的控制，可以修改多个位的状态。

*注意0：要结合学生实验指导书pdf文档来理解本实验。*

*注意1：8个数码管分别对应S0-S7等8个位码插孔。每个位码插孔接5V高电平时，则其对应的数码管被选中；接0V低电平时，则其对应的数码管未被选中。 当某个位码插孔接5V高电平时，则8bit的段码(a-dp)上的内容就会被输出到该位码插孔对应的数码管上显示出来。*

*提示：利用视觉驻留，进行多位的数码管显示。*