

TH Kiến Trúc Và Tổ Chức Máy Tính TS. Phạm Văn Khoa

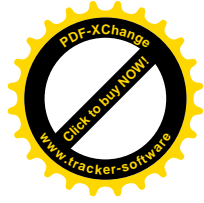
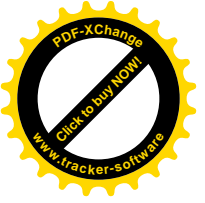
BÀI THỰC HÀNH SỐ 3

Mục đích:

- Giới thiệu phương pháp giao tiếp ngoại vi ngõ vào **nút nhấn**, switch với vi điều khiển
- Giới thiệu phương pháp giao tiếp bàn phím ma trận với vi điều khiển
- Giới thiệu phương pháp giao tiếp ngoại vi sử dụng ngắt ngoài
- Giới thiệu phương pháp sử dụng bộ định thời timer để tạo trễ chính xác

Sau khi kết thúc học phần này, sinh viên có thể:

- Hiểu sự điểm khác biệt của phương pháp giao tiếp ngoại vi sử dụng khái niệm NGẮT
- Hiểu về chu kỳ máy (machine cycle-MC) và sự kết hợp của MC với TIMER để tạo trễ
- Thiết kế một chương trình firmware đơn giản sử dụng ngôn ngữ lập trình cấp cao C dành cho các vi điều khiển (máy tính trên chip) họ 8051 để vận dụng lập trình NGẮT và TIMER

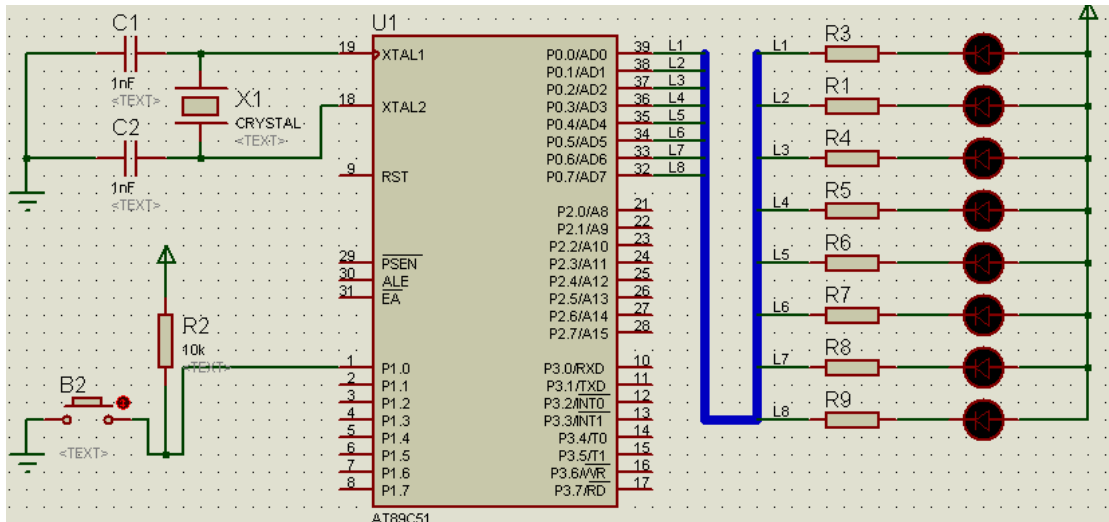


Yêu cầu thực hành:

Tham khảo datasheet của vi điều khiển họ 8051 từ đó hãy trả lời các câu hỏi sau đây:

- Giải thích vai trò của bit C/T trong thanh ghi TMOD
- Làm thế nào vi điều khiển có thể đếm các sự kiện tác động từ bên ngoài
- Nếu một Timer/Counter 0 được sử dụng như một bộ đếm sự kiện, giá trị đếm tối đa là bao nhiêu cho mỗi chế độ Mode 1, Mode 2
- Hãy chỉ ra các chân nào được sử dụng cho: (a) timer/counter 0 (b) timer/counter 1
- Nếu Timer/Counter 0 được sử dụng trong mode 1 nhằm đếm các sự kiện từ bên ngoài, hãy giải thích khi nào bit TF0 được đặt lên mức cao?
- Nếu Timer/Counter 1 được sử dụng trong mode 2 nhằm đếm các sự kiện từ bên ngoài, hãy giải thích khi nào bit TF0 được đặt lên mức cao?

Vấn đề 1 Phương pháp kết nối nút nhấn. Khi nhấn nút B2 (ngõ vào P1.0) nhận mức 0. Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây.



Hình 8: Sơ đồ nguyên lý kết nối VĐK và LED đơn và nút nhấn

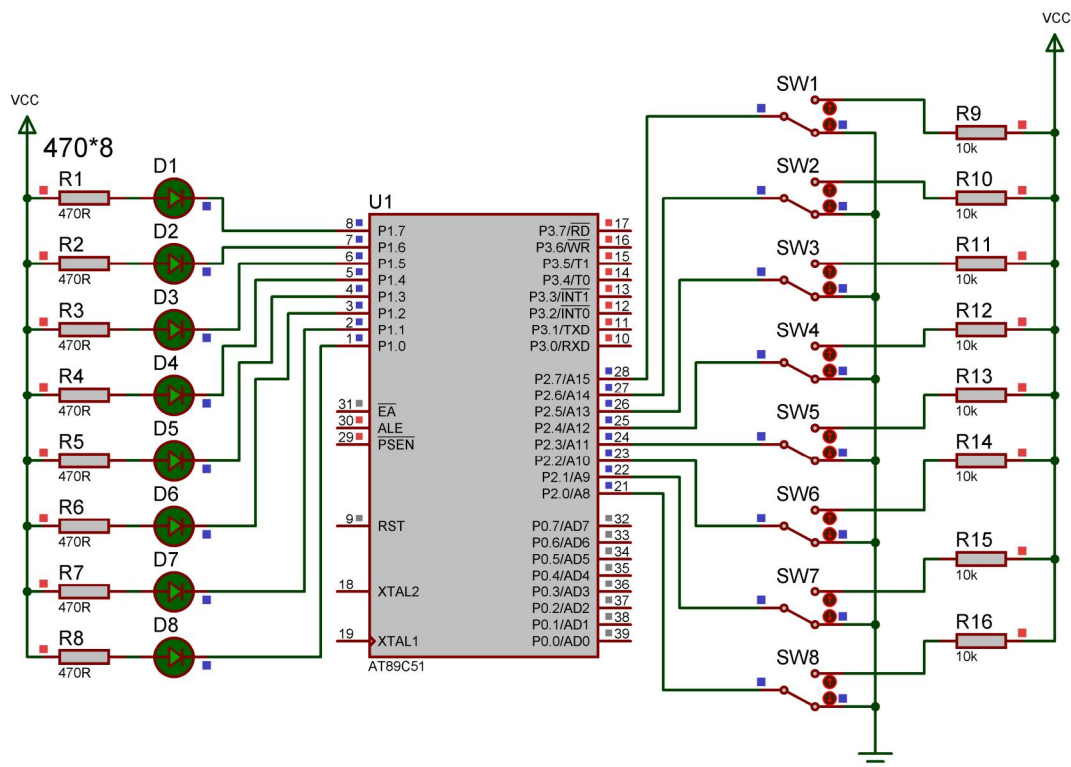
Sử dụng phần mềm Keil C viết chương trình và đặt tên nutnhan_8led.c để điều khiển 8 led lần lượt sáng nhấp nháy, so le nhau khi người dùng không bấm nút B2. Khi người dùng bấm và giữ nút B2 thì các led giữ nguyên trạng thái

```
#include <at89x51.h>
void delay(int interval){
    int i,j;
    for(i=0;i<255;i++){

        for(j=0;j<interval;j++);
    }
}

void main(){
    while(1){
        //Kiểm tra trạng thái
        //chạm P1_0 (dau voi
        //công tắc)
        if(P1_0 == 1){
            P0=0x55;
            delay(100);
            P0=0xAA;
            delay(100); }
        }
}
```

Vấn đề 2 Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây để nhận trạng thái đầu vào từ 8 công tắc (SW) và ngõ ra là 8 đèn LED đơn. Trong đó, trạng thái của 8 đèn LED thể hiện trạng thái của 8 SW



Hình 9: Sơ đồ nguyên lý kết nối VDK và LED đơn và công tắc

Sử dụng phần mềm Keil C viết chương trình và đặt tên là 8sw_8led để điều khiển cho led lần lượt sáng nếu có SW tương ứng tác động

```
#include<at89x51.h>
void main()
{
    unsigned char temp;
    P1=0XFF;
    P2=0XFF;
    while(1)
    {
        temp=P2;
        P1=temp;
        while(P2==temp);
    }
}
```

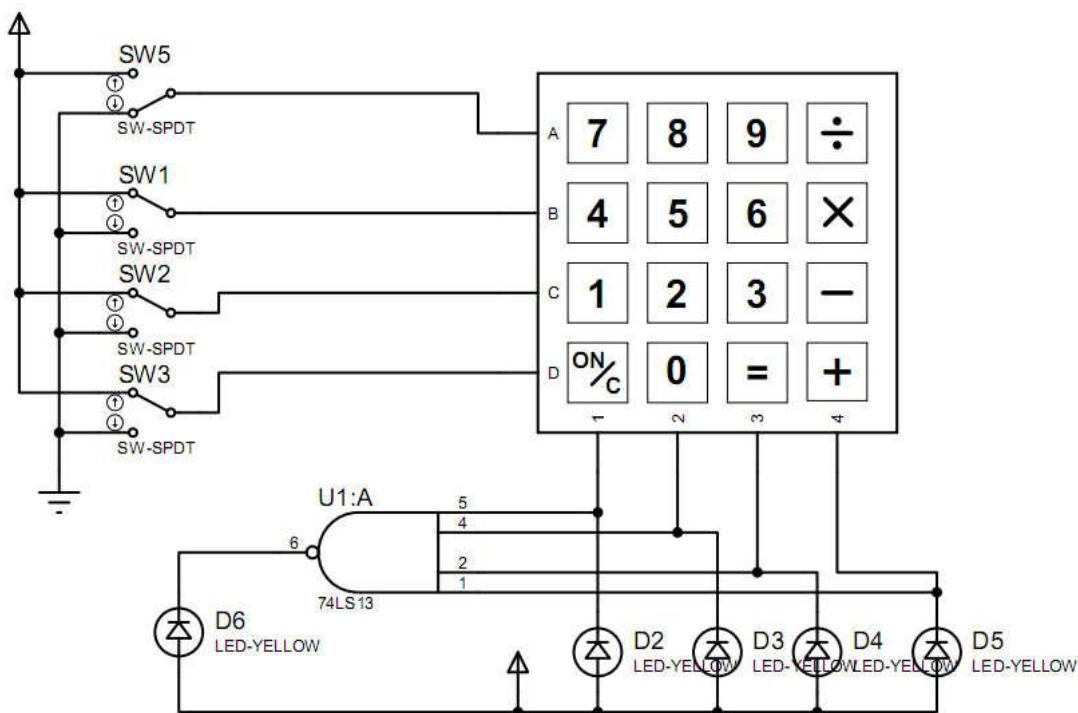
Vấn đề 3 Kiểm tra cơ chế hoạt động của bàn phím.

Bước 1. Thiết kế mạch theo sơ đồ dưới đây

Bước 2. Thử chuyển tất cả các công tắc gạt SW1->SW4 sang nối đất sau đó nhấn các phím, quan sát kết quả và ghi nhận lại

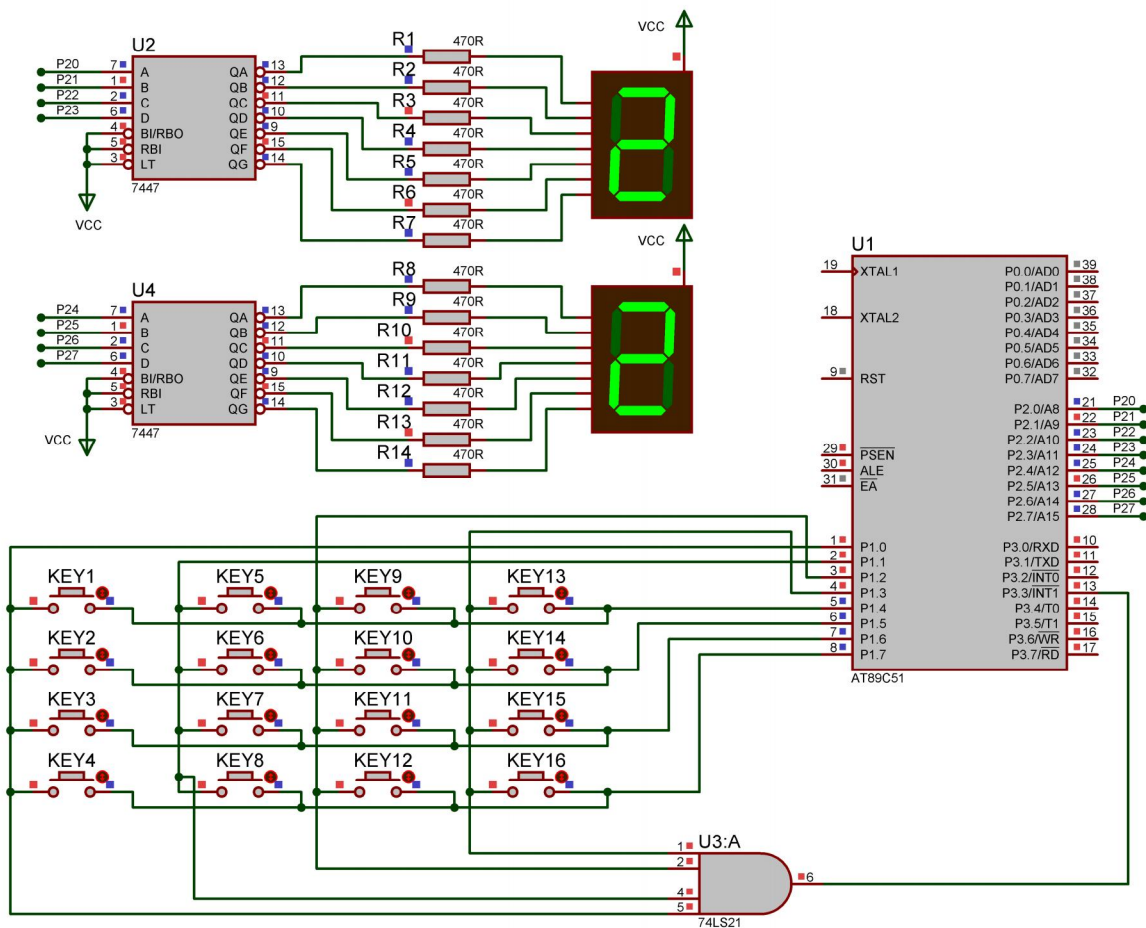
Bước 3. Thử chuyển tất cả các công tắc gạt sang nối nguồn VCC sau đó nhấn các phím, quan sát kết quả và ghi nhận lại

Bước 4. Mỗi thời điểm chỉ cho một công tắc gạt nối đất rồi nhấn các phím và quan sát kết quả và ghi nhận lại



Hình 10: Khảo sát bàn phím ma trận

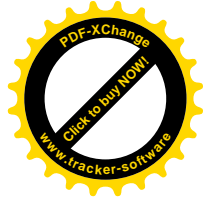
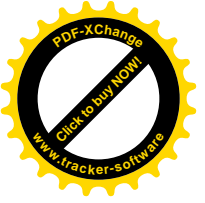
Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây nhằm khảo sát giao tiếp bàn phím 4x4 với vi điều khiển



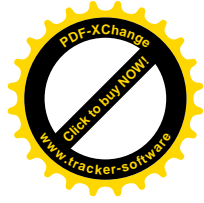
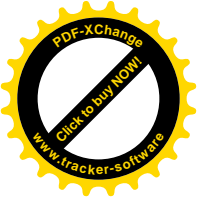
Hình 11: Sơ đồ nguyên lý kết nối bàn phím với vi điều khiển

Sử dụng phần mềm Keil C viết chương trình và đặt tên là matranphim_7doan.c để nhận vị trí của phím gõ vào và xuất báo vị trí ra 2 led 7 đoạn.

```
#include <at89x51.h>
#include <stdio.h>
#define byte unsigned char
void keyinterrupt() interrupt 2
{
    int t;
```



```
byte keycode, scancode, fl ag=0xff;
t=5000;
while(t--);
if(INT1==1)return;
EX1=0;
scancode=0xef;
while(scancode!=0xff)
{
    P1=scancode;
    keycode=P1;
    if((keycode&0x0f)!=0x0f)
        break;
    scancode=(keycode<<1)|0x0f;
}
keycode=~keycode;
P2=keycode;
P1=0X0F;
while(1)
{
    if(INT1==1)
    {
        fl ag=~fl ag;
        if(fl ag==0)
            break;
    }
    t=10000;
    while(t--);
}
EX1=1;
return;
}
void mai n(void)
{
    IE=0;
    EX1=1;
    EA=1;
    P2=0XFF;
    P1=0X0F;
    while(1)
```

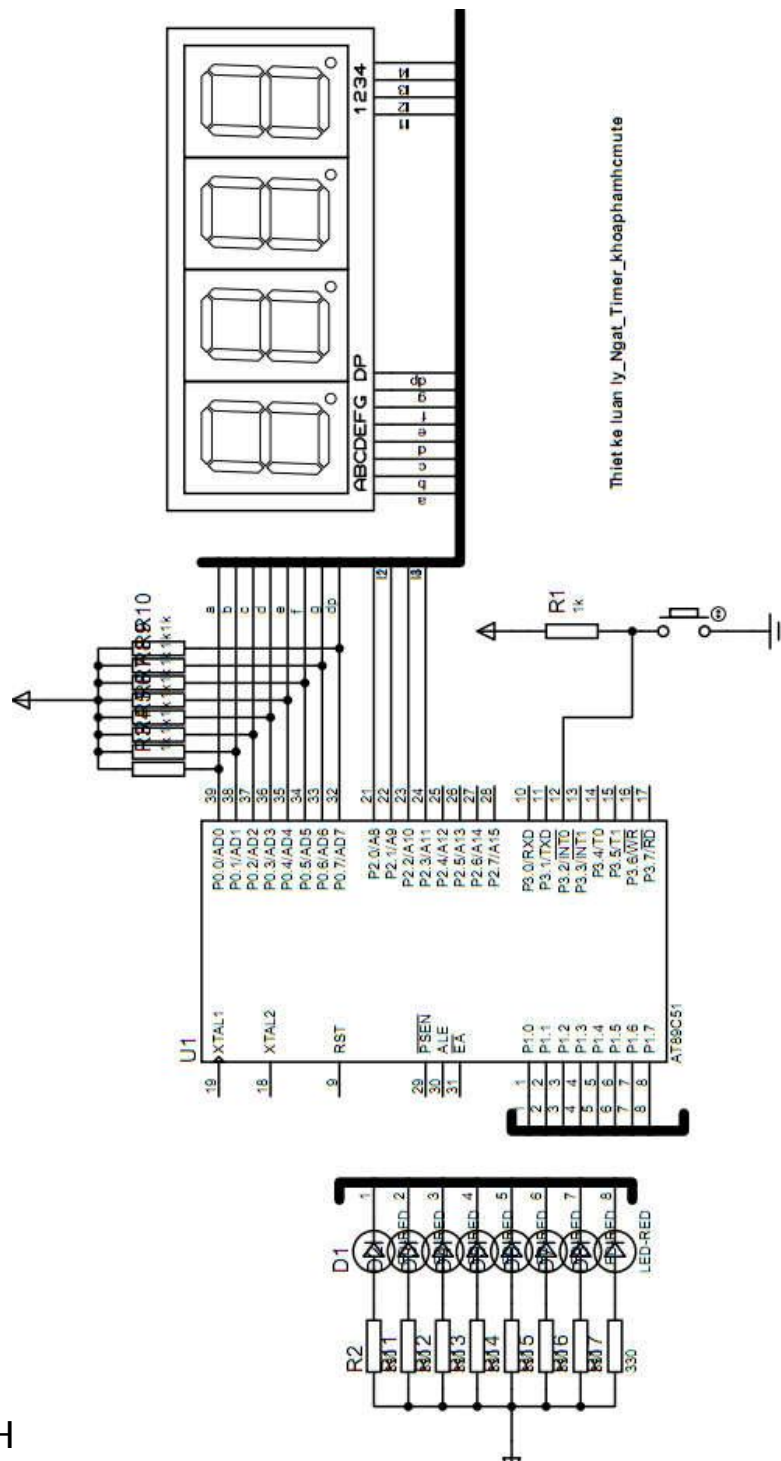


```
{  
    }  
}
```

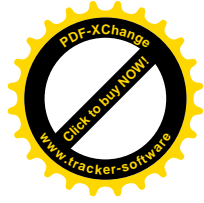
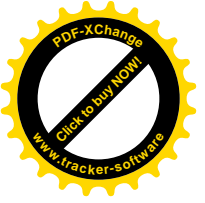
Yêu cầu thực hành:

- Khảo sát ý nghĩa của các thanh ghi chức năng đặc biệt (SFR) IE. Ý nghĩa của việc gán EX=1, EA=1.
- Ôn tập lại lý thuyết về giao tiếp ngoại vi. Hãy trình bày ưu điểm của phương pháp sử dụng ngắt trong việc giao tiếp thiết bị ngoại vi

Vấn đề 4 Áp dụng ngắt và giao tiếp hiển thị LED 7 đoạn. Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây.



H

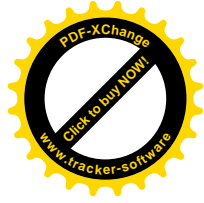
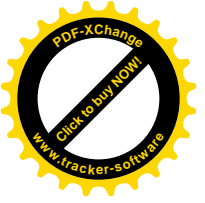


Sử dụng keilC để viết chương trình với tên ex_int1.c để tạo ứng dụng tác động led mỗi khi có sự kiện nhấn phím nhấn trên vi điều khiển

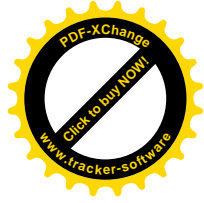
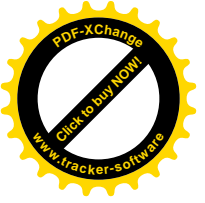
<pre>#include <AT89X51.H> void SetupEx0(void); void main(void) { SetupEx0(); while(1); }</pre>	<pre>void SetupEx0(void) { EA=0; IT0 = 1; EX0=1; EA = 1; } void Ex0Isr(void) interrupt 0 { P1_0 = !P1_0; }</pre>
--	--

Vấn đề 5 Sử dụng cùng sơ đồ nguyên lý như trên. Sử dụng keilC để viết chương trình với tên ex_int2.c sử dụng ngắt ngoài 0 để tạo ứng dụng hiển thị LED 7 thanh tăng giá trị lên 1 mỗi khi có sự kiện nhấn phím nhấn trên vi điều khiển

<pre>#include <at89x51.h> //Khai bao bien toan cuc int count=0; //Chương trình tạo độ trễ void delay(int interval) { int i,j; for(i=0;i<100;i++) { for(j=0;j<interval;j ++); } //Chương trình hiển thị đèn led 7 thanh (không điều khiển dot) void output_7seg(unsigned</pre>	<pre>//Chương trình con hiển thị số (<9999) void display_number(int iNum) { int i; unsigned char pos=0x08; unsigned char temp; for(i=0;i<4;i++) temp=iNum%10; iNum=iNum/10; P2=pos; output_7seg(temp); delay(5); pos=pos>>1;</pre>
---	---

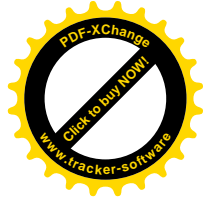
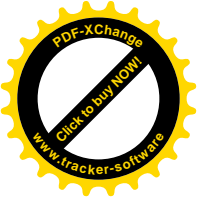


<pre>char value) { unsigned char const mask[10]={0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90}; if(value < 10){ P0=mask[value]; } } //Chương trình con xử lý ngat ngoại 0 void EXT0_Process() interrupt 0{ EA=0; //Cấm ngat count++; EA=1; //Cho phép ngat }</pre>	<pre>} //Chương trình khởi tạo hệ thống void init(){ P3_2=1; //Thiết lập chân P3_2 làm chân vào IE=0x81; //Cho phép ngat ngoại 0 IT0=1; //Ngat theo sườn } void main() { while(1){display_number(count); } }</pre>
---	--



Vấn đề 6 Sử dụng cùng sơ đồ nguyên lý như trên. Sử dụng keilC để viết chương trình với tên delay_timer.c để tạo sự trì hoãn chính xác

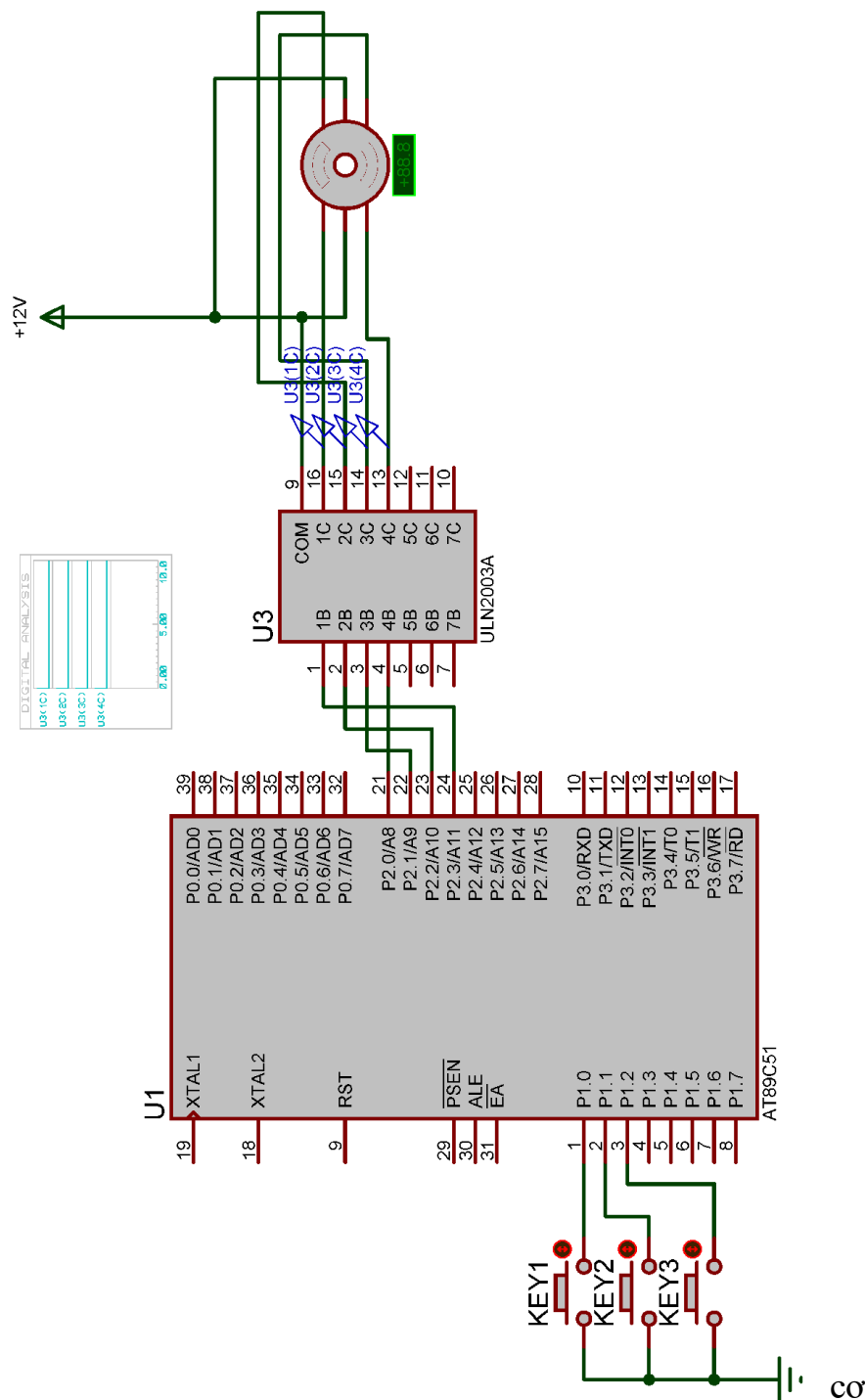
```
#include <at89x51.h>
//tao do tre chinh xac su dung Timer
void delay_hardware_50ms()
{
    TMOD=TMOD & 0xF0; //Xoa thiet lap Timer0
    TMOD=TMOD | 0x01;
    ET0=0; //Khong phat sinh ngat
    TH0=0x3C; //Thiet lap gia tri khoi dau la 3CB0
    TL0=0xB0; //Tuong duong 15536 he 10
    TF0=0; //Xoa co tran timer 0
    TR0=1; //Khoi dong timer 0
    while(TF0==0); //Cho den khi tran
    TR0=0; //Dung timer 0
}
//Chuong trinh tao tre chinh xac 1 s
void delay_hardware_1s()
{
    int i;
    for(i=0; i<20; i++){delay_hardware_50ms();}
}
void main()
{
    while(1)
    {
        P1=0xAA;
        delay_hardware_1s();
        P1=0x55;
        delay_hardware_1s();
    }
}
```



Yêu cầu thực hành:

- Cho biết hạn chế của phương pháp tạo trễ ở các bài đã trình bày trước
- Cho biết ý nghĩa của thanh ghi TMOD
- Để sử dụng bộ định thời 0 (Timer 0) thì cần thiết phải thiết lập các bit nào?
- Sử dụng bộ đo dao động được hỗ trợ trong phần mềm Proteus, sẽ xem xét thời gian trễ mà bộ định thời timer 0 tạo ra

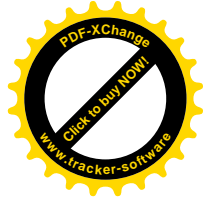
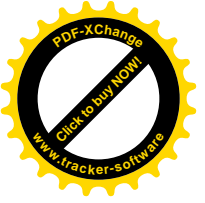
Vấn đề 7 Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây.



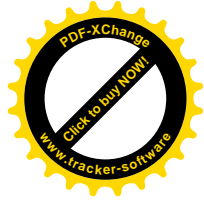
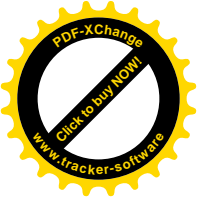
Sử dụng keilC để viết chương trình với tên motor_int1.c để tạo ứng dụng điều khiển động cơ có nội dung như sau

```
#include <reg51.h>    swi tch(temp)
```

Lập trình giao tiếp 8051 và ngoại vi sử dụng C



<pre>sbi t p10=P1^0; sbi t p11=P1^1; sbi t p12=P1^2; #define UP 20 #define DOWN 30 #define STOP 40 void del ay() { unsigned i , j , k; for(i=0; i <0x02; i + +) for(j=0; j <0x02; j + +) for(k=0; k<0xff; k+ +); } mai n() { unsigned char temp; whi l e(1) { i f(p10==0) { temp=UP; P2=0X00; del ay(); } i f(p11==0) {</pre>	<pre> { case DOWN : P2=0X01; del ay(); del ay(); P2=0X02; //0110 del ay(); del ay(); P2=0X04; //1100 del ay(); del ay(); P2=0X08; //1001 del ay(); del ay(); break; case UP : P2 del ay(); del ay(); P2=0X04; del ay(); del ay(); P2=0X02; del ay(); del ay(); P2=0X01; del ay(); del ay(); break; case STOP : P2=0X00; del ay(); del ay(); break; } } }</pre>
---	---



<pre>temp=DOWN; P2=0X00; del ay(); } if(p12==0) { temp=STOP; }</pre>	
---	--

Yêu cầu thực hành:

- Có thể so sánh giá trị k trong hàm delay() với giá trị 1000 không? Tại sao?
- Sửa lại hàm delay(n) để áp dụng tạo độ trễ từ bộ định thời được hỗ trợ từ ngoại vi của bộ vi điều khiển ho 8051. Trong đó, n là tham số quyết định độ trễ của delay theo mili giây (ms)
- Vi điều khiển họ 8051 hỗ trợ các nguồn ngắt nào? Trong những trường hợp nào thì phương pháp thiết kế cần sử dụng đến ngắt? Nêu 3 ví dụ minh họa