 信息工程学院实验报告

|  |
| --- |
| 成 绩： |
| 指导老师 ( 签名 ) ： |
|  |

课程名称：微机原理与接口技术

实验项目名称：键盘扫描及显示实验 实验时间： 2017.1.6

班级： 姓名： 学号：

一、实 验 目 的

1. 掌握 8254 的工作方式及应用编程。

2. 掌握 8254 典型应用电路的接法。

二、实 验 设 备

了解键盘扫描及数码显示的基本原理，熟悉 8255 的编程。

3、 实 验 原 理

将 8255 单元与键盘及数码管显示单元连接，编写实验程序，扫描键盘输入，并将扫描结果送数码管显示。键盘采用 4 × 4 键盘，每个数码管显示值可为 0 ～ F 共 16 个数。实验具体内容如下：将键盘进行编号，记作 0 ～ F ，当按下其中一个按键时，将该按键对应的编号在一个数码管上显示出来，当再按下一个按键时，便将这个按键的编号在下一个数码管上显示出来，数码管上可以显示最近 6 次按下的按键编号。

键盘及数码管显示单元电路图如图 7-1 和 7-2 所示。 8255 键盘及显示实验参考接线图如图 7-3 所示。

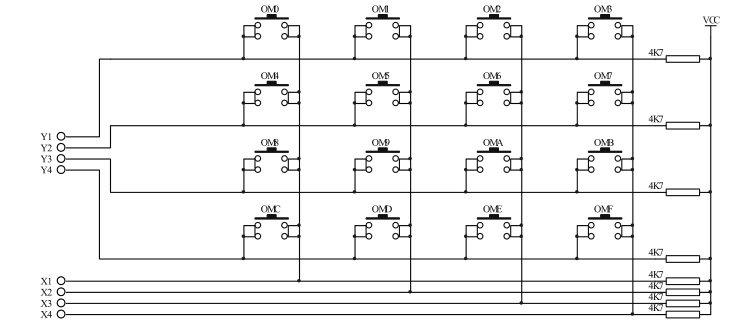


图 7-1 键盘及数码管显示单元 4 × 4 键盘矩阵电路图

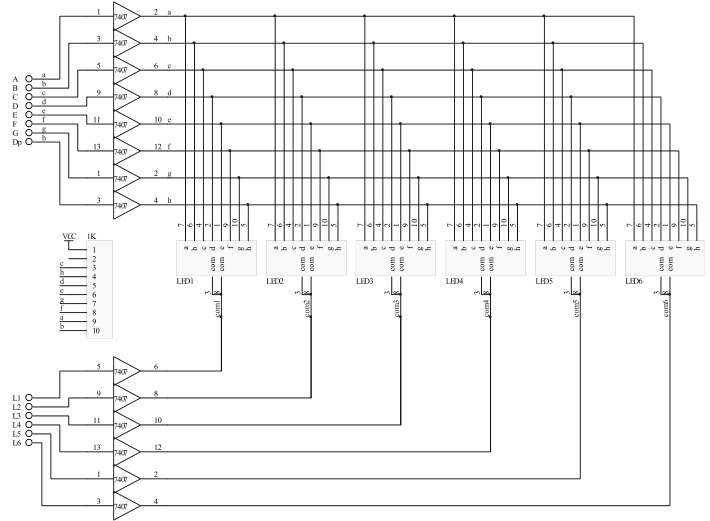


图 7-2 键盘及数码管显示单元 6 组数码管电路图

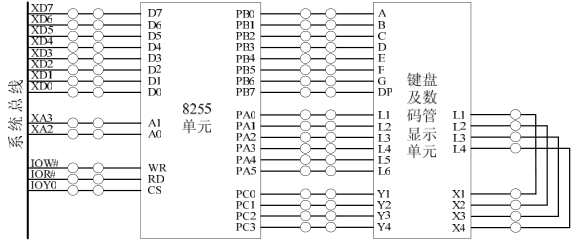


图 7-3 8255 键盘扫描及数码管显示实验线路图

四、实 验 内 容 与 步 骤

1. 实验接线图如图 7-3 所示，按图连接实验线路图。

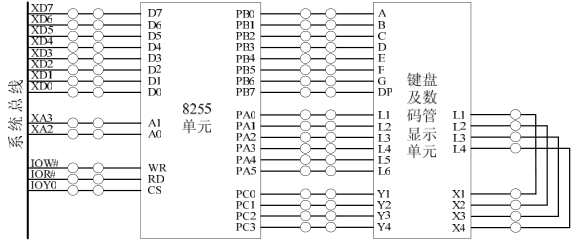


图 7-4 8255 键盘扫描及数码管显示实验实物连接图

2. 运行 Tdpit 集成操作软件，根据实验内容，编写实验程序，编译、链接。

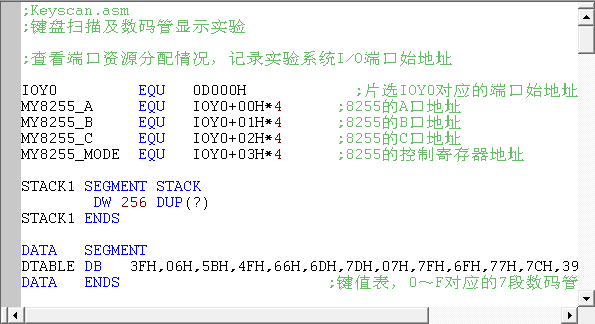


图 7-5 8255 键盘扫描及数码管显示实验程序编辑界面

3. 运行程序，按下按键，观察数码管的显示，验证程序功能。

五、实 验 结 果 及 分 析：

1. 运行程序，按下按键，观察数码管的显示 。

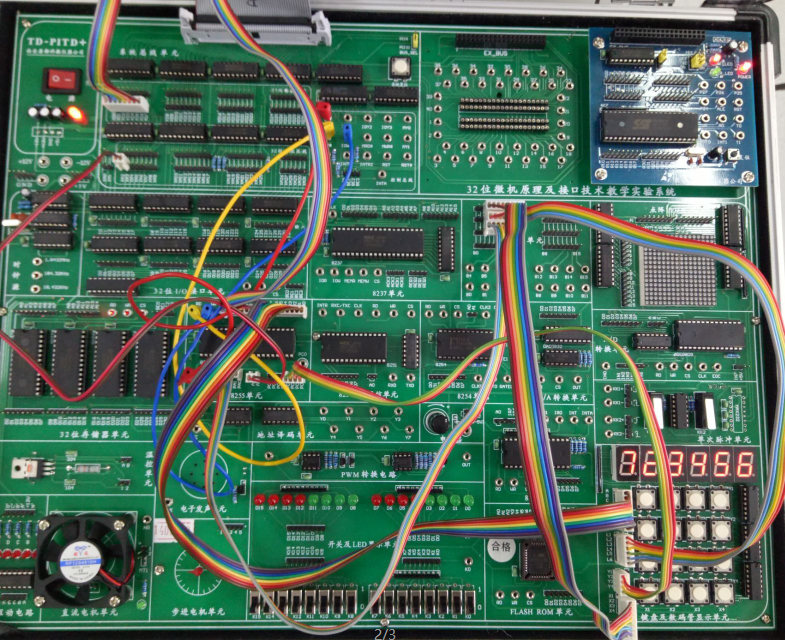


图 7-6 实验结果

分析： 当按下键盘其中一个按键时，将该按键对应的编号在一个数码管上显示出来，当再按下一个按键时，便将这个按键的编号在下一个数码管上显示出来，数码管上可以显示最近 6 次按下的按键编号。

六、实 验 总 结：

通过本实验，我了解键盘扫描及数码显示的基本原理，以及熟悉 8255 的编程。通过本次实验加深了对理论编程的知识的理解，巩固了我的理论知识。

附 录：

;Keyscan.asm

; 键盘扫描及数码管显示实验

; 查看端口资源分配情况，记录实验系统 I/O 端口始地址

IOY0 EQU 0D000H ; 片选 IOY0 对应的端口始地址

MY8255\_A EQU IOY0+00H\*4 ;8255 的 A 口地址

MY8255\_B EQU IOY0+01H\*4 ;8255 的 B 口地址

MY8255\_C EQU IOY0+02H\*4 ;8255 的 C 口地址

MY8255\_MODE EQU IOY0+03H\*4 ;8255 的控制寄存器地址

STACK1 SEGMENT STACK

DW 256 DUP(?)

STACK1 ENDS

DATA SEGMENT

DTABLE DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH,77H,7CH,39H,5EH,79H,71H

DATA ENDS ; 键值表， 0 ～ F 对应的 7 段数码管的段位值

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV SI,3000H ; 建立缓冲区，存放要显示的键值

MOV AL,00H ; 先初始化键值为 0

MOV [SI],AL

MOV [SI+1],AL

MOV [SI+2],AL

MOV [SI+3],AL

MOV [SI+4],AL

MOV [SI+5],AL

MOV DI,3005H

MOV DX,MY8255\_MODE ; 初始化 8255 工作方式

MOV AL,81H ; 方式 0 ， A 口、 B 口输出， C 口低 4 位输入

OUT DX,AL

MOV CX,0040h

BEGIN1: PUSH CX

BEGIN:

CALL DIS ; 显示刷新

CALL CLEAR ; 清屏

CALL CCSCAN ; 扫描按键

JNZ GETKEY1 ; 有键按下则跳置 GETKEY1

POP CX

LOOP BEGIN1

MOV CX,0040h

MOV AH,01h ; 判断 PC 键盘是否有按键按下

INT 16H

JZ BEGIN1 ; 无按键则跳回继续循环，有则退出

QUIT: MOV AX,4C00H ; 返回到 DOS

INT 21H

GETKEY1:CALL DIS ; 显示刷新

CALL DALLY

CALL DALLY

CALL CLEAR ; 清屏

CALL CCSCAN ; 再次扫描按键

JNZ GETKEY2 ; 有键按下则跳置 GETKEY2

JMP BEGIN ; 否则跳回开始继续循环

GETKEY2:MOV CH,0FEH

MOV CL,00H ; 设置当前检测的是第几列

COLUM: MOV AL,CH ; 选取一列，将 X1 ～ X4 中一个置 0

MOV DX,MY8255\_A

OUT DX,AL

MOV DX,MY8255\_C ; 读 Y1 ～ Y4 ，用于判断是哪一行按键闭合

IN AL,DX

L1: TEST AL,01H ; 是否为第 1 行

JNZ L2 ; 不是则继续判断

MOV AL,00H ; 设置第 1 行第 1 列的对应的键值

JMP KCODE

L2: TEST AL,02H ; 是否为第 2 行

JNZ L3 ; 不是则继续判断

MOV AL,04H ; 设置第 2 行第 1 列的对应的键值

JMP KCODE

L3: TEST AL,04H ; 是否为第 3 行

JNZ L4 ; 不是则继续判断

MOV AL,08H ; 设置第 3 行第 1 列的对应的键值

JMP KCODE

L4: TEST AL,08H ; 是否为第 4 行

JNZ NEXT ; 不是则继续判断

MOV AL,0CH ; 设置第 4 行第 1 列的对应的键值

KCODE: ADD AL,CL ; 将第 1 列的值加上当前列数，确定按键值

CALL PUTBUF ; 保存按键值

PUSH AX

KON: CALL DIS ; 显示刷新

CALL CLEAR ; 清屏

CALL CCSCAN ; 扫描按键，判断按键是否弹起

JNZ KON ; 未弹起则继续循环等待弹起

POP AX

NEXT: INC CL ; 当前检测的列数递增

MOV AL,CH

TEST AL,08H ; 检测是否扫描到第 4 列

JZ KERR ; 是则跳回到开始处

ROL AL,1 ; 没检测到第 4 列则准备检测下一列

MOV CH,AL

JMP COLUM

KERR: JMP BEGIN

CCSCAN PROC NEAR ; 扫描是否有按键闭合子程序

MOV AL,00H

MOV DX,MY8255\_A ; 将 4 列全选通， X1 ～ X4 置 0

OUT DX,AL

MOV DX,MY8255\_C

IN AL,DX ; 读 Y1 ～ Y4

NOT AL

AND AL,0FH ; 取出 Y1 ～ Y4 的反值

RET

CCSCAN ENDP

CLEAR PROC NEAR ; 清除数码管显示子程序

MOV DX,MY8255\_B ; 段位置 0 即可清除数码管显示

MOV AL,00H

OUT DX,AL

RET

CLEAR ENDP

DIS PROC NEAR ; 显示键值子程序

PUSH AX ; 以缓冲区存放的键值为键值表偏移找到键值并显示

MOV SI,3000H

MOV DL,0DFH

MOV AL,DL

AGAIN: PUSH DX

MOV DX,MY8255\_A

OUT DX,AL ; 设置 X1 ～ X4 ，选通一个数码管

MOV AL,[SI] ; 取出缓冲区中存放键值

MOV BX,OFFSET DTABLE

AND AX,00FFH

ADD BX,AX

MOV AL,[BX] ; 将键值作为偏移和键值基地址相加得到相应的键值

MOV DX,MY8255\_B

OUT DX,AL ; 写入数码管 A ～ Dp

CALL DALLY

INC SI ; 取下一个键值

POP DX

MOV AL,DL

TEST AL,01H ; 判断是否显示完？

JZ OUT1 ; 显示完，返回

ROR AL,1

MOV DL,AL

JMP AGAIN ; 未显示完，跳回继续

OUT1: POP AX

RET

DIS ENDP

PUTBUF PROC NEAR ; 保存键值子程序

MOV SI,DI

MOV [SI],AL

DEC DI

CMP DI,2FFFH

JNZ GOBACK

MOV DI,3005H

GOBACK: RET

PUTBUF ENDP

DALLY PROC NEAR ; 软件延时子程序

PUSH CX

MOV CX,0fffH

D1: MOV AX,00FFH

D2: DEC AX

JNZ D2

LOOP D1

POP CX

RET

DALLY ENDP

CODE ENDS

END START