

# 实验二 8255A 并行接口应用

## 一、实验目的

1. 掌握 8255A 的功能及方式 0、1 的实现
2. 熟悉 8255A 与 CPU 的接口，以及传输数据的工作原理及编程方法。
3. 了解七段数码管显示数字的原理。
4. 掌握同时显示多位数字的技术。

## 二、8255 应用小结

### 1. 8255 的工作方式

一片 8255 内部有 3 个端口，A 口可以工作在方式 0、方式 1 或方式 2，B 口可以工作在方式 0、方式 1，C 口可以工作在方式 0。

方式 0 是基本型输入/输出。这种方式和外设交换数据时，8255 端口与外设之间不使用联络线。

方式 1 为选通型输入/输出。用这种方式和外设交换数据时，端口和外设之间要有联络信号。

方式 2 是双向数据传送，仅 A 口有这项功能。当 A 口工作在方式 2 时，B 口仍可以工作在方式 0 或方式 1，但此时 B 口方式 1 只能用查询方式与 CPU 交换信息。

### 2. 工作方式选择字

8255 工作方式选择字共 8 位（如图），存放在 8255 控制寄存器中。最高位 D7 为标志位，D7=1 表示控制寄存器中存放的是工作方式选择字，D7=0 表示控制寄存器中存放的是 C 口置位/复位控制字。

	D7	D6 D5	D4	D3	D2	D1	D0
		00-方式 0	A 口	C 口高四位	0-方式 0	B 口	C 口低四位
1		01-方式 1	1: 输入	1: 入	1-方式 1	1: 入	1: 入
		1x-方式 2	0: 输出	0: 出		0: 出	0: 出

标志位

A 组

B 组

### 3. C 口置/复位控制字

8255 的 C 口可进行位操作，即：可对 8255C 口的每一位进行置位或清零操作，该操作是通过设置 C 口置/复位字实现的（图 8-10）。C 口置/复位字共 8 位，各位含义如下：

D7	D6D5D4	D3D2D1	D0
0		000 : PC0 001 : PC1 ..... 111 : PC7	1 : 置位 0 : 复位
标志位	x x x		

### 3. 8255A 的控制信号与传输动作的对应关系

A1	A0	/RD	/WR	/CS	工作状态
0	0	0	1	0	A 口数据→数据总线
0	1	0	1	0	B 口数据→数据总线
1	0	0	1	0	C 口数据→数据总线
0	0	1	0	0	数据总线→A 口数据
0	1	1	0	0	数据总线→B 口数据
1	0	1	0	0	数据总线→C 口数据
1	1	1	0	0	数据总线→控制寄存器
X	X	X	X	1	数据总线→三态
1	1	0	1	0	非法状态
X	X	1	1	0	数据总线→三态

### 4. 命令字与初始化编程

8255 有两个命令字，即方式选择控制字和 C 口置 0/置 1 控制字，初始化编程的步骤是：

① 向 8255 控制寄存器写入“方式选择控制字”，从而预置端口的工作方式。

②当端口预置为方式 1 或方式 2 时，再向控制寄存器写入“C 口置 0/置 1 控制字”。这一操作的主要目的是使相应端口的中断允许触发器置 0，从而禁止中断，或者使相应端口的中断允许触发器置 1，从而允许端口提出中断请求。

注意：“C 口置 0/置 1 控制字”虽然是对 C 口进行操作，但是该控制字是命令字，所以要写入控制寄存器，而不是写入 C 口控制寄存器。

③ 向 8255 数据寄存器写入“数据”或从 8255 数据寄存器读出“数据”

### 三、实验内容：

在实验一的基础上学习 PIO 芯片（8255）编程应用，熟悉平台的主要内容。CS 用 Y<sub>0</sub> (EE00H)

#### （一）简要说明：

在方式 0（输入/输出）下，以 A 口为输出口，B 口为输出口，A 口接六个共阴极数码管的八位段码，高电平点亮数码管的某一段，B 口接数码管的位选（即要使哪个数码管亮），高电平选中某一位数码管点亮。

8255A 中	A 端口地址	EE00H
	B 端口地址	EE01H

C 端口地址      EE02H  
控制地址        EE03H

八段数码管的显示规律及数码管的位选规律自己查找，可用实验一中，学过的 I、O 命令来做。

### （二）6 位数码管静态显示

在数码管电路上静态地显示 6 位学号，当主机键盘按下任意键时，停止显示，返回 DOS。

提示：该电路 6 个数码管的同名阳极段已经复接，当段选寄存器寄存了一个字型编码之后，6 个数码管都有可能显示出相同的数字。如果要使 6 个数码管“同时”显示不同的数字，必须采用扫描显示的方法，通过选位寄存器选择某一位数码管，显示其数字（对应段值为 1），然后关闭此数码管，再选择下一位数码管进行显示；如果在一秒钟内，每一位数码管都能显示 30 次以上，则人眼看到的是几位数码管同时在显示。

实验证明，在扫描显示过程中，每一位显示延迟 1ms 是最佳选择。

### （三）6 位数码管动态显示

要求在数码管电路 1—6 位数码管上按图 3.2 所示的规律，动态显示字符串 HELLO，当主机键盘按下任意键时结束。

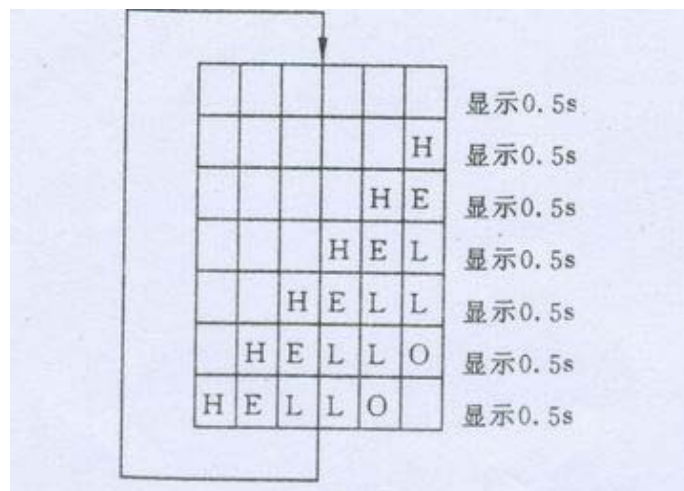


图 3.2 字符串动态显示示意图

提示：

① 本实验应在上面实验的基础上完成

②在数据段，按下列规律设置 12 个字型码：

MESG	DB	0, 0, 0, 0, 0, 0, 3DH, 0DCH, 8CH, 8CH, 0EDH, 0
POINT	DW	MESG
	...	...

POINT 单元存放 MESG 单元的有效地址，程序取出 POINT 单元的内容→BX，然后用 BX 间址取数送数码管电路，扫描显示 6 个字符。每过 0.5s 将 POINT 单元的内容加 1，再将 POINT 单元的内容→BX，……。POINT 单元内容加 1，使字符串显示的首地址向高地址移动一个单元，从而使 6 位字符串向“左”移动一位，实现动态显示。

③动态显示的速度可控制（快或慢），利用实验一读入端口的功能

（四）扩展部分：完成一个扫描键盘（PC 口）输入自己的学号，并在数码管上显示

端口 C 地址 EE02H      PC4、PC5、PC6、PC7 接行  
                                  PC3、PC2、PC1、PC0 接列  
 控制端口地址 EE03H

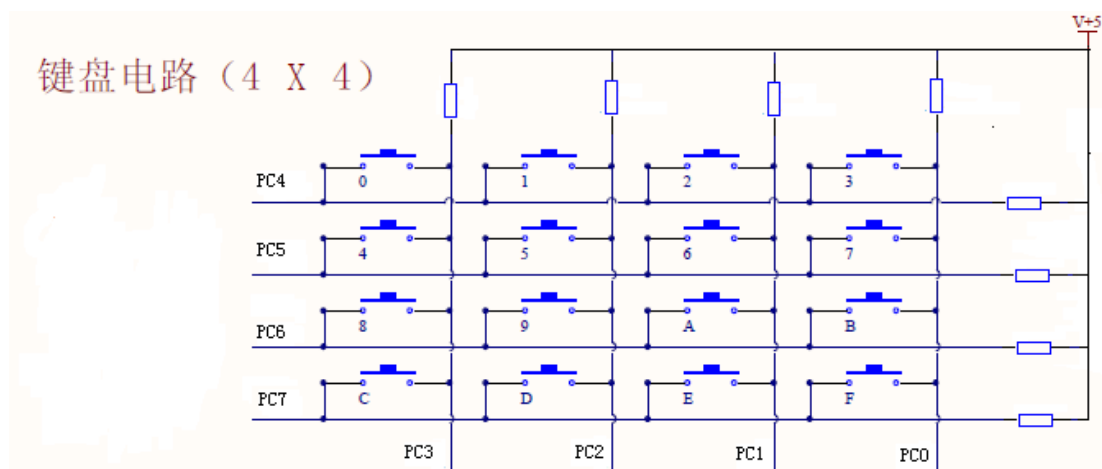


图 (二)

编程提示:

#### 1. 识别闭合键的方法

行扫描法是使键盘某一行线为低电平，其余为高电平，然后读取列值，如列值中有某位低电平，则表明行列交叉点处的键被按下；否则扫描下行，知道扫完全部的行线为止。

行反转法识别闭合键时，要将行线接一个并行口，先让他工作在输出方式，将列线接另一个并行口，先让它工作在输入方式，程序使 CPU 通过输出端口往各行线上全部送低电平，然后读入列线的值，如此时某键按下则必使某一列线值为 0，然后程序再对两个并行端口的工作方式进行设置，使行线工作在输入方式，列线工作在输出方式，并且将刚才读得的列线值从列线所接的并行端口输出，再读取行线上的输入值，那么在闭合键所在的行线上的值必定为 0。这样，有按键时，可读得唯一的行值和列值。

2. 设计时，行、列值应放在一个表中，通过查表确定按下的是哪个键。

3. 键闭合时，注意加延时防抖动。

4. 键松开时，加判断释放程序。

5. 按下键盘某键能够返回系统。

## 四、实验报告

1. 出程序流程图，打印程序清单，并加注释。

2. 实验结果讨论

3. 实验心得与收获建议