

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

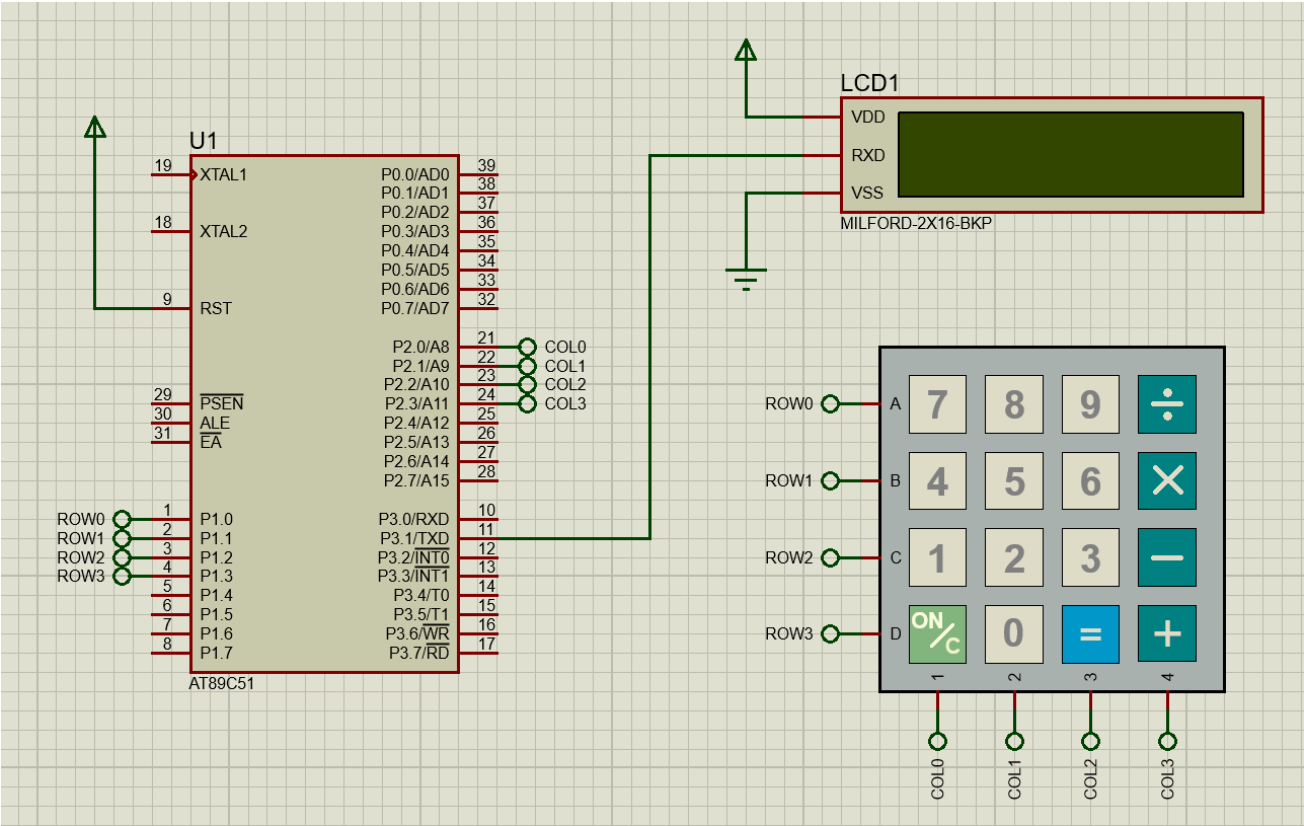
Mã lớp: CE103.O22

TP. HỒ CHÍ MINH, 30 THÁNG 5 NĂM 2024

Yêu cầu thực hành:

- 1. Xây dựng bộ bàn phím 4x4 gồm các nút sau:**
 - Từ 0 → 9
 - Các dấu +, -, *, /
 - 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	

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

<pre> LCALL DELAY MOV A,P1 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,#111111101B 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 </pre>	<p>Hàm <i>K2</i> giúp chống đợi phím khi nhấn. Hàm delay để tránh việc phím nhấn nhiều lần khi ta nhấn.</p> <p>Dò theo từng cột để xác nhận cột đang được nhấn.</p> <p>4 hàm <i>ROW_0</i>, <i>ROW_1</i>, <i>ROW_2</i>, <i>ROW_3</i> sẽ đại diện 4 hàng trở đến ký tự tương ứng.</p>
---	---

<pre> INC DPTR SJMP FIND MATCH: CLR A MOVC A,@A+DPTR JZ ON RET ON: ACALL Xoa JMP Start XL_input: 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 CLR DIF CLR OIF SETB AIF RET next5: SETB DIF CLR OIF CLR AIF RET found: CLR DIF SETB OIF CLR AIF </pre>	<p>Hàm kiểm tra ký tự nhập vào thuộc nhóm nào trong 3 nhóm.</p> <p>Nếu là dấu “=” thì sẽ nhảy đến hàm này, set DIF lên 1.</p> <p>Nếu là dấu +, -, *, / thì sẽ nhảy đến hàm này, set OIF lên 1.</p>
---	--

<pre> RET TinhToan: MOV A,N1 MOV B,N2 MOV R7,OP CJNE R7,#"+",NEXT11 ADD A,B MOV R,A MOV SIGN,#"+" RET NEXT11: CJNE R7,#"-",NEXT22 SUBB A,B JC Truam MOV R,A MOV SIGN,#"+" RET Truam: CPL A INC A MOV R,A MOV SIGN,#"- " RET NEXT22: CJNE R7,#"*",NEXT33 MUL AB MOV R,A MOV SIGN,#"+" RET NEXT33: CJNE R7,#"/",NEXT44 DIV AB MOV R,A MOV SIGN,#"+" RET NEXT44: RET </pre>	<p>Hàm tính toán các giá trị:</p> <ul style="list-style-type: none"> - Hai toán hạng sẽ được chuyển từ N1, N2 đến hai thanh ghi A, B. - Dấu sẽ chuyển từ OP sang R7. <p>Phép cộng 2 số.</p> <p>Phép trừ 2 số.</p> <p>Nhảy đến <i>Truam</i> khi c = 1 (phép trừ trên ra kết quả âm)</p> <p>Xử lý trường hợp số bé trừ số lớn ra kết quả âm:</p> <ul style="list-style-type: none"> - Tính ra số bù 1. - Cộng 1 ra số bù 2. <p>Phép nhân 2 số.</p> <p>Phép chia 2 số.</p>
--	---


```
ACALL LCD_hien
DJNZ R5,Clear
MOV R5,Space
RET
```

LCD_hien:

MOV SBUF,A

Wait:

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

LOOP2:

JNB TF0, LOOP2

CLR TR0

CLR TF0

RET

ORG 300H

```
KCODE0: DB "7","4","1",0
```

```
KCODE1: DB "8","5","2","0"
```

Sử dụng UART để truyền tín hiệu xuất cho màn hình.

Gía trị cần xuất lưu trong thanh ghi A.

Truyền giá trị cần xuất vào SBUF.

Định nghĩa bảng ký tự:

- #!ROW0

- #!ROW1

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>

