ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC HÀNH MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN BÀI THỰC HÀNH SỐ 6: SỬ DỤNG UART

Sinh viên thực hiện:

Trần Ngọc Ánh

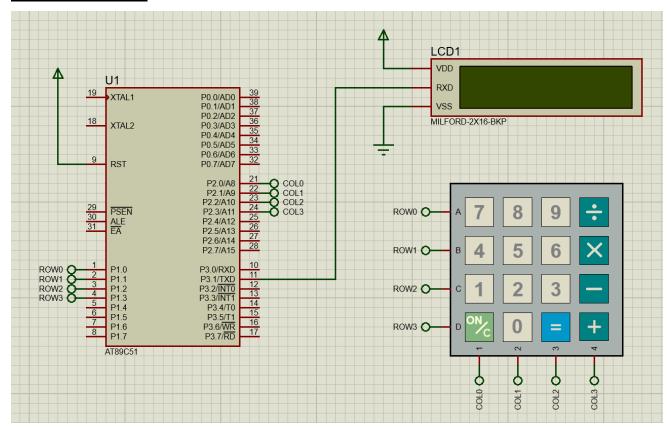
22520077

Giảng viên hướng dẫn: Phạm Minh Quân Mã lớp: CE103.O22

TP. HÒ CHÍ MINH, 30 THÁNG 5 NĂM 2024

Yêu cầu thực hành:

- 1. Xây dựng bộ bàn phim 4x4 gồm các nút sau:
 - o Từ 0 → 9
 - Các dấu +, -, *, /
 - o Dấu =
 - Nút Reset
- 2. Sử dụng AT89C51/AT89C52 kết hợp với module vừa thiết kế trên để thiết kế một máy tính cầm tay, hiển thị phép tính và kết quả trên 1 LCD nhận dữ liệu qua UART.
 - a) Mô phỏng thiết kế:



b) Source code và giải thích nguyên lí:

Code	Nguyên lí hoạt động
N1 EQU 30H	Địa chỉ lưu toán hạng đầu tiên
N2 EQU 31H	Địa chỉ lưu toán hạng thứ hai
OP EQU 32H	Địa chỉ lưu dấu
R EQU 33H	Địa chỉ lưu kết quả
SIGN EQU 34H	Biến SIGN để xác định số xuất ra là âm hay dương
TEMP EQU 35H	

```
DIF BIT 0AH
                                Thể hiện số
                                Thể hiện các ký tự +, -, *, /
OIF BIT 0BH
                                Thể hiện dấu "="
AIF BIT 0CH
Space DATA 32d
; Reset Vector
org 0000h
jmp Start
; Code Segment
; Main Program
org 0100h
Start:
   MOV TMOD, #021h
   LCALL LCDInit
   MOV R5, Space
Loop:
   LCALL KTdchnho
   LCALL Nhan
   LCALL Tinhtoan
   LCALL IN
   JMP Loop
LCDInit:
   MOV TH1, #0FDh
   MOV SCON, #50h
   SETB TR1
   RET
KTdchnho:
                                Hàm KTdchnho đảm bảo các thanh ghi đang lưu kết
   CLR A
                                quả là 0.
   MOV N1,A
   MOV N2,A
   MOV OP,A
   MOV R,A
   MOV SIGN, A
   MOV TEMP, A
   CLR C
   MOV DIF, C
   MOV OIF, C
```

```
MOV AIF, C
   RET
Nhan:
                                  Hàm Nhan nhận tín hiệu của phép tính.
   ACALL docnutnhan
                                  Đọc nút nhấn.
   ACALL XL_input
                                  Xử lý các nhóm tín hiệu.
   ACALL LCD_hien
                                  Đọc toán hạng đầu tiên.
   DEC<sub>R5</sub>
   ANL A,#0FH
   MOV N1,A
                                  Đọc dấu của phép tính.
   ACALL docnutnhan
   ACALL XL_input
   ACALL LCD hien
   DEC<sub>R5</sub>
   MOV OP, A
                                  Đọc toán hạng thứ hai.
   ACALL docnutnhan
   ACALL XL input
   ACALL LCD_hien
   DEC<sub>R5</sub>
   ANL A,#0FH
   MOV N2,A
                                  Đọc dấu bằng.
   ACALL docnutnhan
   ACALL XL_input
   ACALL LCD_hien
   DEC<sub>R5</sub>
   RET
docnutnhan:
   MOV P1,#0FH
K1:
   MOV P2,#0
   MOV A,P1
   ANL A,#00001111B
   CJNE A,#00001111B,K1
K2:
```

```
LCALL DELAY
   MOV A,P1
                              nhấn.
   ANL A,#00001111B
   CJNE A,#00001111B,Doco
   SJMP K2
Doco:
   MOV P2,#11111110B
   MOV A,P1
   ANL A,#00001111B
   CJNE A,#00001111B,ROW_0
   MOV P2,#11111101B
   MOV A,P1
   ANL A,#00001111B
   CJNE A,#00001111B,ROW 1
   MOV P2,#11111011B
   MOV A,P1
   ANL A,#00001111B
   CJNE A,#00001111B,ROW_2
   MOV P2,#11110111B
   MOV A,P1
   ANL A,#00001111B
   CJNE A,#00001111B,ROW_3
   LJMP K2
ROW 0:
   MOV DPTR, #KCODE0
   SJMP FIND
ROW_1:
   MOV DPTR, #KCODE1
   SJMP FIND
ROW 2:
   MOV DPTR, #KCODE2
   SJMP FIND
ROW_3:
   MOV DPTR, #KCODE3
FIND:
   RRC A
   JNC MATCH
```

Hàm *K2* giúp chống đợi phím khi nhấn. Hàm delay để tránh việc phím nhấn nhiều lần khi ta nhấn.

Dò theo từng cột để xác nhận cột đang được nhấn.

4 hàm *ROW_0*, *ROW_1*, *ROW_2*, *ROW_3* sẽ đại diện 4 hàng trỏ đến ký tự tương ứng.

```
INC DPTR
   SJMP FIND
MATCH:
   CLR A
   MOVC A, @A+DPTR
   JZ ON
   RET
ON:
   ACALL Xoa
   JMP Start
                                Hàm kiểm tra ký tự nhập vào thuộc nhóm nào trong
XL_input:
                                3 nhóm.
   CJNE A,#"+", next1
   AJMP found
next1:
   CJNE A,#"-", next2
   AJMP found
next2:
   CJNE A,#"*", next3
   AJMP found
next3:
   CJNE A,#"/", next4
   AJMP found
next4:
   CJNE A,#"=", next5
                                Nếu là dấu "=" thì sẽ nhảy đến hàm này, set DIF lên
   CLR DIF
                                 1.
   CLR OIF
   SETB AIF
   RET
next5:
   SETB DIF
   CLR OIF
   CLR AIF
   RET
found:
   CLR DIF
                                Nếu là dấu +, -, *, / thì sẽ nhảy đến hàm này, set
   SETB OIF
                                OIF lên 1.
   CLR AIF
```

```
RET
                                   Hàm tính toán các giá trị:
Tinhtoan:
                                   - Hai toán hạng sẽ được chuyển từ N1, N2 đến hai
   MOV A,N1
                                   thanh ghi A, B.
   MOV B, N2
                                   - Dấu sẽ chuyển từ OP sang R7.
   MOV R7, OP
   CJNE R7,#"+",NEXT11
                                   Phép cộng 2 số.
   ADD A,B
   MOV R,A
   MOV SIGN, #"+"
   RET
NEXT11:
                                   Phép trừ 2 số.
   CJNE R7, #"-", NEXT22
   SUBB A,B
                                   Nhảy đến Truam khi c = 1 (phép trừ trên ra kết quả
   JC Truam
                                   âm)
   MOV R,A
   MOV SIGN, #"+"
   RET
Truam:
                                   Xử lý trường hợp số bé trừ số lớn ra kết quả âm:
   CPL A
                                   - Tính ra số bù 1.
   INC A
                                   - Cộng 1 ra số bù 2.
   MOV R,A
   MOV SIGN, #"-"
   RET
NEXT22:
                                   Phép nhân 2 số.
   CJNE R7, #"*", NEXT33
   MUL AB
   MOV R,A
   MOV SIGN, #"+"
   RET
NEXT33:
                                   Phép chia 2 số.
   CJNE R7, #"/", NEXT44
   DIV AB
   MOV R,A
   MOV SIGN, #"+"
NEXT44:
   RET
```

IN:	Hàm <i>IN</i> in ra giá trị output.
MOV R6,SIGN	
CJNE R6,#"+",SINGED	
RETURN:	
MOV A, R	
MOV B,#0AH	
DIV AB	
JZ LESSTEN	
ADD A,#30h	
ACALL LCD_hien	
DEC R5	
MOV A,B	
ADD A,#30h	
ACALL LCD_hien	
DEC R5	
JMP DONE	
LESSTEN:	
MOV A,B	
ADD A,#30h	
ACALL LCD_hien	
DEC R5	
JMP DONE	To no dôn also turnim a la ons aô ôns
SINGED:	In ra dấu cho trường hợp số âm.
MOV A,#"-"	
ACALL LCD_hien	
DEC R5	
JMP RETURN	
DONE:	
RET	
Xoa:	Hàm Vog vớn hồng nính lớn lại hy ớn in nón
MOV A,#" "	Hàm <i>Xoa</i> – xóa bằng cách lặp lại bước in các khoảng trắng.
ResetVT:	knoding uding.
ACALL LCD_hien	
DJNZ R5, ResetVT	
MOV R5,Space	
Clear:	

```
ACALL LCD_hien
   DJNZ R5, Clear
   MOV R5, Space
   RET
                                Sử dụng UART để truyền tín hiệu xuất cho màn
LCD_hien:
                                hình.
   MOV SBUF, A
                                Gía trị cần xuất lưu trong thanh ghi A.
Wait:
                                Truyền giá trị cần xuất vào SBUF.
   JNB TI, Wait
   CLR TI
   RET
DELAY:
L00:
   MOV TL0, #LOW(-20000)
   MOV TH0, #HIGH(-20000)
   SETB TR0
LOOP2:
   JNB TF0, LOOP2
   CLR TR0
   CLR TF0
   RET
LDELAY:
LLOP:
   MOV TL0, #LOW(-10000)
   MOV TH0, #HIGH(-10000)
   SETB TR0
LLOOP2:
   JNB TF0, LLOOP2
   CLR TR0
   CLR TF0
   RET
   ORG 300H
                                Định nghĩa bảng ký tự:
                                - #!ROW0
KCODE0: DB "7","4","1",0
                                - #!ROW1
KCODE1: DB "8","5","2","0"
```

```
KCODE2: DB "9","6","3","=" -#!ROW2
KCODE3: DB "/","*","-","+" -#!ROW3

End
```

c) Kết quả thực thi:

Video mô phỏng lại quá trình mạch chạy trên phần mềm Proteus được đính kèm trong link: https://drive.google.com/file/d/1jY5cB-IbMcfIHCYD7upLdfdwcXFNXQyN/view?usp=sharing

