

TH Kiến Trúc Và Tổ Chức Máy Tính TS. Phạm Văn Khoa
BÀI THỰC HÀNH SỐ 2

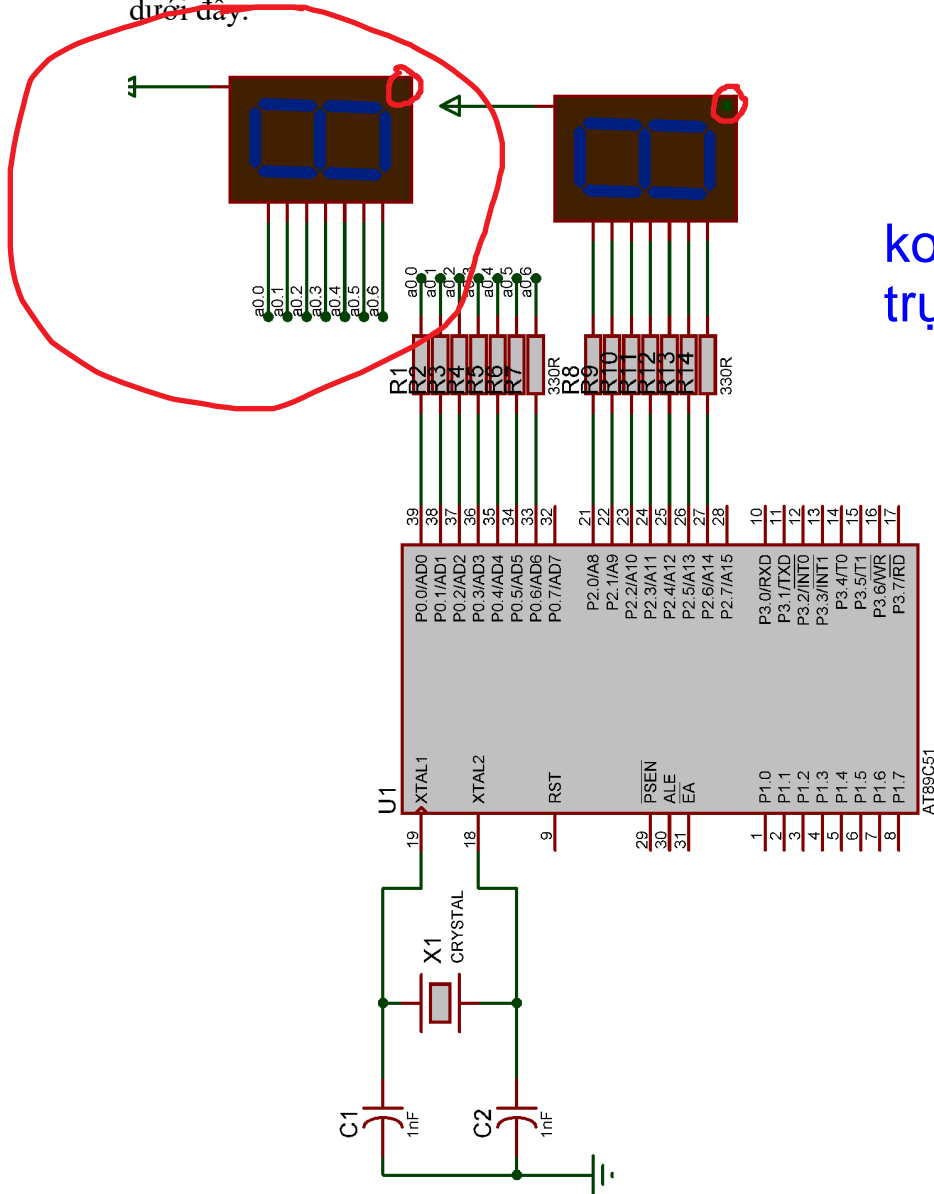
Mục đích:

- Giới thiệu các phương pháp giao tiếp vi điều khiển và ngoại vi LED 7 đoạn như kết nối trực tiếp, giải mã, quét và một số phương pháp khác.
- Giới thiệu một số hiệu ứng đơn giản có thể thực hiện được với LED đơn

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

- Hiểu và lựa chọn các phương pháp giao tiếp giữa vi điều khiển và ngoại vi LED 7 đoạn
- 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 giao tiếp với LED 7 đoạn

Vấn đề 1 Phương pháp kết nối trực tiếp, kết nối với LED 7 thanh kiểu Anode chung. Sử dụng phần mềm Proteus ISIS vẽ mạch điện như hình dưới đây.



ko nên do điều khiển trực tiếp

Hình 3: Sơ đồ nguyên lý kết nối VĐK và LED 7 đoạn trực tiếp

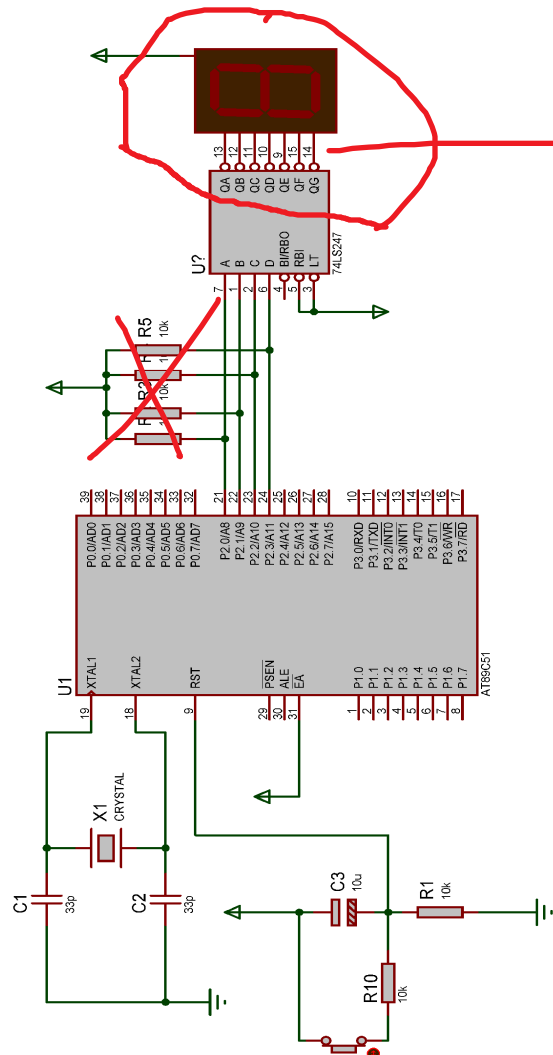
Sử dụng phần mềm Keil C viết chương trình và đặt tên là **7doan_tructiep.c** để điều khiển cho LED 7 thanh lần lượt hiển thị các số từ 00 đến 99

<pre>#include<at89x51.h> #define uchar unsigned char #define uint unsigned int void delay_ms(uint x); void giaima(void); void hienthi(void); uchar donvi,chuc; uint i,j,a; int dig[]={0xC0,0xF9,0xA4,0xB0,0 x99,0x92,0x82,0xF8,0x80,0x90 }; void main (void) { while(1) { for(i=0;i<100;i++) { a=i; for(j=0;j<5000;j++) { giaima(); hienthi(); } } }</pre>	<pre>} } //chuong trinh delay void delay_ms(uint x) { uchar k; while(x-->0) { for(j=0;k<125;k++) }{;} } } //chuong trinh giai ma void giaima(void) { chuc=a/10; donvi=a%10; } //chuong trinh hien thi void hienthi(void) { P0=dig[donvi]; P2=dig[chuc]; } }</pre>
--	---

du?

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

Vấn đề 2 Phương pháp kết nối sử dụng bộ giải mã, kết nối với LED 7 thanh kiểu Anode chung. Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây. IC 74LS47 và IC 7446 là các IC giải mã từ mã BCD ra mã 7 thanh.



có thể mở rộng led 7 đoạn
nhưng tốn thêm ic -> tăng chi
phí, diện tích thiết kế

Hình 4: Sơ đồ nguyên lý kết nối VĐK và LED 7 đoạn thông qua IC giải mã

Sử dụng phần mềm Keil C viết chương trình và đặt tên là 7doan_giaima.c để điều khiển cho LED 7 thanh lần lượt hiển thị các số từ 0 đến 9 trên 1 LED 7 đoạn

<pre>#include<at89x51.h> int dem=0; void delay(unsigned long time) { unsigned long i; for(i=0;i<time;i++) { } } void main(void)</pre>	<pre>{ P2=0x00; while(dem<10) { P2=dem; dem++; delay(10000); } }</pre>
--	---

<https://www.echipkool.com/2011/11/ic-74ls47-giai-ma-led-7-oan.html>

Yêu cầu thực hành:

<https://www.bachkhoadienfu.com/2018/08/ic-7447-ic-giai-ma-bcd-ra-led-7-thanh.html>

- Khảo sát chức năng và đặc tính về điện của 2 IC giải mã 7447 và 7446
- Hãy cho biết ưu và nhược điểm của phương pháp sử dụng IC giải mã so với phương pháp kết nối trực tiếp

IC 7447: giải mã kí giành riêng cho LED 7 thanh Anot chung. Ứng dụng khi ta cần hiện thị số trên led 7 thanh trong mạch. Chân 16 cấp nguồn Vcc cụ thể ở đây là 5V nếu quá 5V ic này sẽ bị chết

IC 7446: cũng giải mã ra LED 7 đoạn Anode chung. Loại IC Giải mã 7446 có thể cho phép sử dụng cho LED 7 đoạn Anode chung với điện áp tối đa là 30V, IC Giải mã 7447 cho phép sử dụng cho LED 7 đoạn có điện áp tối đa 15V.

<http://tri-heros.net/vi/news/Mach-Logic/7446-567/>

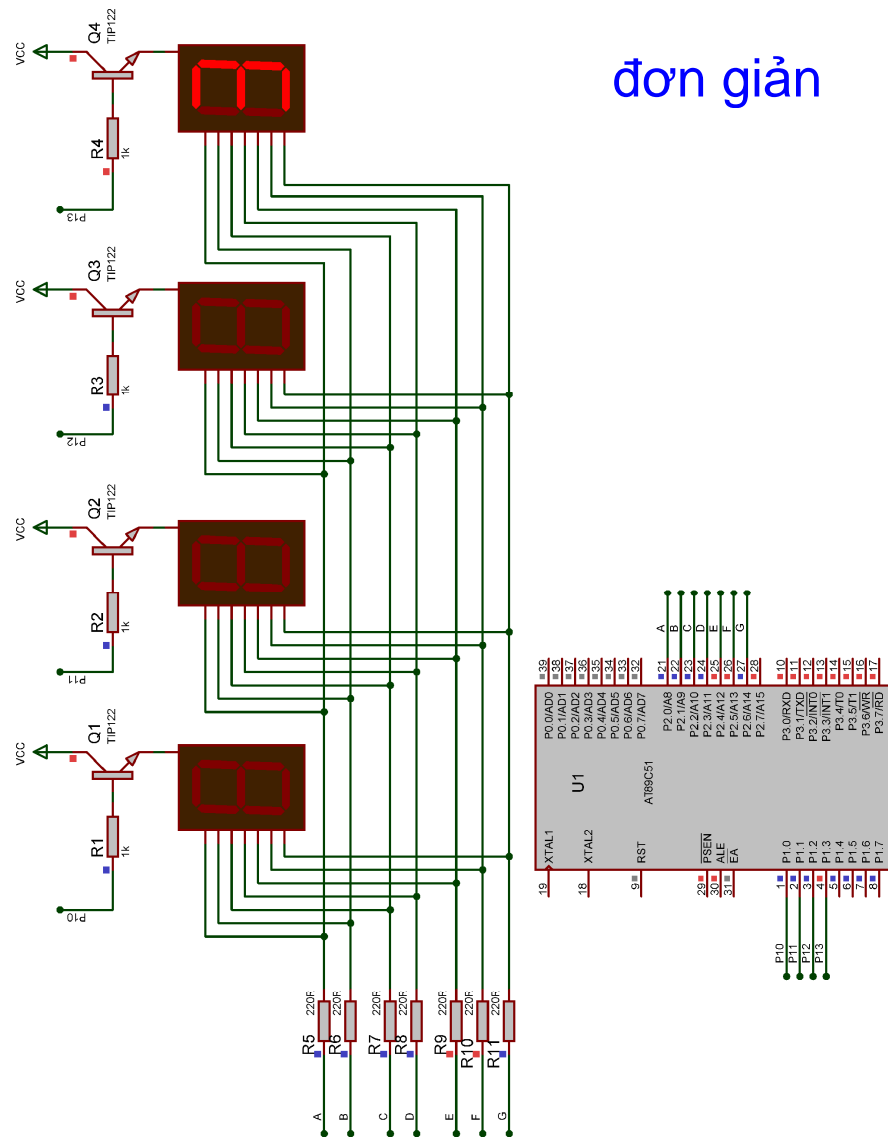
ưu điểm chính là mã BCD cho phép dễ dàng chuyển đổi giữa thập phân và nhị phân
 Nhược điểm, tốn chân IC, nên dùng phương pháp quét led 7 đoạn dùng tranzitor sẽ hiệu quả hơn, đỡ tốn chân IC

<https://mobitool.net/mach-dem-bcd.html>



Vấn đề 3 Phương pháp kết nối trực tiếp kết hợp quét cho nhiều LED 7 đoạn. Sử dụng phương pháp quét LED để hiển thị. Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây. Trong đó

- Các chân A, B, C, D, E, F, G, H kết nối với các chân từ P2.0 đến P2.6
- Các chân quét LED P10, P11, P12, P13 kết nối với các chân từ P1.0 đến P1.3



Hình 5: Sơ đồ nguyên lý kết nối VĐK và LED 7 đoạn thông phương pháp quét sử dụng transistor

Sử dụng phần mềm Keil C viết chương trình và đặt tên là **7doan_quetLED1.c** để điều khiển cho LED 7 đoạn lần lượt hiển thị các số từ 0 đến F lần lượt trên 4 LED 7 đoạn

<pre>#include <at89x51.h> #include <stdio.h> #define DELAYTIME 65000 unsigned int temp1; void delay(unsigned int temp) { while(--temp); } void main() { P1=0; //LED is off while(1) { P1=1; P2=0Xc0; temp1=DELAYTIME; delay(temp1); P1=2; P2=0XF9; temp1=DELAYTIME; delay(temp1); P1=4; P2=0XA4; temp1=DELAYTIME; delay(temp1); P1=8; </pre>	<pre> P1=8; P2=0Xf8; temp1=DELAYTIME; delay(temp1); P1=1; P2=0X80; temp1=DELAYTIME; delay(temp1); P1=2; P2=0X90; temp1=DELAYTIME; delay(temp1); P1=4; P2=0X08; temp1=DELAYTIME; delay(temp1); P1=8; P2=0X03; temp1=DELAYTIME; delay(temp1); P1=1; P2=0X46; </pre>
--	--

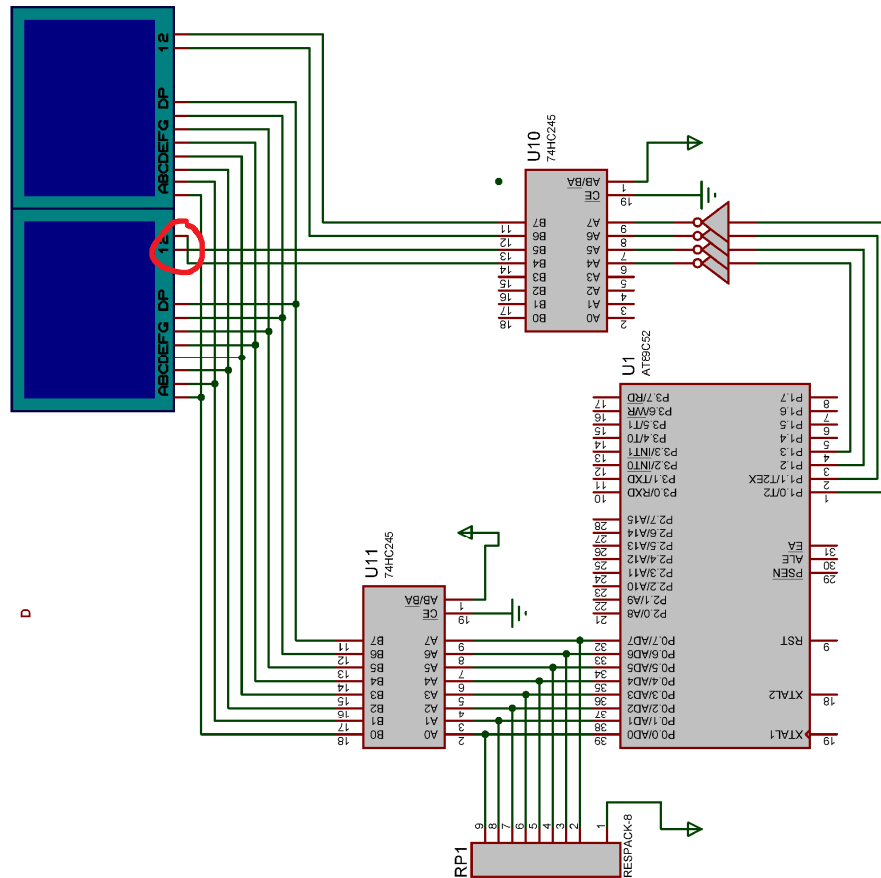
<i>P2=0XB0;</i>	<i>temp1=DELAYTIME;</i>
<i>temp1=DELAYTIME;</i>	<i>delay(temp1);</i>
<i>delay(temp1);</i>	<i>P1=2;</i>
<i>P1=1;</i>	<i>P2=0X21;</i>
<i>P2=0X99;</i>	<i>temp1=DELAYTIME;</i>
<i>temp1=DELAYTIME;</i>	<i>delay(temp1);</i>
<i>delay(temp1);</i>	<i>P1=4;</i>
<i>P1=2;</i>	<i>P2=0X06;</i>
<i>P2=0X92;</i>	<i>temp1=DELAYTIME;</i>
<i>temp1=DELAYTIME;</i>	<i>delay(temp1);</i>
<i>delay(temp1);</i>	<i>P1=8;</i>
<i>P1=4;</i>	<i>P2=0X0E;</i>
<i>P2=0X82;</i>	<i>temp1=DELAYTIME;</i>
<i>temp1=DELAYTIME;</i>	<i>delay(temp1);</i>
<i>delay(temp1);</i>	<i>temp1=DELAYTIME;</i>
	<i>delay(temp1);</i>
	<i>}</i>
	<i>}</i>

Yêu cầu thực hành:

Thay đổi thông số DELAYTIME ở các giá trị (20000, 40000, 80000) biên dịch lại chương trình và nạp lại cho VĐK. Sau đó quan sát kết quả mô phỏng

20000	20000: nhanh
-25536	40000: chậm hơn 20000
14464	80000: nhanh hơn 40000

- ✓ **Vấn đề 4** Phương pháp kết hợp quét LED và sử dụng các IC đệm 74245. Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây.



Hình 6: Sơ đồ nguyên lý kết nối VĐK và LED 7 đoạn thông phương pháp quét sử dụng IC đệm

Sử dụng phần mềm Keil C viết chương trình và đặt tên là **7doan_quet_dem245.c** để điều khiển cho LED 7 thanh lần lượt hiển thị các số từ 0000 đến 9999 lần lượt trên 4 LED 7 đoạn

<pre>#include < at89x51.h > void delay() {</pre>	<pre>for(m=0;m<10;m++) for (n=0;n<10;n++) for (i=0;i<10;i++) for (j = 0 ; j <10 ; j ++)</pre>
---	--

<pre> int i; for(i=1;i<200;i++); } unsigned char num[10] = {0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8,0x80,0x90 }; void main() { int x,m,n,j,i=0 ; unsigned char p0; while(1) { i=0;n=0,m=0; j=0; </pre>	<pre> for(x=100;x>0;x--) { P0 = num[j] ; P1 = 0xfe; delay(); P1=0xff; P1 = 0xfd; P0 = num[i]; delay(); P1=0xff; P1 = 0xfb; P0 = num[n]; delay(); P1=0xff; P1 = 0xf7; P0 = num[m]; delay(); P1=0xff; } } </pre>
--	---

Yêu cầu thực hành:

- Áp dụng phương pháp kết nối ở trên hãy mở rộng thiết kế nhằm giao tiếp với 8 LED 7 đoạn
- Hãy so sánh ưu và nhược điểm trong phương pháp vấn đề 3 và 4
- Chỉnh sửa lại chương trình **vấn đề 2** nhằm xây dựng chương trình con hiển thị LED 7 thanh có khai báo như sau **void**

Display_7seg (unsigned char value) Hàm này có chức năng

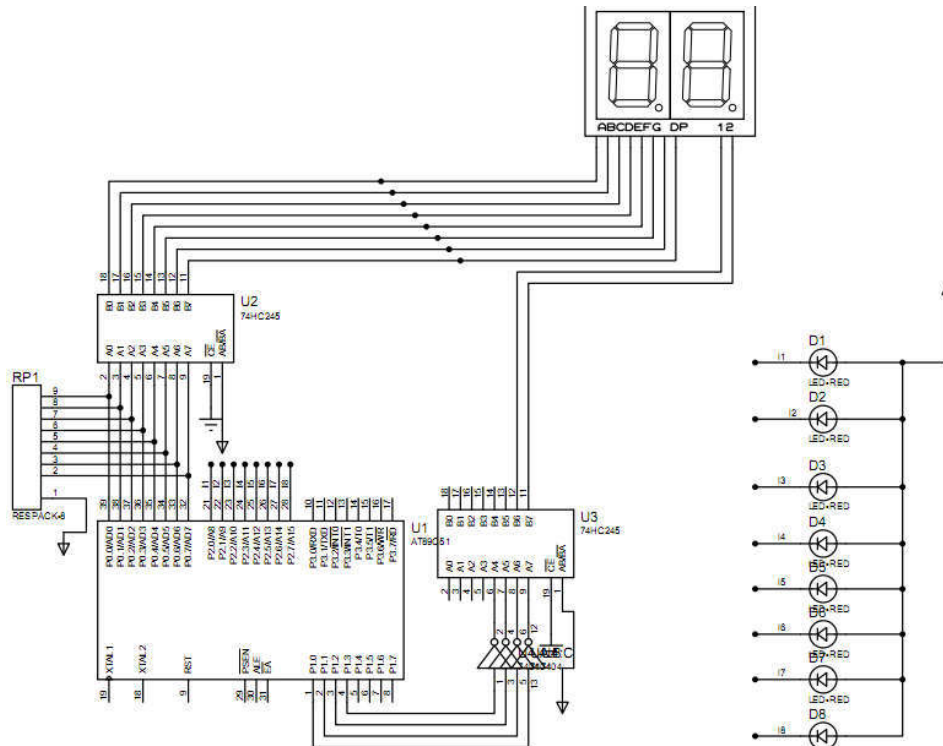
Nhược điểm, tốn chân IC, nên dùng phương pháp quét led 7 đoạn
dùng tranzitor sẽ hiệu quả hơn, đỡ tốn chân IC

nhận đầu vào là các số $0,1,\dots,9$ và điều khiển LED 7 thanh để hiển thị các số đó.

- Với 8 đường ngõ ra từ vi điều khiển, có thể hiển thị được tối đa được bao nhiêu led7 đoạn. Trình bày phương án thiết kế với sơ đồ nguyên lý.

25 → P P 3

- ✓ - Sử dụng phần mềm Proteus ISIS vẽ mạch điện dưới đây. Mạch hoạt động với nguyên lý sau: 2 LED 7 đoạn hoạt động với phương pháp quét và hiển thị từ 00-90. Số LED đơn sáng tương ứng với số hàng chục của 7 đoạn hiển thị. Ví dụ 23 thì 2 LED đơn sáng. Khi đếm đến 90 thì quay lại đếm lùi về 00. Mỗi lần giảm số hàng chục đi 1 thì 1 LED đơn tắt. Quá trình hoạt động liên tục theo nguyên lý ở trên.



Hình 7: Sơ đồ nguyên lý kết nối VDK và LED 7 đoạn
Lập trình giao tiếp 8051 và ngoại vi sử dụng C