## 《微机系统与接口实验》报告

|  |  |
| --- | --- |
| **实验名称：** | 键盘扫描及数码管显示实验 |
| **实验目的：** | 1)学习了解键盘扫描的基本原理。  2)掌握通过并行接口芯片对键盘进行扫描并在多位数码管显示数字的方法。 |
| 实验内容： | （1）基础实验：  编写程序，实现如下功能：初始时数码管无显示；第一次按下键盘时，在最右侧数码管显示对应的十六进制数字；以后每次按下键盘，则将当前显示的数字全部向左移动一位（最左侧的数字移出数码管），并将刚刚键入的数字显示在数码管的最右侧。  （2）扩展实验：  编写程序，将所按键对应的数字（0~E）按要求显示在指定的数码管上：  初始时，6位LED数码管上无显示；  第一次键入的数字显示在最右端的数码管上；之后每次键入新数字，其显示位置向左移动一位；（只显示最后键入的数字）  若当前显示的数字已处于数码管的最左端，此时再键入数字，则将其显示在最左端的数码管上；之后键入数字的显示位置逐次向右移动；  如此往复，直至按下数字键F，程序退出。 |
| **程序框图：** | |
| **程序代码：**  MY8255\_A EQU 0600H  MY8255\_B EQU 0602H  MY8255\_C EQU 0604H  MY8255\_CON EQU 0606H  SSTACK SEGMENT STACK  DW 16 DUP(?)  SSTACK ENDS  DATA SEGMENT  DTABLE DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H  DB 7FH,6FH,77H,7CH,39H,5EH,79H,00H    FLAG DB 00H  XX DW 00H  NOW DW 3000H  DATA ENDS  CODE SEGMENT  ASSUME CS:CODE,DS:DATA  START: MOV AX,DATA  MOV DS,AX  MOV SI,3000H  MOV AL,0FH  MOV [SI],AL ;清显示缓冲  MOV [SI+1],AL  MOV [SI+2],AL  MOV [SI+3],AL  MOV [SI+4],AL  MOV [SI+5],AL  MOV DI,3000H  MOV DX,MY8255\_CON ;写8255控制字  MOV AL,81H  OUT DX,AL  BEGIN: CALL DIS ;调用显示子程序  CALL CLEAR ;清屏  CALL CCSCAN ;扫描  JNZ INK1  JMP BEGIN  INK1: CALL DIS  CALL DALLY  CALL DALLY  CALL CLEAR  CALL CCSCAN  JNZ INK2 ;有键按下，转到INK2  JMP BEGIN  ;========================================  ;确定按下键的位置  ;========================================  INK2: MOV CH,0FEH  MOV CL,00H  COLUM: MOV AL,CH  MOV DX,MY8255\_A  OUT DX,AL  MOV DX,MY8255\_C  IN AL,DX  L1: TEST AL,01H ;is L1?  JNZ L2  MOV AL,00H ;L1  JMP KCODE  L2: TEST AL,02H ;is L2?  JNZ L3  MOV AL,04H ;L2  JMP KCODE  L3: TEST AL,04H ;is L3?  JNZ L4  MOV AL,08H ;L3  JMP KCODE  L4: TEST AL,08H ;is L4?  JNZ NEXT  MOV AL,0CH ;L4  KCODE: ADD AL,CL  CALL PUTBUF  PUSH AX  KON: CALL DIS  CALL CLEAR  CALL CCSCAN  JNZ KON  POP AX  NEXT: INC CL  MOV AL,CH  TEST AL,08H  JZ KERR  ROL AL,1  MOV CH,AL  JMP COLUM  KERR: JMP BEGIN  ;========================================  ;键盘扫描子程序  ;========================================  CCSCAN: MOV AL,00H  MOV DX,MY8255\_A  OUT DX,AL  MOV DX,MY8255\_C  IN AL,DX  NOT AL  AND AL,0FH  RET  ;========================================  ;清屏子程序  ;========================================  CLEAR: MOV DX,MY8255\_B  MOV AL,00H  OUT DX,AL  RET  ;========================================  ;显示子程序  ;========================================  DIS: PUSH AX  MOV SI,NOW  MOV DL,0DFH  MOV AL,DL  AGAIN: PUSH DX  MOV DX,MY8255\_A  MOV XX,DI  SUB XX,3000H  NEXT2:  CMP XX,00H  JE NEXT1  ROR AL,1  DEC XX  JMP NEXT2  NEXT1:  OUT DX,AL  MOV AL,[SI]  MOV BX,OFFSET DTABLE  AND AX,00FFH  ADD BX,AX  MOV AL,[BX]  MOV DX,MY8255\_B  OUT DX,AL    CALL DALLY  POP DX      OUT1: POP AX  RET  ;====== 延时子程序 ======  DALLY: PUSH CX  MOV CX,0006H  T1: MOV AX,009FH  T2: DEC AX  JNZ T2  LOOP T1  POP CX  RET  ;========================================  ;存键盘值到相应位的缓冲中  ;========================================  PUTBUF:  CMP FLAG,1  JE RIGHT  LEFT:  MOV SI,DI  MOV [SI],AL  MOV NOW,DI  INC DI  CMP DI,3006H  JNZ GOBACK  MOV DI,3005H  MOV FLAG,1  JMP GOBACK  RIGHT:  MOV SI,DI  MOV [SI],AL  MOV NOW,DI  DEC DI  CMP DI,2FFFH  JNZ GOBACK  MOV DI,3000H  MOV FLAG,0  JMP GOBACK  GOBACK: RET  CODE ENDS  END START | |
| **实验结果分析与体会：**  这次实验对我来说是个不小的挑战。一开始，我对汇编语言的使用还不太熟练，加上硬件控制方面的知识也有限，所以在实验的前期，我真的遇到了不少困难。  首先，我在实现键盘扫描的时候遇到了麻烦。理解如何通过MY8255芯片读取键盘输入的逻辑花了我不少时间。有几次我按下键盘时，数码管上完全没有反应，或者显示的数字完全不对。这让我有点沮丧，但也激发了我的好奇心。我开始反复检查代码，查找资料，试图弄清楚问题出在哪里。经过一番努力，我终于理解了如何正确设置控制字和端口，以及如何处理从键盘获取的信号。  数码管显示部分也让我头疼了一阵子。我最初没能正确实现数字的移动效果。数字要么一直卡在第一位，要么移动得太快，甚至有时候程序就直接崩溃了。我意识到这和缓冲区的管理有关，于是开始仔细分析缓冲区的操作逻辑。通过一些试错和调整，我最终让数码管上的数字能够按照预期左移或右移。  通过这次实验，我不仅学到了很多关于汇编语言和硬件控制的知识，还锻炼了我的问题解决能力。每次解决一个问题，看到数码管上正确显示结果的时候，那种成就感真的无法用言语表达。我也认识到，在编程和硬件操作方面，还有很多知识需要我去学习和掌握。 | |