

TS. NGUYỄN MẠNH GIANG

CẤU TRÚC - LẬP TRÌNH - GHÉP NỐI VÀ ỨNG DỤNG VI ĐIỀU KHIỂN

TẬP HAI
GHÉP NỐI VÀ ỨNG DỤNG VDK 8051/8052

(Tái bản lần thứ nhất)

NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM

**Công ty Cổ phần sách Đại học - Dạy nghề – Nhà xuất bản Giáo dục Việt Nam
giữ quyền công bố tác phẩm.**

04-2009/CXB/211-2117/GD

Mã số: 7B661y9-DAI

LỜI NÓI ĐẦU

Đây là tập hai của bộ sách “Cấu trúc, lập trình, ghép nối và ứng dụng vi điều khiển”. Tài liệu này giới thiệu về ghép nối với các thiết bị ngoài (hiển thị, bàn phím, nhớ ngoài, cổng mở rộng (song song và nối tiếp)) và các hệ VDK dùng trong đo lường, điều khiển tự động và truyền thông.

Tiếp theo tập một, tập hai gồm 8 chương. Trong mỗi chương, đều có mô tả các thiết bị ngoài, cách ghép nối và lập trình với các ví dụ để chạy thử mô phỏng trên PINNACLE và PROTEUS.

Chương 11 Ghép nối và ứng dụng đèn LED.

Chương 12 Ghép nối và ứng dụng đèn LED ma trận.

Chương 13 Ghép nối với màn hình tinh thể lỏng (LCD).

Chương 14 Ghép nối với bàn phím.

Chương 15 Ghép nối với bộ nhớ ngoài của vi điều khiển 8051/8052.

Chương 16 Ghép nối mở rộng cổng song song cho vi điều khiển.

Chương 17 Ghép nối mở rộng cổng nối tiếp cho vi điều khiển.

Chương 18 Ghép nối trong hệ vi điều khiển.

Trong tài liệu này cũng giới thiệu cách dùng phần mềm Proteus trong thiết kế như vẽ sơ đồ nguyên lý của hệ VDK, trong chạy mô phỏng các ví dụ với tệp đuôi .HEX sau khi hợp dịch và chạy mô phỏng (từng bước và cả chương trình) trên Pinnacle.

Tài liệu được biên soạn từ cơ bản tới chuyên sâu, cho người tự học, công nhân học nghề, sinh viên cao đẳng, đại học, cao học và nhà nghiên cứu trong các lĩnh vực tự động hoá, điện tử, vật lý kỹ thuật, cơ tin và thông tin.

Tùy trình độ và mục đích sử dụng, người đọc có thể bỏ qua hay chỉ chú ý tới phần mình quan tâm. Vì trong thời đại thông tin, mọi người có thể truy cập mạng Internet, nên cách trình bày chỉ tóm tắt ý chính và cơ bản, người đọc có thể tra cứu thêm về các vi mạch và ứng dụng trên mạng Internet.

Tác giả xin chân thành cảm ơn các Ban lãnh đạo, các đồng nghiệp ở Trường Đại học Bách Khoa Hà Nội, Đại học Sư phạm Kỹ thuật Nam Định, Viện Nghiên cứu Kỹ thuật hạt nhân Hà Nội, nơi tác giả đã công tác và cộng tác về sự khích lệ, cung cấp tài liệu và trao đổi chuyên môn.

Tác giả xin chân thành cảm ơn Ban Giám đốc, Ban lãnh đạo Công ty Cổ phần Sách Đại học - Dạy nghề của Nhà xuất bản Giáo dục Việt Nam đã giúp đỡ để tài liệu sớm ra mắt bạn đọc.

Tác giả mong được bạn đọc góp ý cho tài liệu được hoàn thiện hơn. Thư góp ý xin gửi về Công ty Cổ phần Sách Đại học - Dạy nghề (HEVOBCO), Nhà xuất bản Giáo dục Việt Nam, 25 Hàn Thuyên, Hà Nội, điện thoại 043 8.264974.

TÁC GIẢ

GHÉP NỐI VÀ ỨNG DỤNG ĐÈN LED

Mục đích của chương:

- Hiểu được nguyên tắc hoạt động, cách mắc và lập trình cho các đèn hiển thị LED đơn.
- Hiểu được nguyên tắc hoạt động, cách mắc và lập trình cho các đèn hiển thị LED 7 đoạn.

Trong quá trình sử dụng thiết bị số, trong đó có máy vi tính, vi xử lý, vi điều khiển, chúng ta cần nhìn thấy được nội dung của dữ liệu (địa chỉ, dữ liệu, mã lệnh) dưới dạng các con số tự nhiên (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) hay các chữ cái (a, b, c, d, ... z).

Có các loại hiển thị sau:

– *Đèn nóng sáng dây tóc thế nuôi thấp (5V) hay đèn diot phát quang*: khi có dòng điện nhỏ chạy qua (cỡ mA). Loại LED đơn này đơn giản, rẻ tiền, nhưng chỉ biểu diễn nội dung từng bit có giá trị 0 (tắt) hay 1 (sáng). Nếu nhiều bit, các đèn này biểu diễn thành các số nhị phân, khó hình dung thành các số thập phân.

– *Đèn diot (LED) 7 đoạn (thanh)*: (a, b, c, d, e, f, g) bố trí thành hình chữ số 8. Tùy các thanh sáng (có dòng điện chạy qua), ta có các chữ số hay chữ khác nhau. Có hai loại diot là:

+ *Anôt chung*: Các anôt nối với nhau và nối với nguồn nuôi sẽ phát sáng khi các katôt riêng rẽ của từng thanh có mức logic 0.

+ *Katôt chung*: Các katôt nối với nhau và nối với đất chung, sẽ phát sáng khi các anôt riêng rẽ của từng thanh có mức logic 1.

Đèn LED 7 thanh này hiển thị trực giác được các chữ số và các ký tự, nhưng có nhược điểm là phải giải mã (bằng vi mạch, ví dụ 7447 hay bằng chương trình) thành các con số dạng BCD (Binary Coded Decimal – số thập phân mã hoá thành dạng nhị phân).

– *Đèn LED nhiều đoạn*: Các đoạn được bố trí theo các đường ngang, dọc và chéo của 4 hình chữ nhật ghép lại. Loại này biểu diễn được chữ và con số một cách tinh tế hơn LED 7 đoạn nhưng đắt tiền và điều khiển phức tạp.

– *Đèn ma trận diot (ma trận LED)*: Gồm nhiều diot có kích thước rất nhỏ (coi như một điểm), sắp xếp thành các ma trận 5 x 7 theo 7 hàng và 5 cột. Cũng có các loại anôt chung hay katôt chung như đèn LED 7 đoạn. Tùy điều khiển các dòng điện chạy qua, các diot điểm này phát sáng, tạo thành các điểm của bức tranh của các con số thập phân hay các chữ a, b, c, d, ... z. Đèn ma trận diot này biểu diễn đẹp các chữ hay số nhưng đắt tiền, phải xác định các mã của chữ hay số theo từng ma trận với các hàng và cột và đưa ra biểu diễn với tốc độ thích hợp với mắt người để quan sát.

– *Màn hình tinh thể lỏng LCD (Liquid Crystal Diot)*: gồm nhiều diot trong một khối tinh thể lỏng (thường có hai hàng, nhiều cột). LCD này có nguyên tắc sáng tối tương tự ma trận diot, có thể biểu diễn được các số, chữ và hình ảnh (graph). Loại LCD thông minh này có ưu điểm hơn LED 7 đoạn là biểu diễn các chữ giống chữ in, hình ảnh rõ nét, có độ phân giải cao.

Các LCD này được dùng rộng rãi trong các thiết bị số tiên tiến như tivi, đầu đĩa, đồng hồ, các thiết bị điện tử trong gia đình (lò sấy, bếp điện, lò nung,...).

Hiện nay, còn có loại màn hình tinh thể lỏng cỡ lớn (kích thước từ nhiều centimet tới mét), có nhiều hàng và cột, có thể biểu diễn được cả hình ảnh của một bức tranh lớn hay cả một thông báo gồm nhiều hàng chữ và số. Vì giá thành của loại LCD này còn đắt nên chưa thông dụng.

11.1. ĐÈN HIỂN THỊ LED RỜI

11.1.1. Cấu tạo và cách mắc

Đèn dây tóc nóng sáng và đèn diot phát sáng LED (Luminescence Emitted Diot) sẽ sáng khi có dòng điện đủ lớn (cỡ mA) chạy qua. Do đó, hai đèn này sáng dùng để chỉ thị:

- *Giá trị logic 1 (cỡ 3V–5V)*: Khi đèn sáng vì có dòng điện chạy qua.
- *Giá trị logic 0 (cỡ 0V–0,2V)*: Khi đèn tắt vì không có dòng điện chạy qua.

Đèn dây tóc nóng sáng có cấu tạo đơn giản nhưng tiêu tốn điện năng (dòng điện chạy qua lớn mới đủ sáng) nên ít dùng. Đèn diot dùng nhiều hơn vì tiêu tốn ít điện năng và có nhiều loại (hình 11.1) với màu sắc khác nhau (xanh, đỏ, tím, vàng...).

Đèn LED rời hay đơn này là một diot có hai cực, đặt trong một vỏ bọc thủy tinh hay nhựa, dùng để chỉ thị từng bit với hai trạng thái logic 0 (đèn tắt) và logic 1 (đèn sáng). Các LED đơn này được mắc với nguồn nuôi 5V qua một điện trở hạn chế dòng điện cỡ $1k\Omega - 3k\Omega$ hay qua một mạch

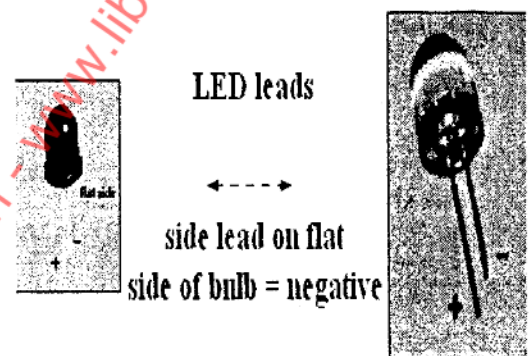
logic đảo colectơ hở (ví dụ vi mạch 7403) để tăng công suất dòng điện qua diot. Khi đó, đèn sáng chỉ thị logic 0 và đèn tắt chỉ thị logic 1. Đèn có hai cực anôt và katôt nên mắc với nguồn điện qua một điện trở hạn chế dòng (cỡ hàng trăm tới hàng ngàn Ω).

Đèn LED này sử dụng đơn giản nhưng chỉ thị con số dạng nhị phân, người dùng khó hình dung ra con số dạng thập phân dưới dạng nhiều chữ số tự nhiên (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).

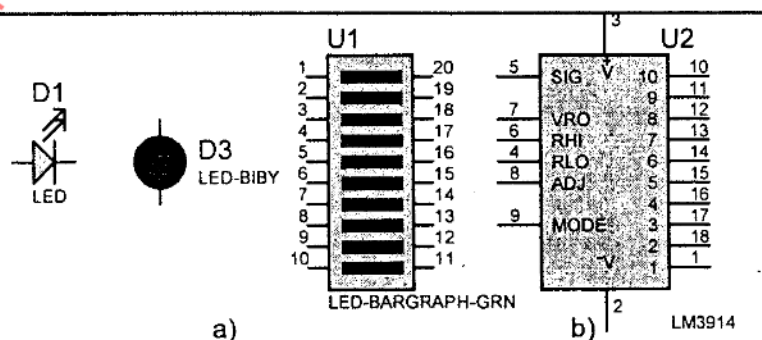
11.1.2. Các loại LED rời

Hiện có nhiều hãng điện tử chế tạo các loại đèn LED với kích cỡ, màu sắc và mục đích sử dụng khác nhau.

Trong chương trình mô phỏng PROTEUS, có mô phỏng một số loại LED như ở hình 11.2.



Hình 11.1. Một số loại đèn LED



Hình 11.2. Các loại LED đơn (a) và vi mạch điều khiển đèn công suất (b)

Hình 11.2 giới thiệu ba loại LED: loại phát quang, loại chỉ thị tròn và loại thanh. Vi mạch LN3914 có trên hình 11.2 là vi mạch điều khiển LED công suất.

11.1.3. Lập trình điều khiển LED rời

Nói chung, các đèn hiển thị LED rời có thể mắc với cổng P0, P1, P2 và P3 của vi điều khiển (VĐK) thuộc họ MCS – 51. Có thể mắc trực tiếp hay qua một điện trở hạn chế dòng điện hoặc một cổng logic để tăng công suất.

Ví dụ 11.1a giới thiệu chương trình đưa dữ liệu cố định ra các cổng Pi dùng lệnh gộp @WritePs < 1, 2 > với :

- Thông số 1 là giá trị của dữ liệu (dạng HEX) đưa vào thanh ghi chứa A.
- Thông số 2 là chỉ số của cổng VĐK.

Phần sau của chương trình trên làm đèn LED sáng dần từ phải qua trái (dùng lệnh RLC A) cho tới khi C = 1. Có thể thay lệnh RLC A bằng lệnh RL A để dịch và lặp lại một số lần (thay bằng lệnh MOV R0, #N lần và DJNZ R0, LAP).

Ví dụ 11.1b là chương trình của ví dụ 11.1a dạng đuôi. LST sau khi hợp dịch trong PINNACLE.

Từ ví dụ trên, ta có thể viết chương trình cho LED đơn sáng dần dịch về bên phải (dùng lệnh RRC A) hay bên trái (dùng lệnh RLC A).

; Ví dụ 11.1a: Chương trình dùng lệnh gộp hiển thị trên LED rời		
macro @WritePs <1,2>	; lệnh gộp ghi dữ liệu	lap: call delay
mov a, #1	; dữ liệu	rlc a
mov P12, a	; ghi ra cổng P12	mov P3, a
endmac	; kết thúc lệnh gộp	jnc lap
ORG 00H		sjmp \$; kết thúc chờ
@WritePs 55h, 0	; ghi 55h ra P0	delay: mov r0, #250
call delay		del1: mov r1, #200
@WritePs 0AAh, 1	; ghi 0AAh ra P1	del2: djnz r1, del2
call delay		djnz r0, del1
@WritePs 00h, 2	; ghi 0AAh ra P1	ret
call delay		end
@WritePs 01h, 3	; ghi 0AAh ra P1	
; Ví dụ 11. 1b: Dạng đuôi. LST của chương trình dùng lệnh gộp hiển thị trên LED đơn		
1 0000	; ví dụ 11.1a. LEDR, hiển thị trên LED rời	
6 0000	ORG 00H	
7 0000	MACLEVEL 00017	
0000	mov a, #55h	; dữ liệu
7 0002	mov P0, a	; ghi ra cổng P0
7 0004	MACLEVEL 0000	
8 0004	call delay	
9 0007	MACLEVEL 0002	
9 0007	mov a, #0AAh	; dữ liệu
9 0009	mov P1, a	; ghi ra cổng P1
9 000B	MACLEVEL 0000	

```

10 000B    call delay
11 000E MACLEVEL 0003
11 000E    mov a, #00h    ; dữ liệu
11 0010    mov P2, a      ; ghi ra cổng P2
11 0012 MACLEVEL 0000
12 0012    call delay
13 0015 MACLEVEL 0004
13 0015    mov a, #01h    ; dữ liệu
13 0017    mov P3, a      ; ghi ra cổng P3
13 0019 MACLEVEL 0000
14 0019 lap: call delay
15 001C    rlc a
16 001D    mov P3, a
17 001F    nc lap
18 0021    sjmp $         ; kết thúc chờ
19 0023 delay: mov r0, #250
20 0025 del1: mov r1, #200
21 0027 del2: djnz r1, del2
22 0029    djnz r0, del1
23 002B    ret

```

11.1.4. Lập trình điều khiển LED rời trong các biển quảng cáo

Với ví dụ 11.1 trên, chúng ta có thể viết chương trình đưa hiển thị như mong muốn:

- Cho tắt cả các đèn nối với một cổng nào đó sáng khi đưa A = #0FFh.
- Cho mỗi đèn sáng lần lượt từ trái qua phải dùng lệnh SETB C và dịch phải qua C với lệnh RRC A.
- Cho mỗi đèn sáng lần lượt từ phải qua trái dùng lệnh SETB C và dịch trái qua C với lệnh RLC A.
- Cho hai đèn sáng dồn từng đèn từ hai đầu vào giữa với các giá trị đưa trực tiếp vào thanh ghi A như 10000001B, 01000010B, 00100100B, 00011000B.
- Cho các đèn sáng dồn từ hai đầu vào giữa với các giá trị A là 10000001B, 11000011B, 11100111B, 11111111B.
- Cho các đèn sáng dồn từ giữa ra hai đầu với các giá trị A là 00011000B, 00111100B, 01111110B, 11111111B.

Có ba cách viết chương trình hiển thị động (đèn sáng chạy) trên các LED của các biển quảng cáo là:

- Đưa các giá trị (0 cho tắt và 1 cho sáng) trực tiếp ra các cổng.
- Thành lập một bảng dữ liệu, rồi đọc bảng để đưa ra cổng (ví dụ 11.2).
- Dùng lệnh logic và vòng lặp (ví dụ 11.1 và ví dụ 11.3), còn có thể dùng lệnh ORL.

; Ví dụ 11.2: Chương trình sáng dồn vào giữa	; Ví dụ 11.3: Chương trình sáng dồn 16 bit
<pre> org 00h lb: mov dptr, #bang lbl: mov a, #00h movc a, @a+dptr </pre>	<pre> macro @sangdon <1, 2, 3, 4> mov r0, #1h mov r2, #01h mov pl1, #12 </pre>

<pre> mov p0, a lcall delay inc dptr cjne a, #0ffh, lbl sjmp lb delay: mov r7, #25 nhan0: mov R6, #20 nhan1: mov r5, #250 djnz r5, \$ djnz r6, nhan1 djnz r7, nhan0 ret bang: DB 00000000B ; hay 00H DB 10000001B ; hay 81H DB 11000011B ; hay 0C3H DB 11100111B ; hay 0E7H DB 11111111B ; hay 0FFH DB 00011000B ; hay 18H DB 00111100B ; hay 3CH DB 01111110B ; hay 7EH DB 11111111B ; hay 0FFH DB 00000000B ; hay 00H end </pre>	<pre> mov pl3, r2 lapl4: acall delay mov a, r0 rl a mov r0, a add a, r2 mov r2, a mov pl3, a acall delay mov a, r2 cjne a, #0ffh, lapl4 endmac org 00h main: @sangdon 0, 00h, 2, 1 ; sáng cổng P2 @sangdon 2, 0ffh, 0, 2 ; sáng cổng P0 @sangdon 2, 00h, 0, 3 ; sáng cổng P0 @sangdon 0, 0ffh, 2, 4 ; sáng cổng P2 ljmp main delay: mov r7, #25 nhan0: mov r6, #20 nhan1: mov r5, #250 djnz r5, \$ djnz r6, nhan1 djnz r7, nhan0 ret end </pre>
--	---

Trong ba cách trên, cách đưa dữ liệu trực tiếp không thể dùng cho một bảng dữ liệu dài. Cách dùng bảng là thông dụng nhất, cho mỗi trường hợp, nhất là cách hiển thị khó, không có giải thuật xác định. Cách dùng lệnh với giải thuật đôi khi ngắn gọn. Hiển thị dồn (sang trái hay sang phải). Còn có giải thuật khác là dùng lệnh logic ORL giá trị đưa ra hiển thị cũ x0 và giá trị dịch một bit y0 (bởi lệnh SETB C và RLC A).

Đèn LED rời trên, ngoài chỉ thị giá trị của thanh ghi hay ô nhớ của VDK, còn hay dùng để quảng cáo hay điều khiển đèn giao thông tại các ngã đường. Khi đó, phải mắc đèn công suất lớn với các tầng khuếch đại công suất dùng tranzito công suất.

Các chương trình trên có thể thay đổi thời gian trễ Delay để thời gian sáng hay tắt tùy ý. Với đèn điều khiển giao thông, có lúc phải cho hai đèn sáng đồng thời (đỏ, vàng) để chỉ sự chuẩn bị chuyển luồng giao thông hay chuyển đèn (đỏ sang xanh hay xanh sang vàng rồi đỏ). Người ta còn đưa thời gian đếm ngược từ thời gian sáng tối đa đưa cho mỗi đèn về 0 và hiển thị số đếm dạng thập phân cho các đèn để người đi đường biết và chuẩn bị đi hay dừng tại các giao lộ (xem phần 11.2 cho hiển thị LED 7 đoạn).

11.1.5. Thiết kế mạch LED đơn và chạy chương trình trên mạch mô phỏng

Trong thiết kế và chạy mô phỏng sơ đồ điện tử, mới xuất hiện chương trình mô phỏng PROTEUS có cả mô phỏng mạch với vi điều khiển các loại như MCS-51 của INTEL, PIC của Microchip, AVR của ATMEL, MC68H11 của Motorola v.v... Chương trình trên còn mô phỏng được các mạch số và tương tự thay cho Workbench và Circuit maker hay Protel.

Nhược điểm cơ bản của PROTEUS là không hợp dịch được các ngôn ngữ khác nhau trên giao diện, mà chỉ chạy mô phỏng được chương trình dạng đuôi.HEX. Muốn chạy được mạch VDK các loại, ta phải:

- Thiết kế mạch nguyên lý với chương trình ISIS.
- Viết chương trình và hợp dịch thành file. HEX với các chương trình hợp dịch tương ứng với ngôn ngữ đó bằng cách vào tệp Labcenter/Proteus/Tool/ASEM51

Ví dụ: – PINNACLE cho họ MCS-51 của INTEL hay ASEM51.

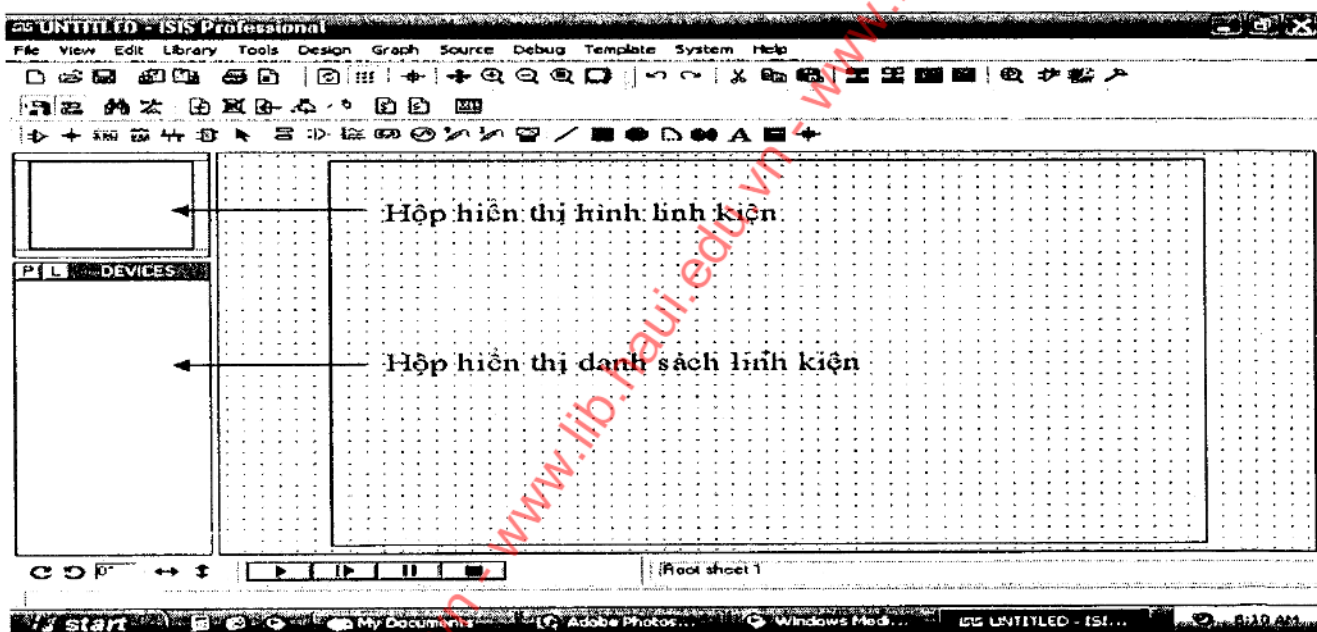
– AVR STUDIO cho AVR của ATMEL.

– MPASM cho PIC của Microchip.

Ngoài các hợp ngữ, còn các ngôn ngữ bậc cao (ngôn ngữ C, C++, PASCAL, C51 và R51 cũng dịch ra tệp đuôi. HEX để chạy trên PROTEUS.

1. Chương trình PROTEUS

PROTEUS có giao diện và 4 thanh công cụ như hình 11.3, 11.4 và 11.5.



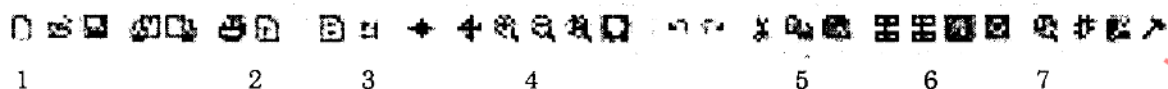
Hình 11.3. Giao diện của PROTEUS

Công dụng của các thanh công cụ như sau:

- Thanh công cụ số 1 dùng để tạo trang, lưu giữ, sao chép, phóng to, thu nhỏ, xem và in v.v... các trang thiết kế.
- Thanh công cụ số 2 dùng để chọn các vi mạch ra danh sách linh kiện, vẽ các đường dây, ghi các nhãn v.v...
- Thanh công cụ số 3 dùng để quay 0° , 90° , 180° các linh kiện và tên nhãn.
- Thanh công cụ số 4 dùng để chạy chương trình mô phỏng.

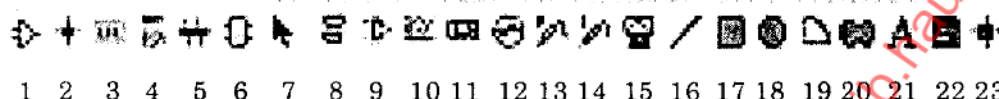
Tên và công dụng từng nút trong nhóm các thanh ghi trên trình bày ở các hình vẽ ở trên.

↓ Thanh số 1



- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
- Cụm 1: Tạo nhanh một trang thiết kế, mở trang thiết kế đã tạo, lưu trang thiết kế
 - Cụm 2: In cả trang, chọn vùng in tùy ý
 - Cụm 3: Chia lưới điểm trang thiết kế
 - Cụm 4: Zoom linh kiện về giữa trang, phóng to trang, thu nhỏ trang, fix trang, zoom vùng tùy chọn
 - Cụm 5: Cắt, copy, paste trang
 - Cụm 6: Copy, move, quay, xoá linh kiện trên trang
 - Cụm 7: Chỉ dùng biểu tượng đầu, chọn linh kiện

↓ Thanh số 2



- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
 - 11
 - 12
 - 13
 - 14
 - 15
 - 16
 - 17
 - 18
 - 19
 - 20
 - 21
 - 22
 - 23
- 1. Computer cho phép chọn linh kiện sau khi nhấp nút P ở hộp thoại DEVICES
 - 2. Điểm nối
 - 3. Tạo nhân của đường dây: nhấp trái chuột lên dây dẫn và đặt tên
 - 4. Ghi văn bản: nhấp trái chuột và viết
 - 5. Tạo bus: nhấp trái kéo đến điểm khác, lại nhấp trái rồi nhấp phải. Phải đặt label cho cùng tên cho các dây dẫn sử dụng chung bus
 - 6. Mạch phụ
 - 7. Nhấp trái lên linh kiện để xem Edit compouent
 - 8. Nguồn, đất, bus,...
 - 9. Chân linh kiện
 - 10. Simulation Graph: mô phỏng hình vẽ
 - 11. Tape recorder, ghi băng từ
 - 12. Generator: DC, xung, sin,...
 - 13. Voltage Probe, giống von kế thường nhưng chỉ có 1 đầu dây, hiện số trực tiếp
 - 14. Curent Probe, như trên nhưng cho giá trị dòng điện
 - 15. Dụng cụ ảo: von kế AC, DC, ampe kế AC, DC,...
 - 16. Vẽ đường 2D
 - 17. Vẽ hộp 2D
 - 18. Vẽ đường tròn 2D
 - 19. Vẽ cung tròn 2D
 - 20. Vẽ đường cong 2D
 - 21. Text 2D, nhấp trái vào trang và type
 - 22. Ký hiệu hình vẽ 2D
 - 23. Đánh dấu cho gốc linh kiện

↓ Thanh số 3



Quay trái, phải, đối xứng chân ngang dọc linh kiện trên hộp hiển thị linh kiện

Hình 11.4. Các thanh công cụ 1, 2 và 3 của giao diện PROTEUS