
THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Trần Hoàng Lộc

Họ và tên sinh viên thực hiện: Đoàn Vũ Phú Minh

Mã số sinh viên: 22520859

BÀI THỰC HÀNH 06: SỬ DỤNG UART

I. Báo cáo

Yêu cầu :

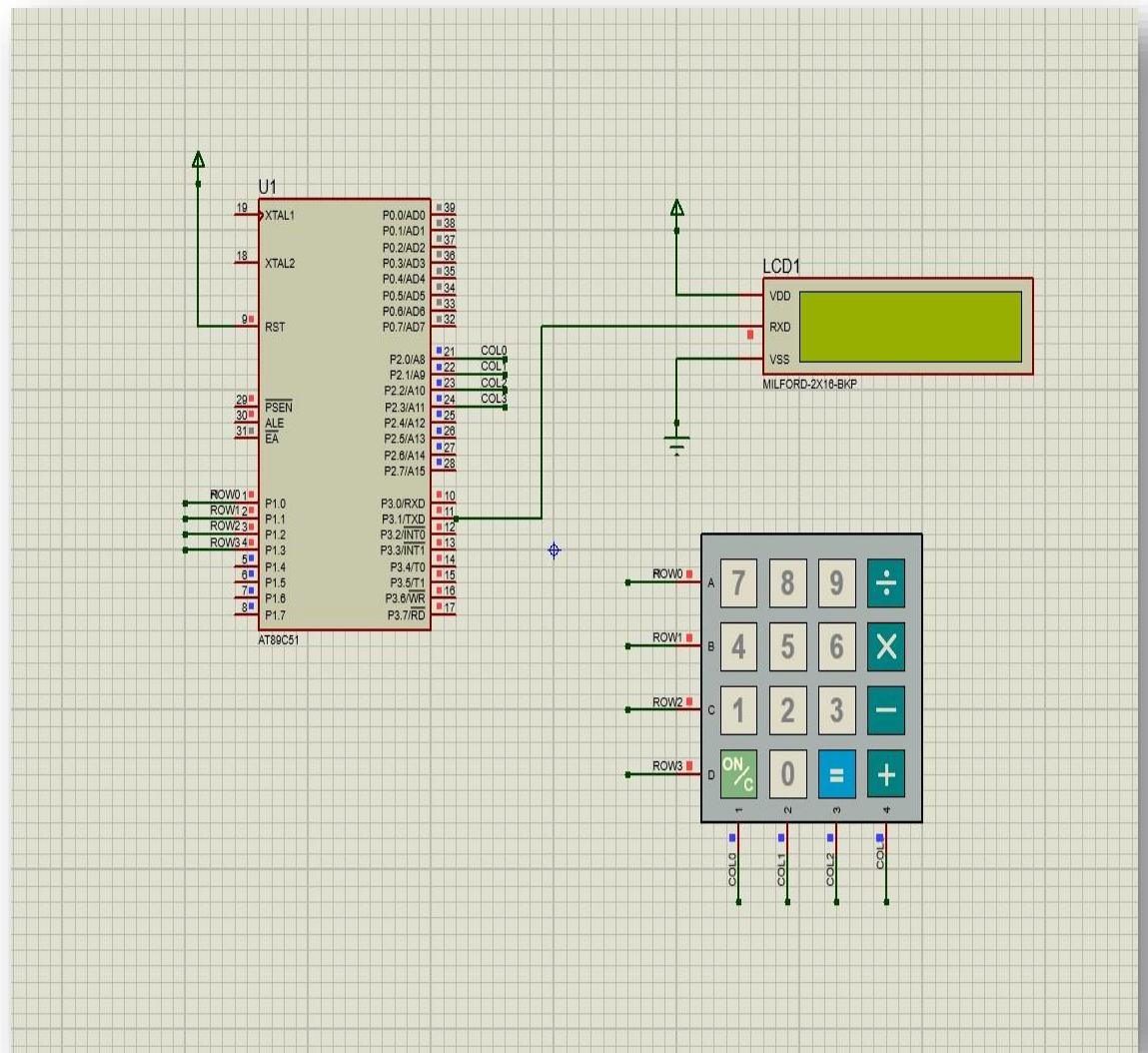
1. Design a 4x4 keyboard set including the following buttons:

+ From 0 to 9.

+ The signs + - * /.

+ Sign = o Reset button

Hình ảnh:



II. Giải thích

Yêu cầu :

Giải thích chi tiết SourceCode:

```

Number1 EQU 30H
Number2 EQU 31H
Operator EQU 32H
R EQU 33H
Sign EQU 34H
Temp EQU 35H
DIF BIT 0AH
OIF BIT 0BH
AIF BIT 0CH

```

```

; The hien la dau =
Space DATA 32d
; Reset Vector
org 0000h
jmp Start

```

-Khai báo các thanh ghi có địa chỉ cụ thể đối với từng tên biến để dễ thao tác và sử dụng:

- + **Number1, Number2:** tương ứng là số hạng thứ 1 và thứ 2
- + **Operator:** Là toán tử toán học (+,-*,/)
- + **R:** là Result kết quả
- + **Sign :** dùng để kiểm tra kết quả có phải là số âm không

+ **Temp:** Khai báo biến tạm

+ Các thanh ghi còn lại để hiện số, toán tử và dấu “=”.

```

; Ham chuong trinh chinh
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

```

- Khai báo thanh ghi TMOD sử dụng Timer1 chế độ 1 cho phép nạp lại 8 bit.
- Hàm KTdchnho: dùng để thực hiện kiểm tra các thanh ghi bằng 0.
- Hàm Nhan: thực hiện thao tác trên Keypad.
- Hàm Tinhtoan: Dùng để thực hiện tính toán trên các số để đưa vào Result -
- Hàm In: dùng để đưa ra giá trị của Output.
- Thực hiện gán các giá trị trên thanh ghi SCON và TH1 để xác định baudret.

```

; Dam bao cac thanh ghi dang luu ket qua la 0
<Tdchnho:
        CLR     A
        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

```

- Thực hiện xóa cờ A và di chuyển giá trị đó vào các số cần thiết để sau này nhập vào giá trị mới.

;Ham nhan tin hieu cua phep tinh

Nhan:

```
ACALL docnutnhan    ;Doc nut nhan
ACALL XL_input      ; Xu ly ra cac nhom tin hieu
```

; Doc toan hang dau tien

```
ACALL LCD_hien
DEC     R5
ANL     A,#0FH
MOV     N1,A
```

;Doc dau cua phep tinh

```
ACALL docnutnhan
ACALL XL_input
ACALL LCD_hien
DEC     R5
MOV     OP,A
```

;Doc Toan hang thu 2

```
ACALL docnutnhan
ACALL XL_input
ACALL LCD_hien
DEC     R5
ANL     A,#0FH
MOV     N2,A
```

; Doc dau bang

```
ACALL docnutnhan
ACALL XL_input
ACALL LCD_hien
DEC     R5
RET
```

- Thực hiện kiểm tra nút nhấn và xuất dữ liệu
- Thực hiện ANL bit với giá trị #0fh để xác định được nhận sau đó di chuyển giá trị đó vào từng Number tương ứng.

```

;=====
docnutnhan:
MOV     P1,#0FH
K1:
MOV     P2,#0
MOV     A,P1
ANL     A,#00001111B
CJNE    A,#00001111B,K1
; Ham giup chong doi phim khi nhan
K2:
LCALL   DELAY    ; Ham delay de tranh viec phim nhan nhieu l
MOV     A,P1
ANL     A,#00001111B
CJNE    A,#00001111B,Doco
SJMP    K2
; Do theo tung cot de xac cot dang dc nhan
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

```

-Thực hiện kiểm tra xác định nút nào được nhận ANL bit của Port P1 với #0fh để xác định nút nào được nhận . Nếu có nút thật sự được nhận thì sẽ nhảy đến hàm kiểm tra cột.

-Ta xác định từng cột 0 đến cột 3 để xác định nút được nhận. Khi xác định được cột thì ta nhảy đến dòng tương ứng với nó. Để thực hiện dùng con trỏ để nhảy tới vị trí xác định trong mạng.

```
;4 ham duoi se dai dien 4 hang tro den ky tu tuong ung
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
            INC      DPTR
            SJMP     FIND
MATCH:      CLR      A
            MOVC     A,@A+DPTR
            JZ        ON
            RET
ON:         ACALL    Xoa
            JMP      Start
```

```

144 ;=====
145 XL_input:
146 ; Kiem tra ky tu nhap vafo thuoc nhom nafo trong 3 nhom
147         CJNE    A,#"+", next1
148         AJMP     found
149 next1:
150         CJNE    A,#"-", next2
151         AJMP     found
152 next2:
153         CJNE    A,#"*", next3
154         AJMP     found
155 next3:
156         CJNE    A,#"/", next4
157         AJMP     found
158 next4:
159         CJNE    A,#"=", next5
160         ; Neu la dau = thi se nhay den ham nay, set DIF len 1
161         CLR      DIF
162         CLR      OIF
163         SETB     AIF
164         RET
165 next5:
166         SETB     DIF
167         CLR      OIF
168         CLR      AIF
169         RET
170 found:  ; Neu la dau +,-,*,/ thi se nhay den ham nay, set OIF len 1
171         CLR      DIF
172         SETB     OIF
173         CLR      AIF
174         RET
175 ;=====

```



```

;-----
; Tính toán các giá trị
TinhToan:
; 2 toán hạng sẽ được chuyển từ N1,N2 -> hai thanh ghi A, B
; Dấu sẽ chuyển từ OP sang R7
    MOV     A,N1
    MOV     B,N2
    MOV     R7,OP
;Phep cong 2 so
    CJNE    R7,"+",NEXT11
    ADD     A,B
    MOV     R,A
    MOV     SIGN,"+"
    RET
NEXT11:
;Phep tru 2 so
    CJNE    R7,"-",NEXT22
    SUBB    A,B
    JC      Truam                ; Nhảy khi C=1 (phep tru tren ra kq am)
    MOV     R,A
    MOV     SIGN,"+"
    RET
;Ham xu ly trah so be tru so lon => kq am
Truam:
    CPL     A                    ;Tinh ra so bu 1
    INC     A                    ;Cong 1 ra so bu 2
    MOV     R,A
    MOV     SIGN,"-"
    RET
; Phep nhan 2 so
NEXT22:
    CJNE    R7,"*",NEXT33
    MUL     AB
    MOV     R,A
    MOV     SIGN,"+"
    RET
;Phep chia 2 so
NEXT33:
    CJNE    R7,"/",NEXT44
    DIV     AB
    MOV     R,A
    MOV     SIGN,"+"
NEXT44: RET

```

-Thực hiện hàm tính toán các giá trị trong các thanh ghi tương ứng, cộng trừ nhân chia, và có trường hợp nếu như phép trừ ra âm chúng ta sẽ phải chuyển giá trị và đặt dấu “-“ tương ứng trước kết quả hiển thị.

```

215 NEX144: RET
216 ;=====
217 ; Ham in gia tri output
218 IN:
219         MOV     R6,SIGN
220         CJNE    R6,#"+",SINGED
221 RETURN:
222         MOV     A,R
223         MOV     B,#0AH
224         DIV     AB
225         JZ      LESSTEN
226         ADD     A,#30h
227         ACALL   LCD_hien
228         DEC     R5
229         MOV     A,B
230         ADD     A,#30h
231         ACALL   LCD_hien
232         DEC     R5
233         JMP     DONE
234 LESSTEN:
235         MOV     A,B
236         ADD     A,#30h
237         ACALL   LCD_hien
238         DEC     R5
239         JMP     DONE
240 SINGED: ;In ra dau cho truong hop so am
241         MOV     A,#"- "
242         ACALL   LCD_hien
243         DEC     R5
244         JMP     RETURN
245 DONE:
246         RET

```

```

247 ;=====
248 ;Ham Xoa
249 ;Xoa bang cach lap lai buoc in cac khoang trang
250 Xoa:
251         MOV     A,#" "
252 ResetVT:
253         ACALL   LCD_hien
254         DJNZ    R5,ResetVT
255         MOV     R5,Space
256 Clear:
257         ACALL   LCD_hien
258         DJNZ    R5,Clear
259         MOV     R5,Space
260         RET

```

```

1 ;=====
2 ; Su dung UART de truyen tin hieu xuat cho man hinh
3 ; Gia tri can xuat luu trong thanh ghi A
4 LCD_hien:
5         MOV     SBUF,A ; Truyen gia tri can xuat vao SBUF
6 Wait:
7         JNB     TI,Wait
8         CLR     TI
9         RET

```

Thực hiện hiển thị giá trị ra màn hình LCD di chuyển giá trị từ thanh ghi A vào sbuf sau đó chờ đến khi cờ TI bật lên kết thúc việc truyền sẽ thoát khỏi hàm con và sau đó tiếp tục nhận giá trị ở thanh ghi A để tiếp tục hiển thị.

Khi thực hiện đủ 4 số hoặc dấu hiển thị tức 1 cái là 8 bit như ta khai báo ở trên R5 tương ứng 32 bit cho dữ liệu.

```
REI
;=====
DELAY:
LOOP:
    MOV    TL0, #LOW(-20000)
    MOV    TH0, #HIGH(-20000)
    SETB   TR0
LOOP2:
    JNB     TF0, LOOP2
    CLR     TR0
    CLR     TF0
    RET
;=====
RET
;=====
ORG      300H

;Bảng ký tự
KCODE0: DB      "7","4","1",0           ;!ROW 0
KCODE1: DB      "8","5","2","0"         ;!ROW 1
KCODE2: DB      "9","6","3","="         ;!ROW 2
KCODE3: DB      "/", "*", "-", "+"      ;!ROW 3
```

-Thực hiện hàm delay bằng Timer 0 để delay thời gian nhấn phím

- Khai báo các mảng tương ứng để sử dụng con trỏ pointer trỏ vào đúng thứ tự phím nhấn.