**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

****

**BÁO CÁO THỰC HÀNH HỆ THỐNG NHÚNG**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:** | ThS. Mai Thế Anh |
| **Sinh viên thực hiện:**  **MSSV :**  **Lớp :** | Chu Quốc Hội  215752020710037  62K KT-ĐTVT |
|  |  |

**Nghệ An–****2024**

**MỤC LỤC**

[BÀI 1. LẬP TRÌNH C VỚI VI ĐIỀU KHIỂN 8051 5](#_Toc185188422)

[1.1. Mục đích 5](#_Toc185188423)

[1.2. Cơ sở lý thuyết 5](#_Toc185188424)

[1.2.1. Phần mềm MikroC cho 8051 5](#_Toc185188425)

[1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 5](#_Toc185188426)

[1.3.1. Thiết bị thực hành thí nghiệm 5](#_Toc185188427)

[1.4. Các bước tiến hành thí nghiệm 5](#_Toc185188428)

[1.4.1. Bài tập thực hành 5](#_Toc185188429)

[BÀI 2: GIAO TIẾP VỚI MA TRẬN BÀN PHÍM VÀ MÀN HÌNH LCD 16x2 8](#_Toc185188430)

[2.1. Mục đích 8](#_Toc185188431)

[2.2. Cơ sở lý thuyết 8](#_Toc185188432)

[2.2.1. Ma trận bàn phím 8](#_Toc185188433)

[2.2.2. Làm việc với LCD 16x2 9](#_Toc185188434)

[2.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 9](#_Toc185188435)

[2.3.1. Thiết bị thực hành thí nghiệm 9](#_Toc185188436)

[2.4. Các bước tiến hành thí nghiệm 9](#_Toc185188437)

[2.4.1. Giao tiếp với ma trận phím và màn hình LCD 16x2 9](#_Toc185188438)

[2.4.2. Bài tập thực hành 10](#_Toc185188439)

[BÀI 3: GIAO TIẾP VỚI CÁC ĐÈN LED 7 THANH 12](#_Toc185188440)

[3.1. Mục đích 12](#_Toc185188441)

[3.2. Cơ sở lý thuyết 12](#_Toc185188442)

[3.2.1. Giới thiệu về đèn LED 7 thanh 12](#_Toc185188443)

[3.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 13](#_Toc185188444)

[3.3.1. Thiết bị thực hành thí nghiệm 13](#_Toc185188445)

[3.4. Các bước tiến hành thí nghiệm 13](#_Toc185188446)

[3.4.1. Lập trình giao tiếp với các đèn LED 7 thanh 13](#_Toc185188447)

[3.4.2. Bài tập thực hành 14](#_Toc185188448)

[BÀI 4: GIAO TIẾP VỚI CẢM BIẾN NHIỆT ĐỘ 15](#_Toc185188449)

[4.1. Mục đích 15](#_Toc185188450)

[4.2. Cơ sở lý thuyết 15](#_Toc185188451)

[4.2.1. Cảm biến nhiệt độ DS1820 15](#_Toc185188452)

[4.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 16](#_Toc185188453)

[4.3.1. Thiết bị thực hành thí nghiệm 16](#_Toc185188454)

[4.4. Các bước tiến hành thí nghiệm 16](#_Toc185188455)

[4.4.1. Lập trình giao tiếp với cảm biến nhiệt độ DS1820 16](#_Toc185188456)

[4.4.2. Bài tập thực hành 17](#_Toc185188457)

[BÀI 5: LẬP TRÌNH SỬ DỤNG NGẮT NGOÀI 19](#_Toc185188458)

[5.1. Mục đích 19](#_Toc185188459)

[5.2. Cơ sở lý thuyết 19](#_Toc185188460)

[5.2.1. Ngắt ngoài của 8051 19](#_Toc185188461)

[5.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 20](#_Toc185188462)

[5.3.1. Thiết bị thực hành thí nghiệm 20](#_Toc185188463)

[5.4. Các bước tiến hành thí nghiệm 20](#_Toc185188464)

[5.4.1. Hiển thị số lần ngắt ngoài trên chân P3.2 lên LCD 16x2 20](#_Toc185188465)

[5.4.2. Bài tập thực hành 21](#_Toc185188466)

[BÀI 6: HỆ ĐIỀU HÀNH THỜI GIAN THỰC 22](#_Toc185188467)

[6.1. Mục đích 22](#_Toc185188468)

[6.2. Cơ sở lý thuyết 22](#_Toc185188469)

[6.2.1. Hệ điều hành thời gian thực 22](#_Toc185188470)

[6.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao 24](#_Toc185188471)

[6.3.1. Thiết bị thực hành thí nghiệm 24](#_Toc185188472)

[6.4. Các bước tiến hành thí nghiệm 24](#_Toc185188473)

BÀI 1. LẬP TRÌNH C VỚI VI ĐIỀU KHIỂN 8051

1.1. Mục đích

- Làm quen với phần mềm MikroC cho 8051.

- Tìm hiểu và viết chương trình điều khiển cơ bản với C/C++.

1.2. Cơ sở lý thuyết

1.2.1. Phần mềm MikroC cho 8051

Là một công cụ lập trình cho 8051 được công ty MikroElectronic phát triển. Gói công cụ Mikro bao gồm các công cụ lập trình cho: 8051, AVR, PIC và mới nhất là ARM : STM32F của ST và Stelaris, Tiva của TI.

Ưu điểm của gói công cụ MikroC 8051 là:

- Giao diện trực quan

- Rất dễ sử dụng đặc biệt là với những người mới làm quen với việc lập trình cho vi điều khiển 8051

- Hỗ trợ thư viện hầu hết cho các ứng dụng như: Uart, Spi, i2c, onewrite,..

Nhược điểm:

- Người sử dụng gặp khó khăn trong việc tìm hiểu cấu trúc của hàm library và khó thay đổi được nó.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

1.3.1. Thiết bị thực hành thí nghiệm

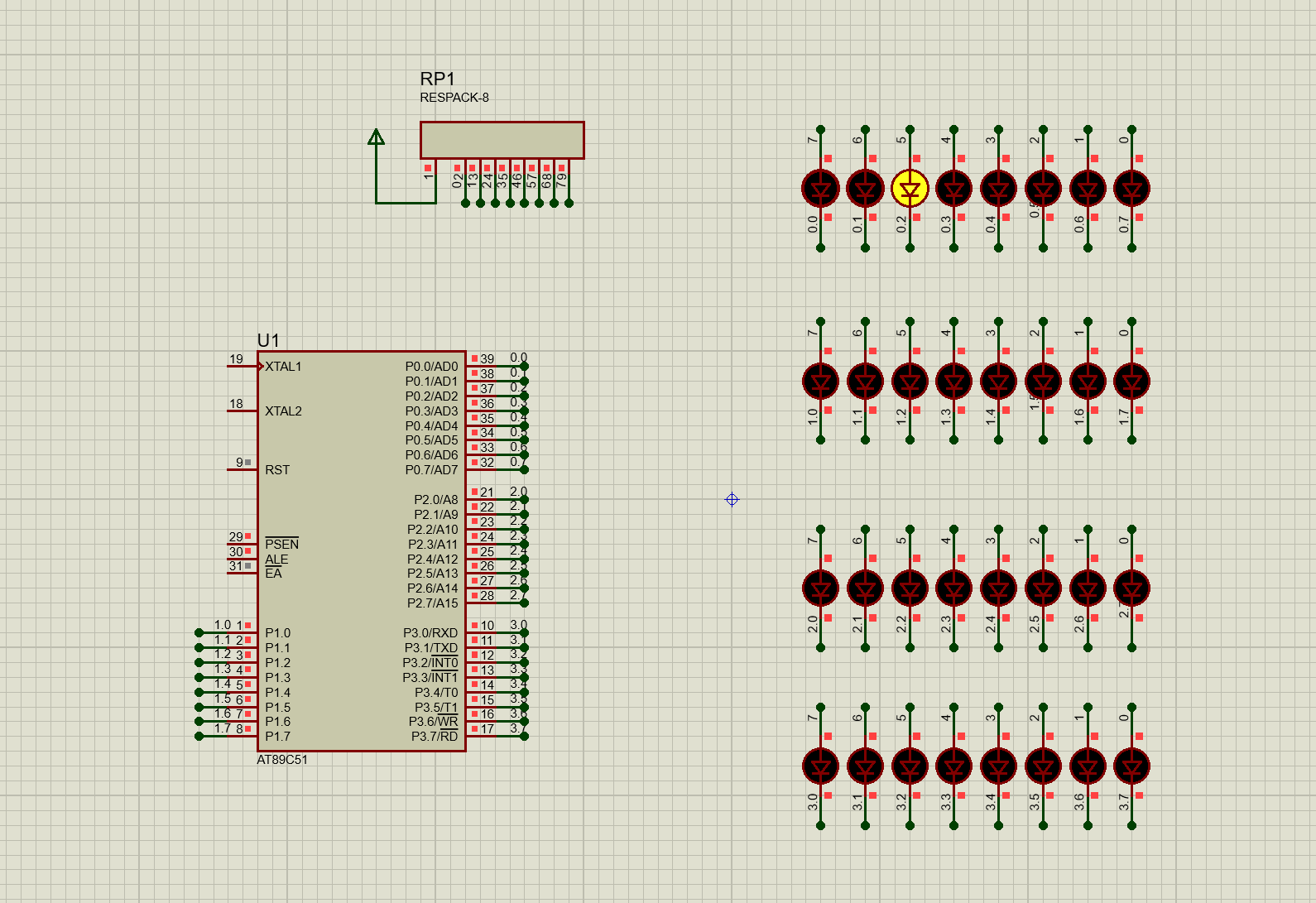
- Máy tính cài đặt phần mềm MikroC for 8051

1.4. Các bước tiến hành thí nghiệm

1.4.1. Bài tập thực hành

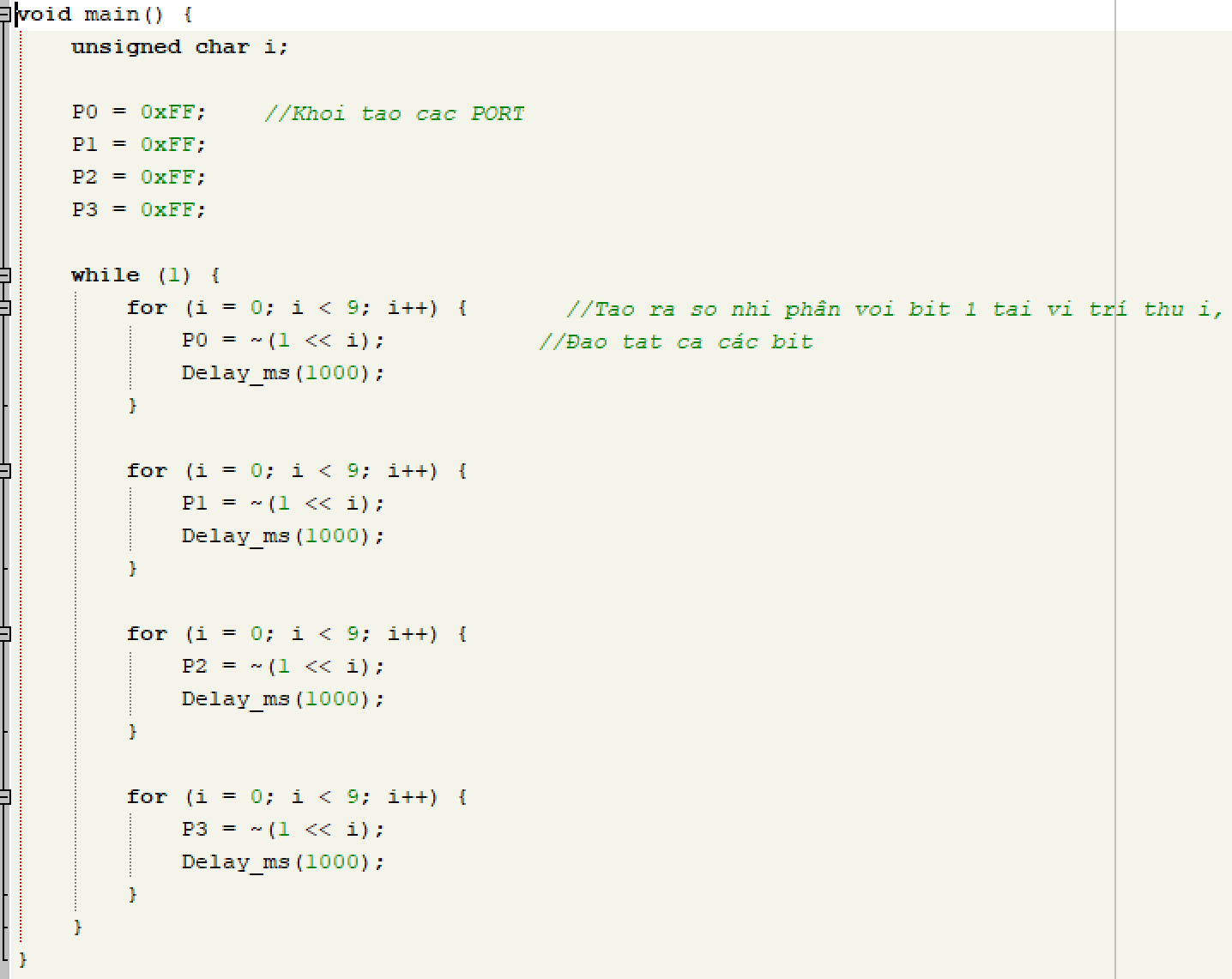
1. Viết lại chương trình điều khiển Led đơn nói trên với các hiệu ứng led đuổi nhau từ P0.0 🠒 P0.7 🠒 P1.0 🠒 P1.7 🠒 P2.0 🠒 P2.7 🠒 P3.0 🠒 P3.7 sử dụng các lệnh dịch bit. Thời gian chuyển trạng thái của mỗi đèn LED là 1 giây. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng.

**- Mô phỏng trên Proteus**

****

Hình 1.1. Hình ảnh mô phỏng trên proteus

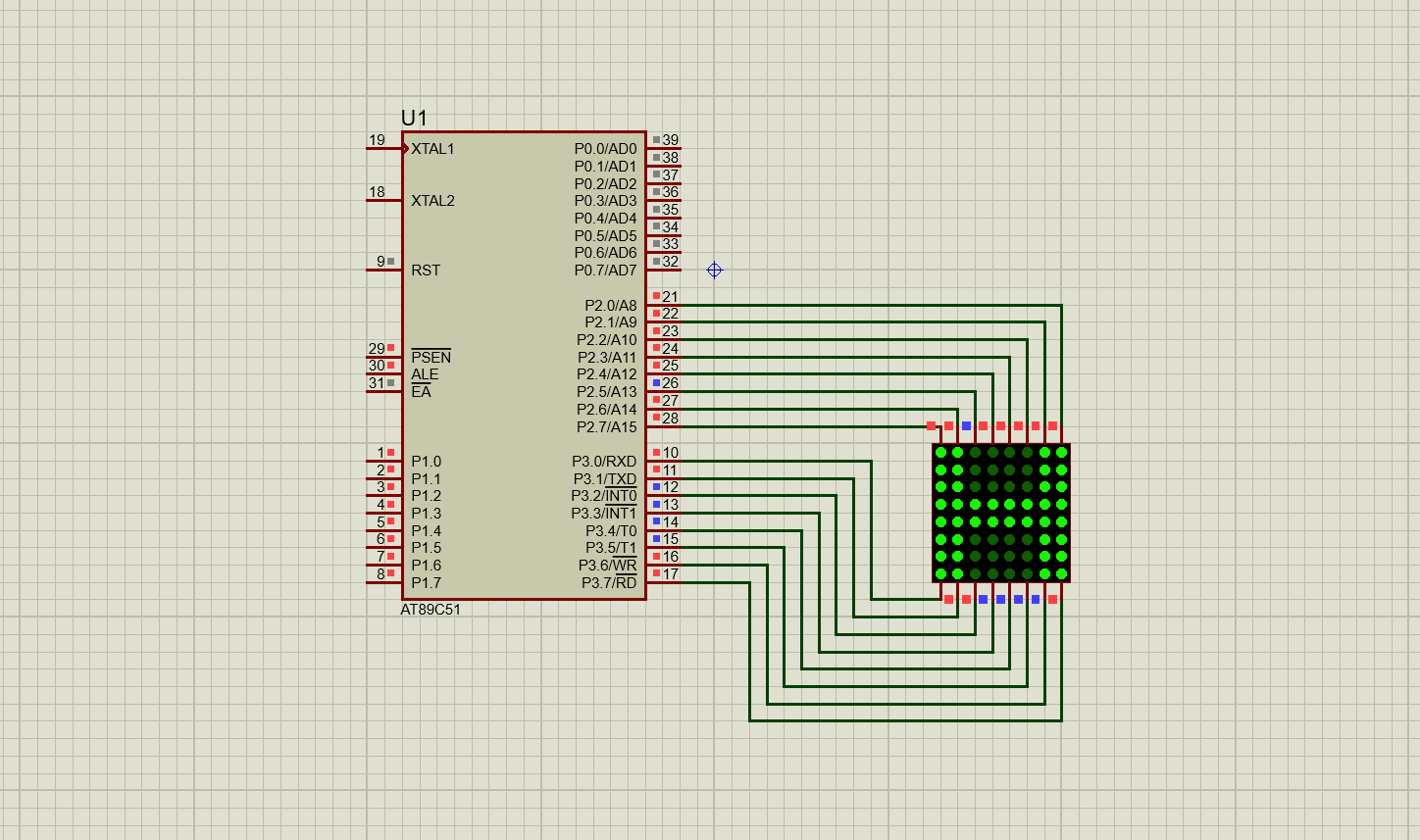
**- Code chương trình**

****

Hình 1.2. Hình ảnh code chương trình

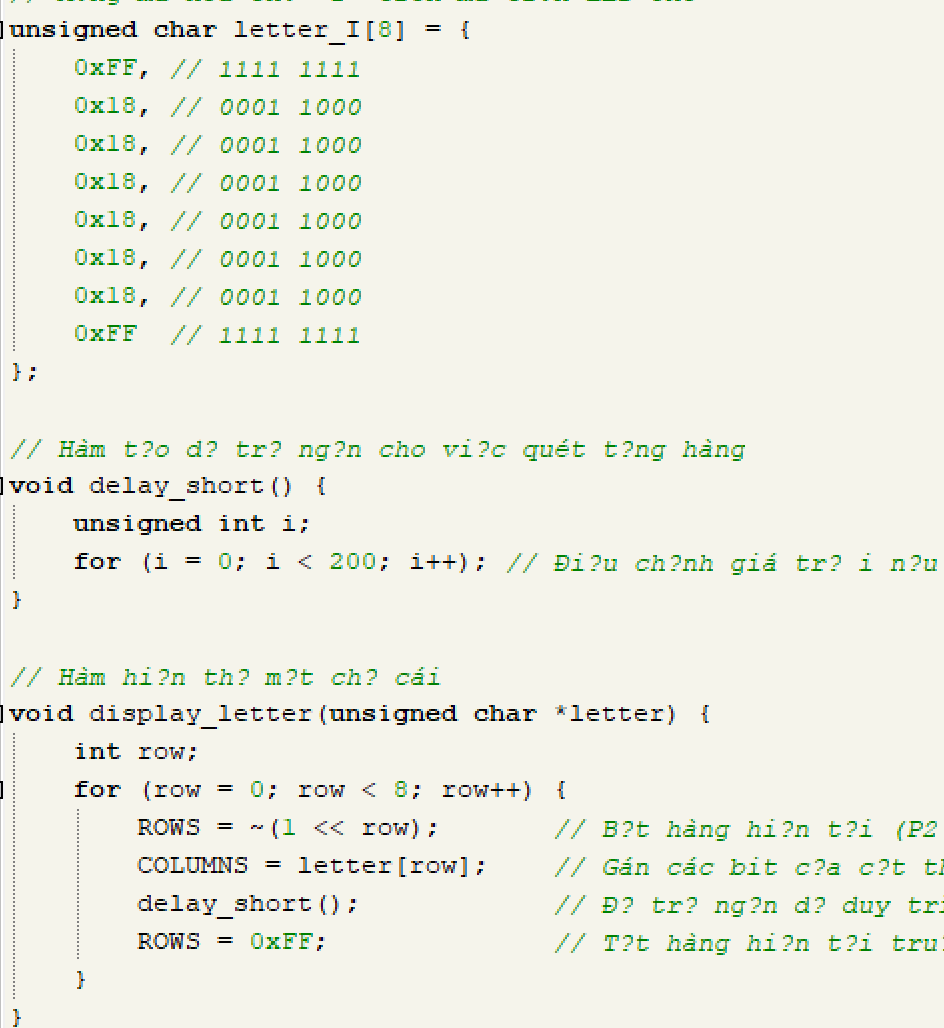
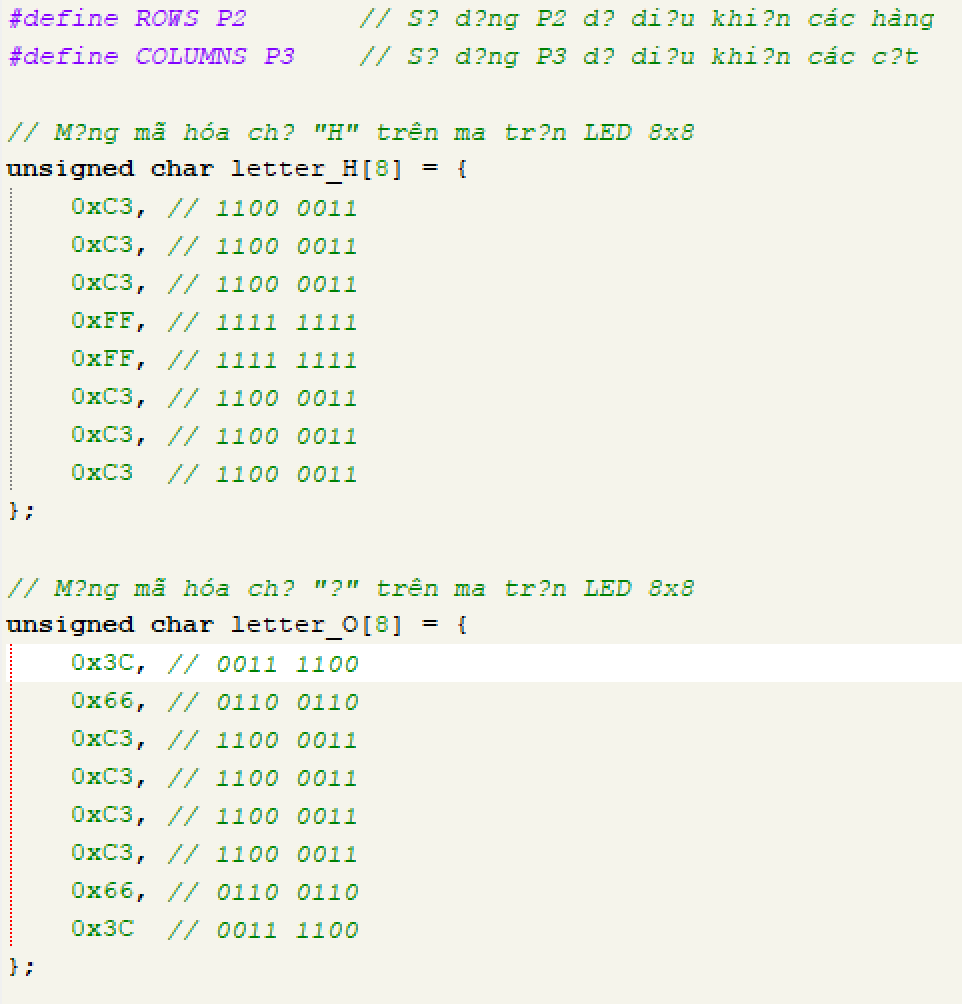
2. Sử dụng ma trận đèn LED 4 × 8 tạo bởi các port P0, P1, P2, P3 để hiện tên của bản thân, thời gian chuyển trạng thái của mỗi ký tự là 500ms. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**



Hình 1.3. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**

****

Hình 1.4. Hình ảnh code chương trình

BÀI 2: GIAO TIẾP VỚI MA TRẬN BÀN PHÍM VÀ MÀN HÌNH LCD 16x2

2.1. Mục đích

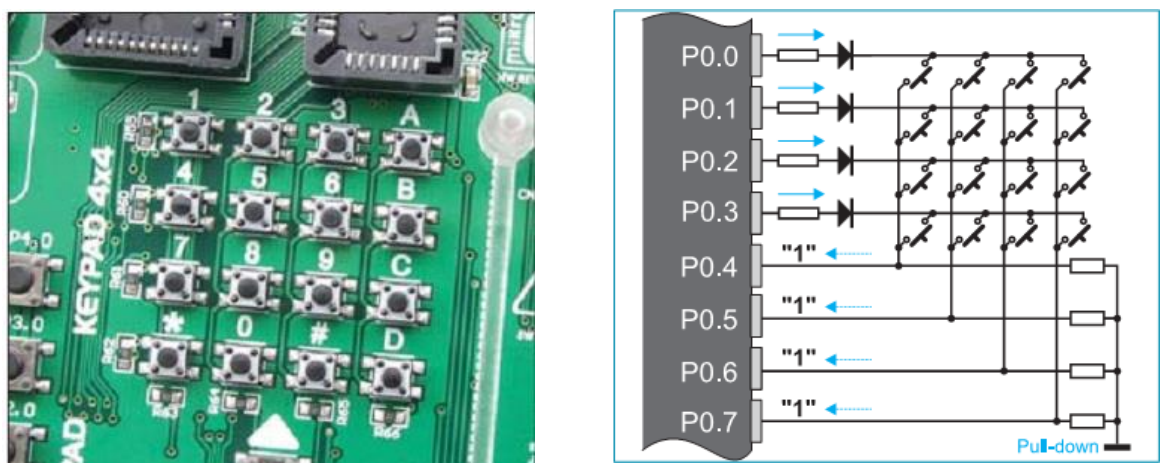
- Tìm hiểu và viết chương trình giao tiếp với keypad 4x4;

- Hiển thị trạng thái phím bấm lên LCD 16x2.

2.2. Cơ sở lý thuyết

2.2.1. Ma trận bàn phím

Trong các hệ thống sử dụng vi điều khiển, bàn phím, LCD là công cụ giúp cho việc giao tiếp giữa người và vi xử lý.



Hình 2.1. Ma trận bàn phím 4x4

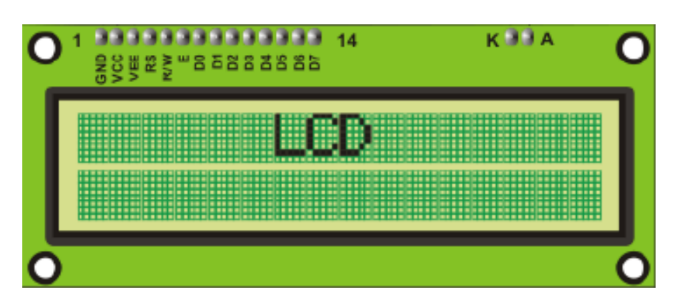
Ma trận bàn phím kết nối với vi điều khiển thông qua PORT0, nguyên tắc hoạt động của ma trận phím là: kích lần lượt các chân P0.0, P0.1, P0.2, P0.3 và đọc trạng thái của các phím bấm thông qua P0.4, P0.5, P0.6, P0.7.

Trong mikroC cung cấp các thư viện hàm giao tiếp với keypad 4x4. Tham khảo trong phần help của MikroC.

2.2.2. Làm việc với LCD 16x2



Hình 2.2. Màn hình LCD 16x2



Hình 2.3. Các chân chức năng trên màn hình LCD 16x2

2.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

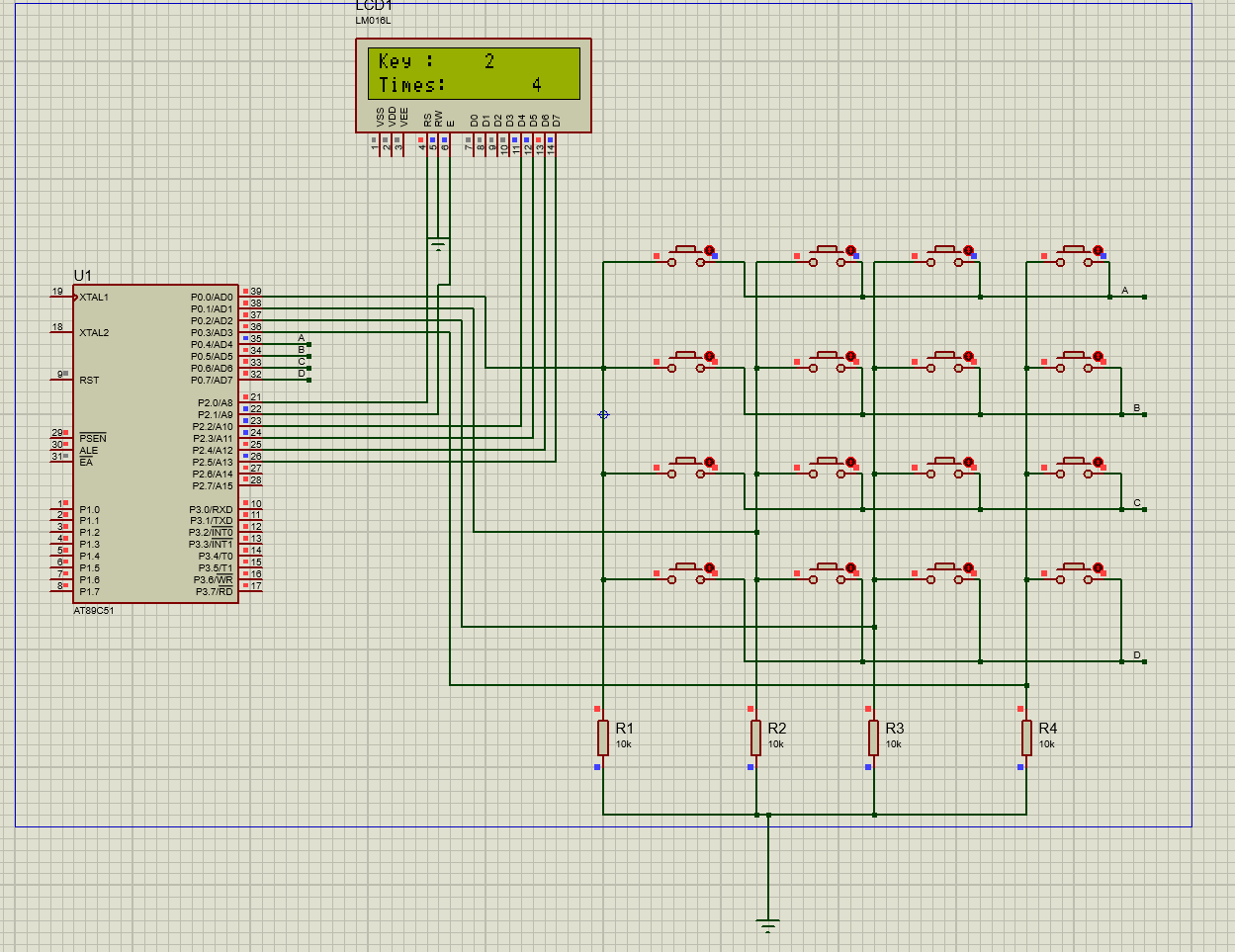
2.3.1. Thiết bị thực hành thí nghiệm

- Máy tính cài đặt phần mềm MikroC for 8051.

2.4. Các bước tiến hành thí nghiệm

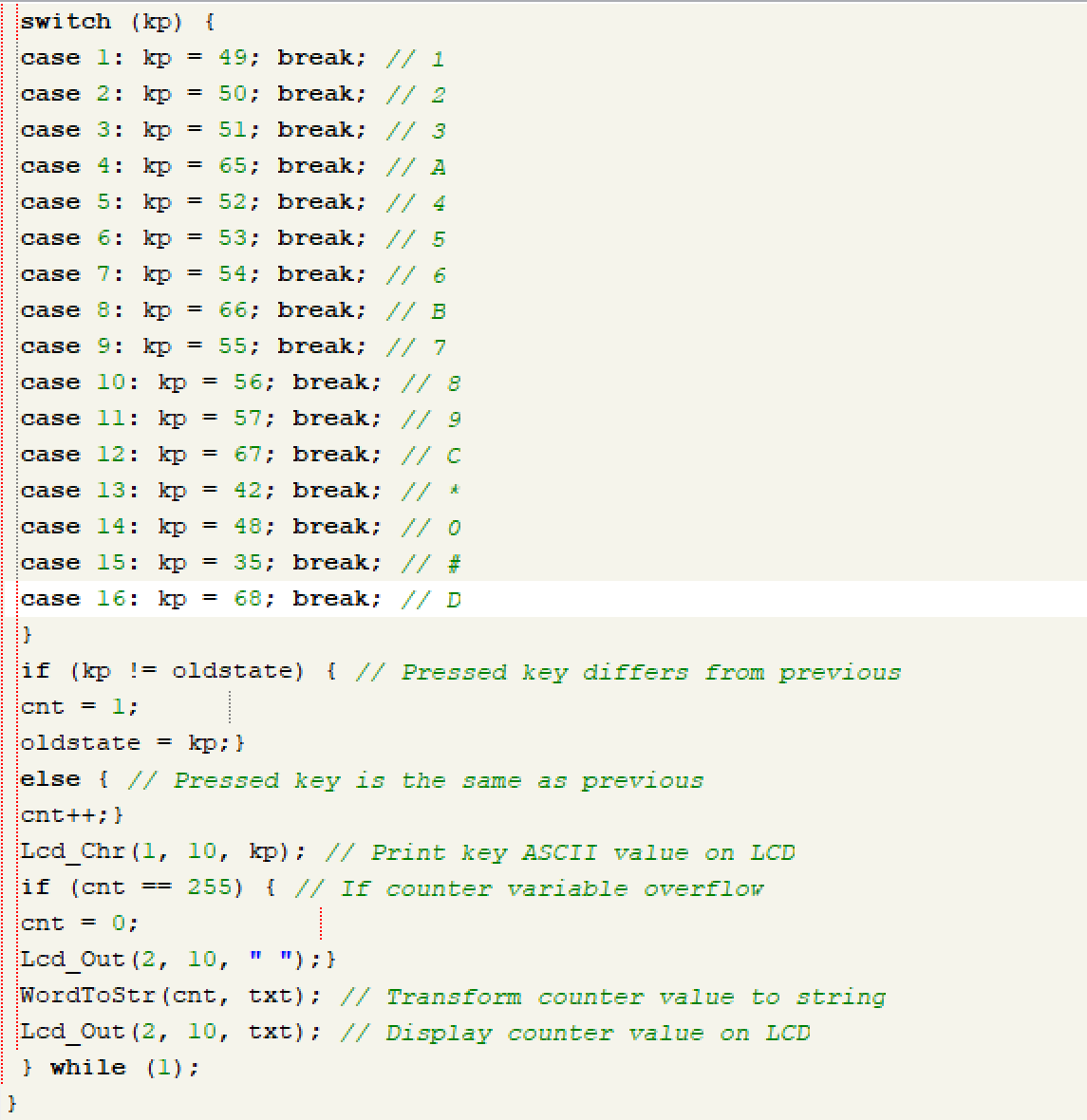
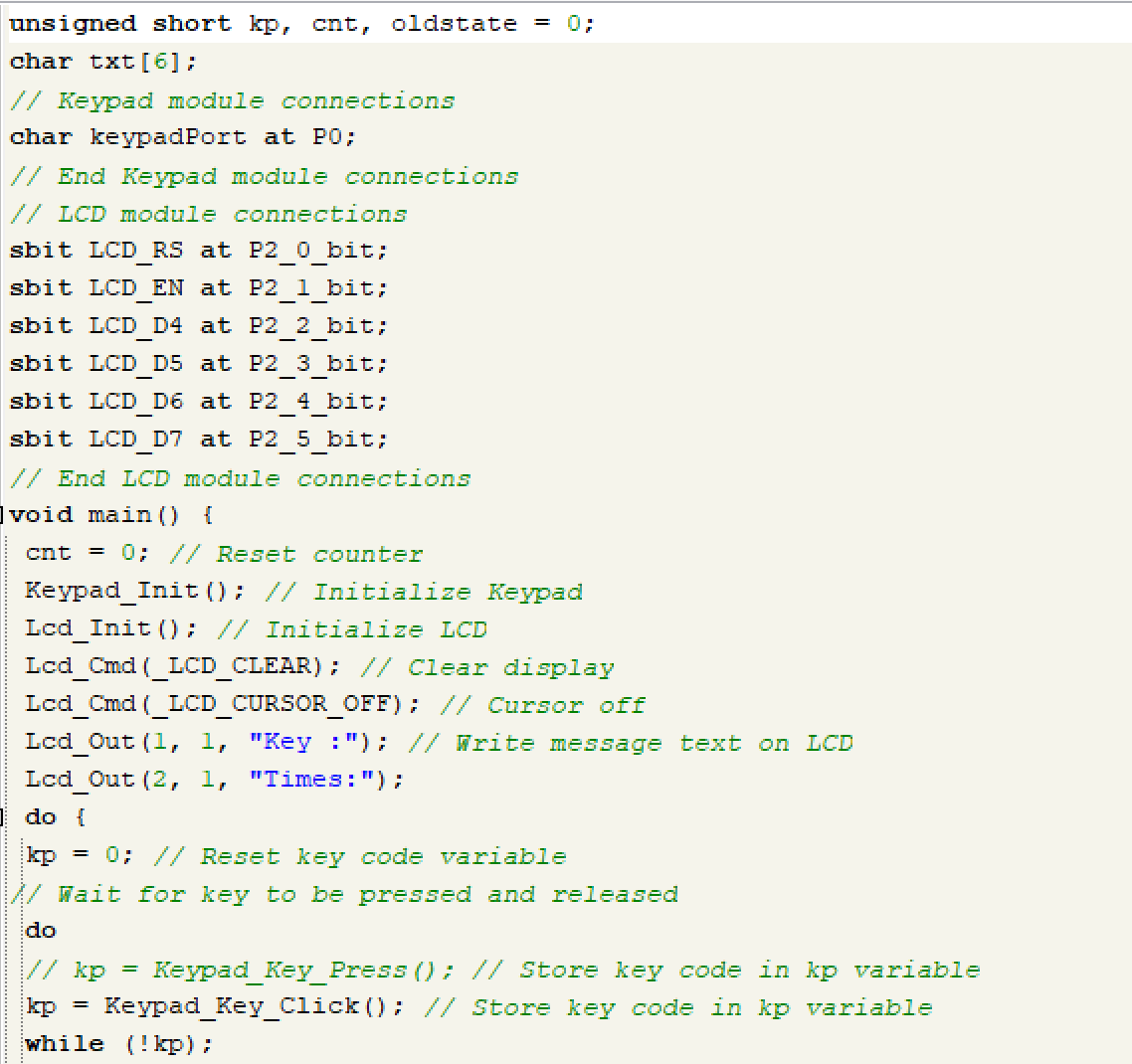
2.4.1. Giao tiếp với ma trận phím và màn hình LCD 16x2

**- Mô phỏng trên Proteus**

****

Hình 2.4. Hình ảnh mô phỏng phần cứng trên Proteus

**- Code chương trình**

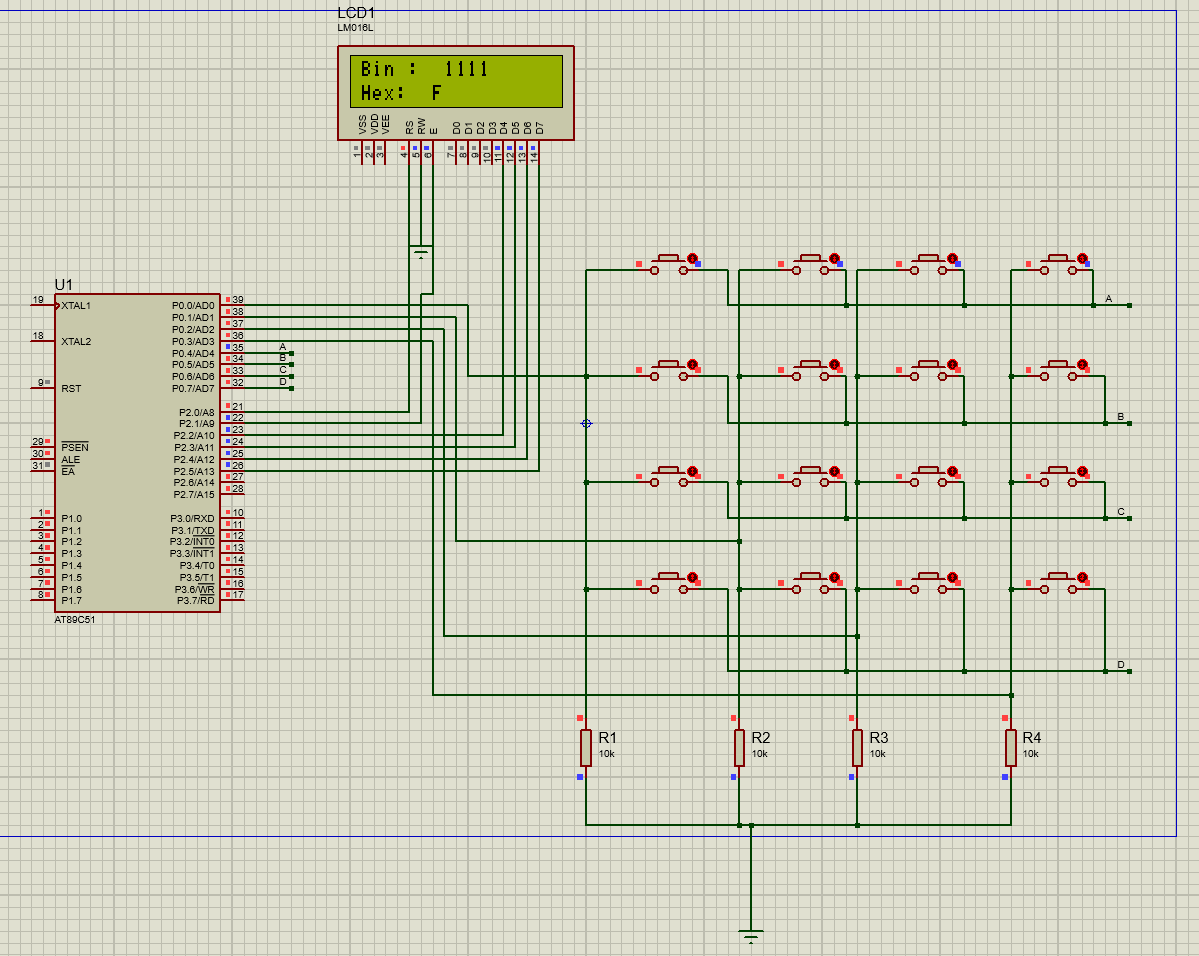
****

Hình 2.5. Hình ảnh code chương trình

2.4.2. Bài tập thực hành

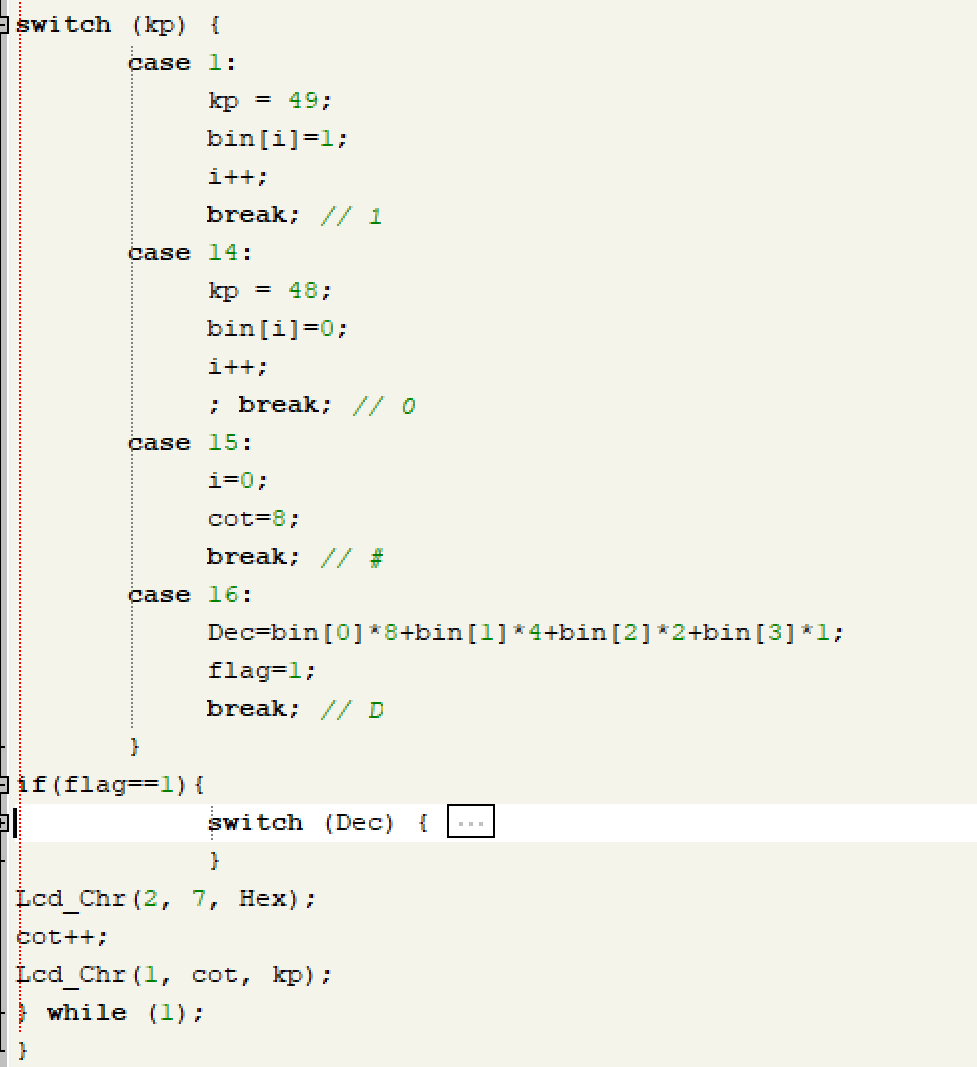
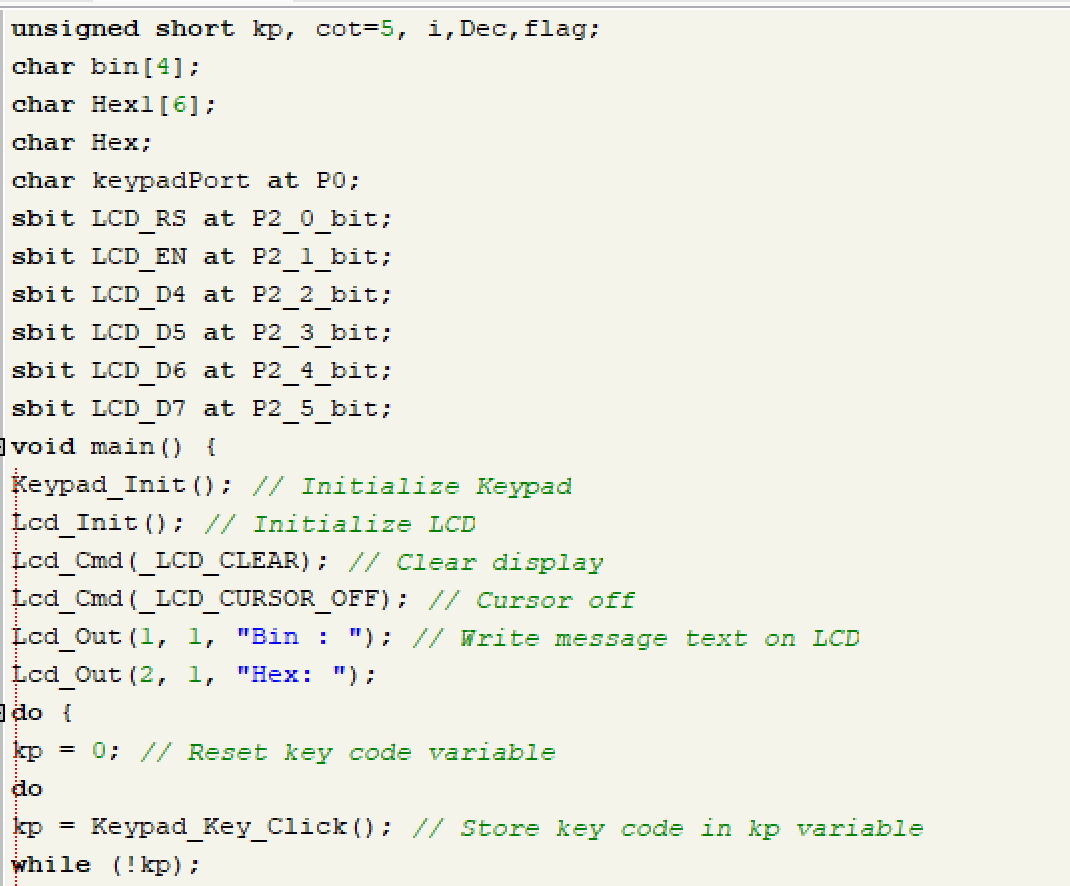
1. Viết chương trình nhập vào một số nhị phân 4bit và chuyển đổi số nhị phân đó sang số hexa hiển thị lên màn hình LCD. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**

****

Hình 2.6. Hình ảnh mô phỏng trên proteus

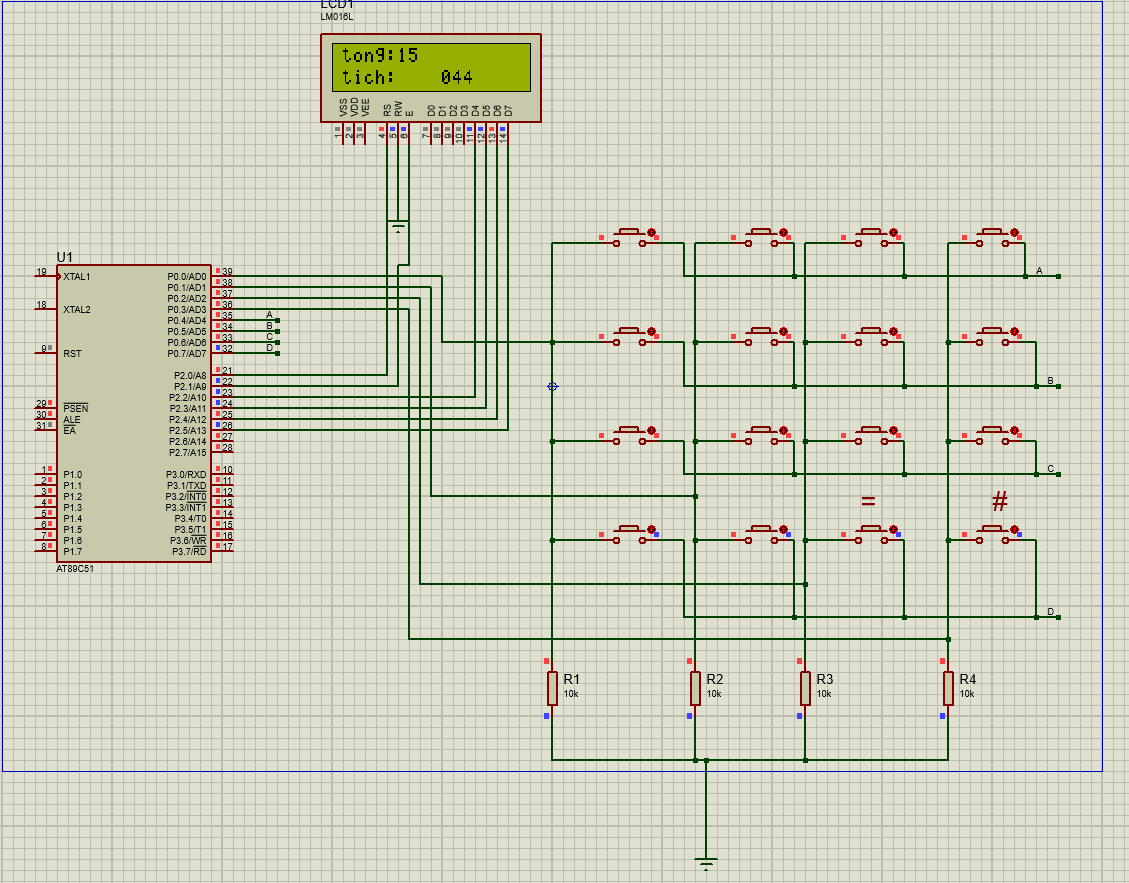
**- Code chương trình**

****

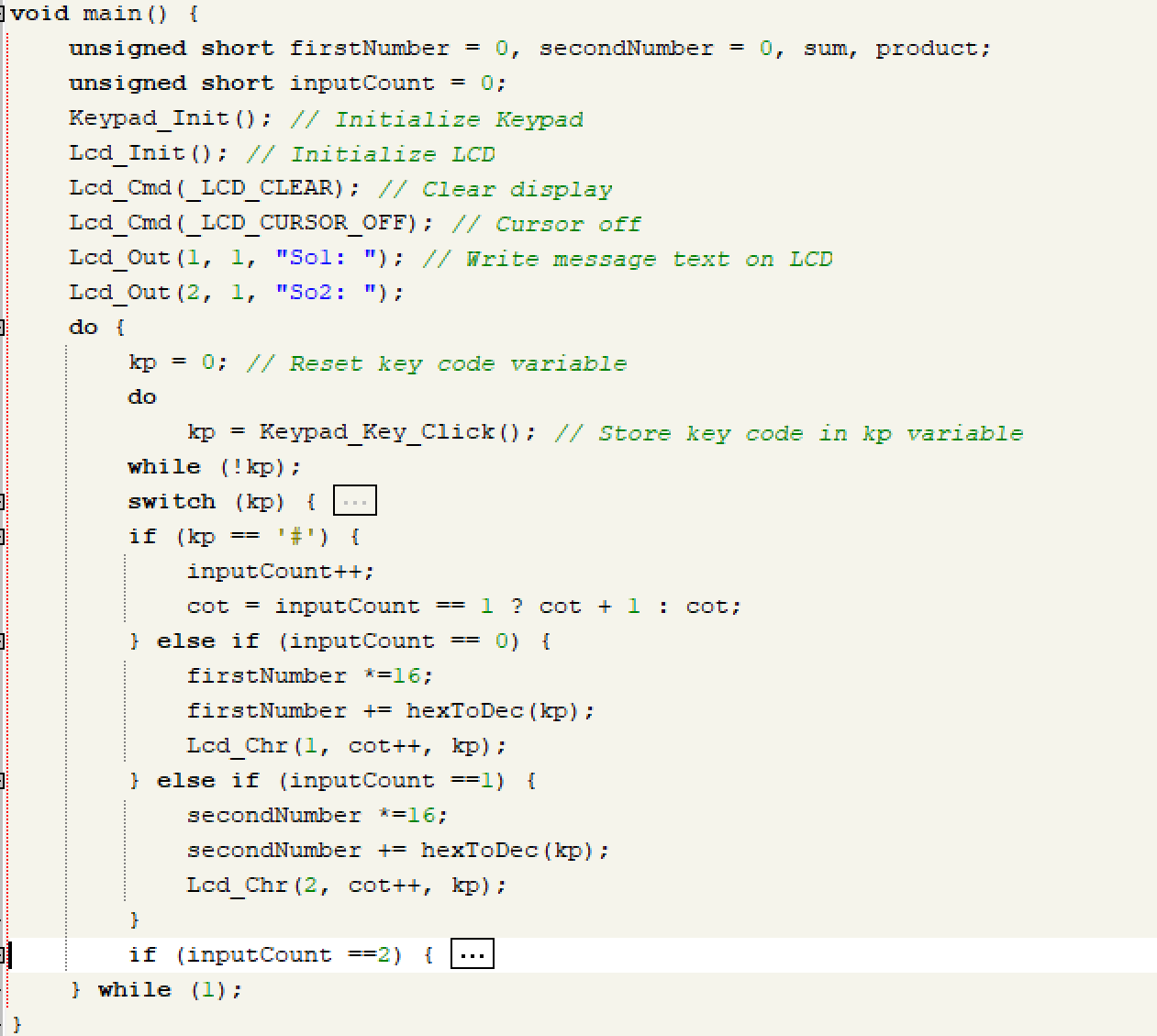
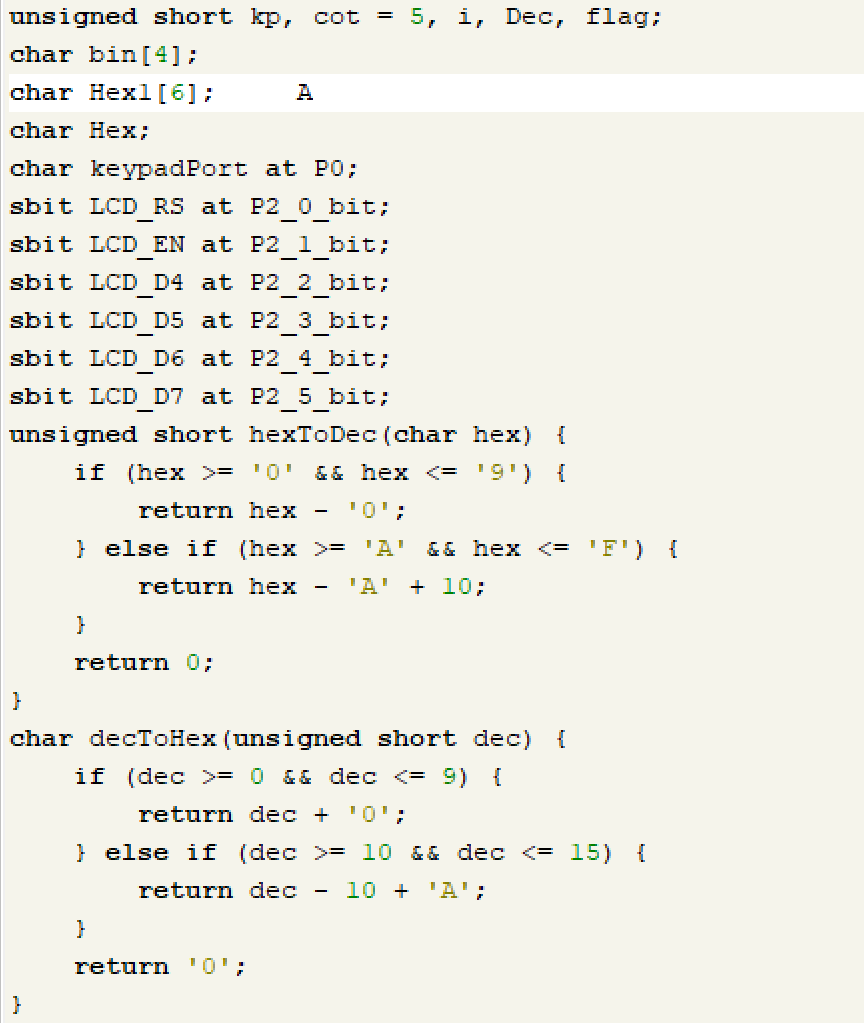
Hình 2.7. Hình ảnh code chương trình

2. Viết chương trình nhập vào hai số hexa và tính tổng, tích của hai số hiển thị hai số đó lên màn hình LCD. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**

****

Hình 2.8. Hình ảnh mô phỏng trên Proteus



Hình 2.9. Hình ảnh code chương trình

BÀI 3: GIAO TIẾP VỚI CÁC ĐÈN LED 7 THANH

3.1. Mục đích

- Tìm hiểu về Led 7 thanh và phương pháp điều khiển

- Lập trình ứng dụng hiển thị Led 7 thanh.

3.2. Cơ sở lý thuyết

3.2.1. Giới thiệu về đèn LED 7 thanh

Led 7 thanh là thiết bị chuyên dùng để hiển thị số 0 -> 9.

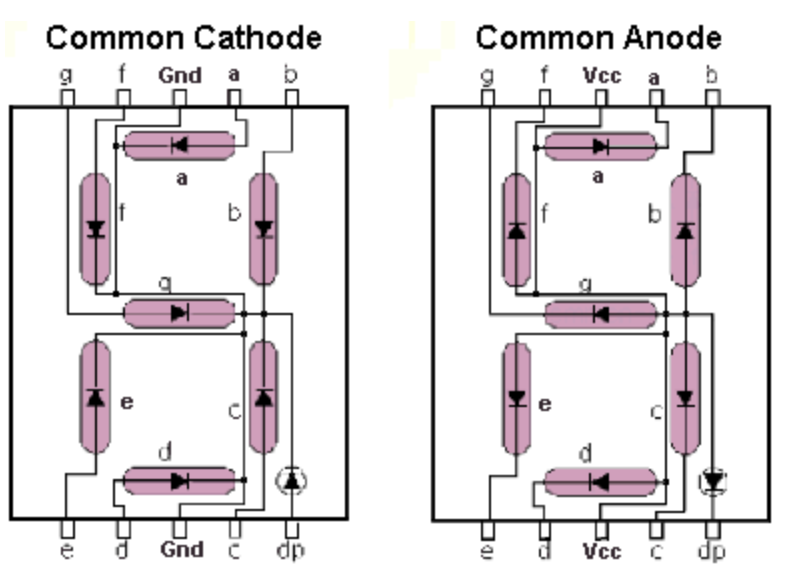
Có 2 loại Led 7 thanh:

- Led 7 thanh Anot chung

Loại dương chung (Common Anode): Cực dương (anode) của tất cả 8 Led được nối với nhau và các cực âm (cathode) đứng riêng lẻ.

- Led 7 thanh Katot chung

Loại âm chung (Common Cathode): Cực âm (cathode) của tất cả 8 Led được nối với nhau và các cực dương (anode) đứng riêng lẻ.



Hình 3.1. Cấu tạo bên trong của Led 7 thanh

3.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

3.3.1. Thiết bị thực hành thí nghiệm

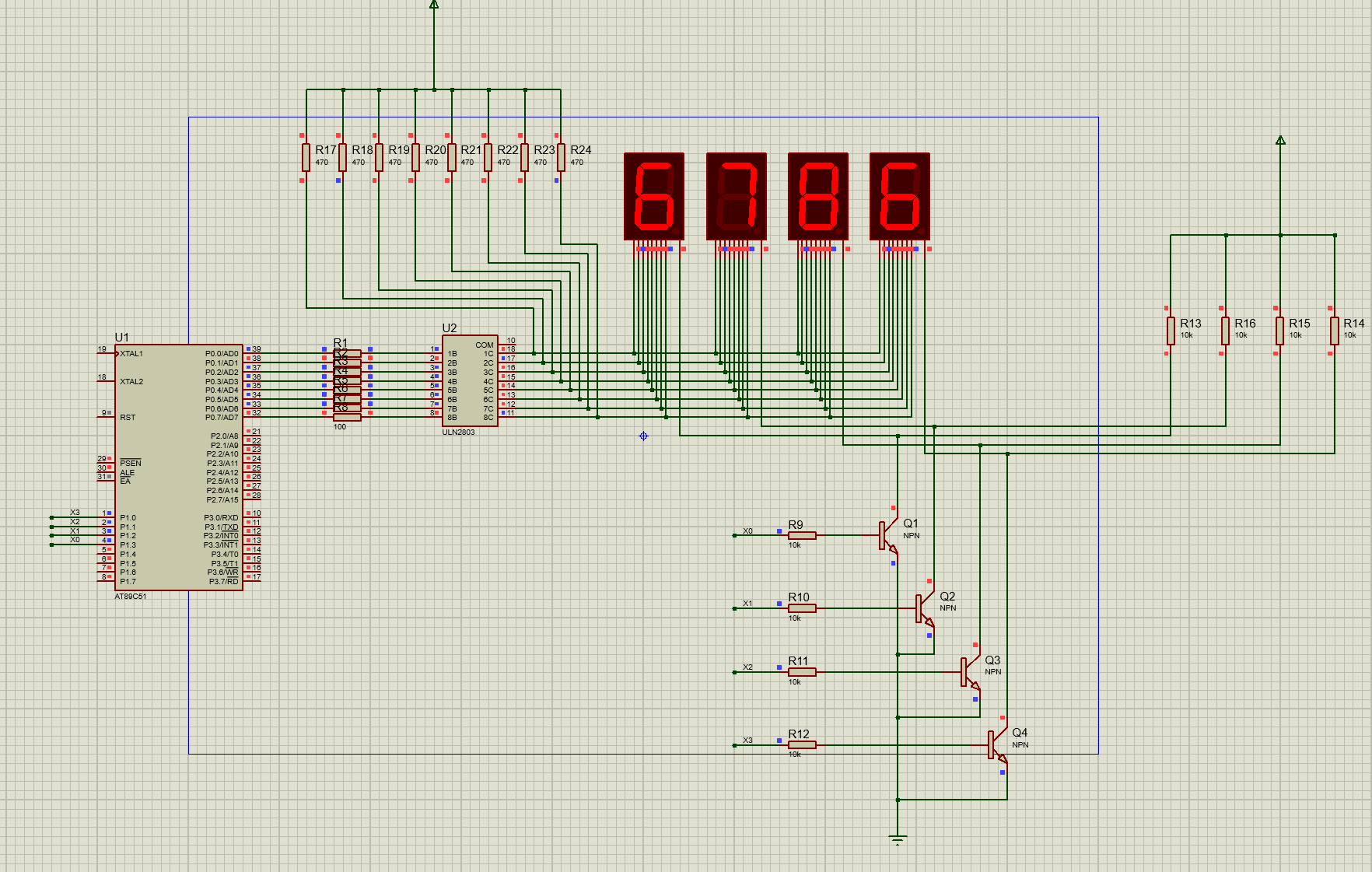
- Máy tính cài đặt phần mềm MikroC for 8051.

3.4. Các bước tiến hành thí nghiệm

3.4.1. Lập trình giao tiếp với các đèn LED 7 thanh

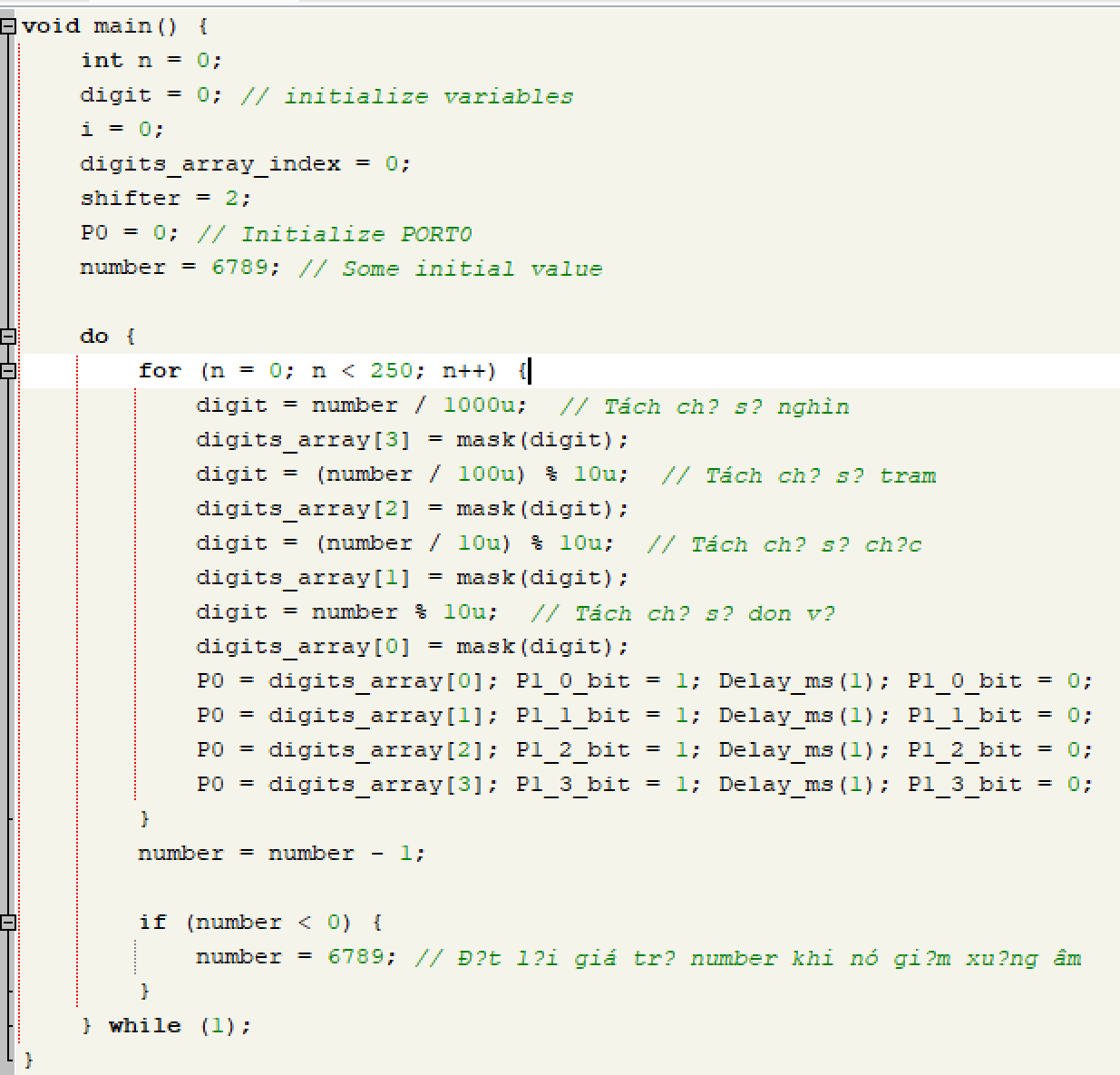
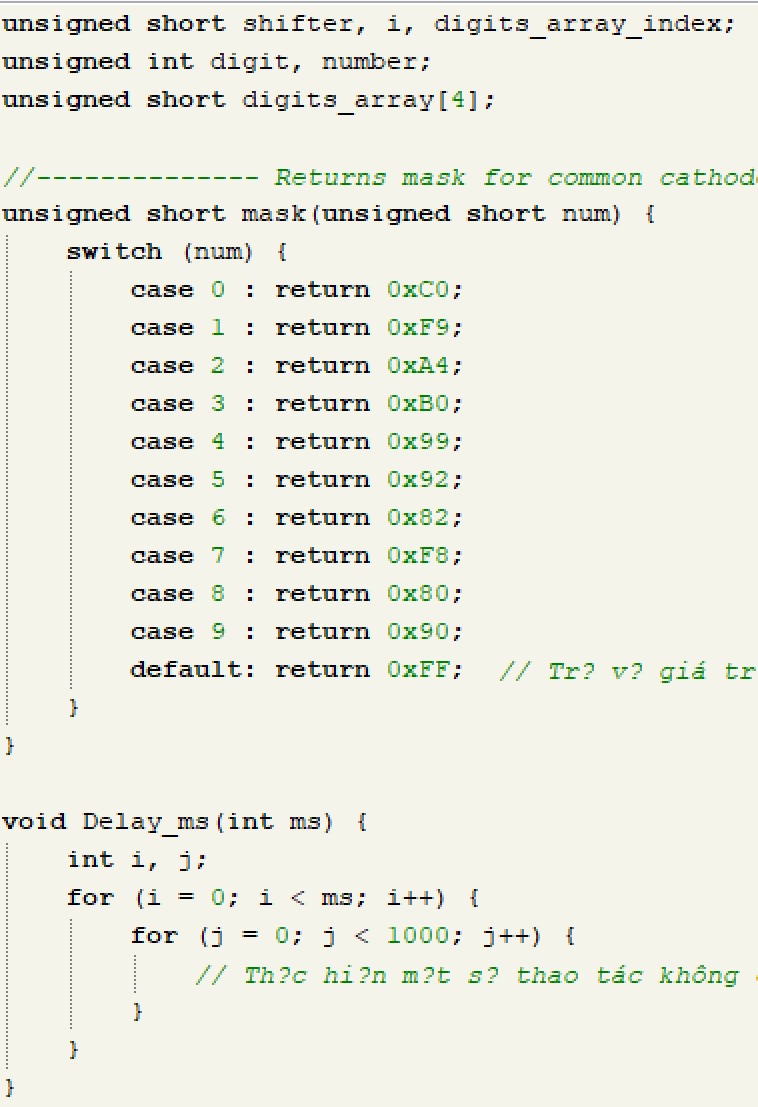
Viết chương trình thực hiện hiển thị số có 4 chữ số lên đèn LED 7 thanh và đếm lùi về 0000. Vẽ sơ đồ nguyên lý trên proteus

**- Mô phỏng trên Proteus**



Hình 3.2. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**

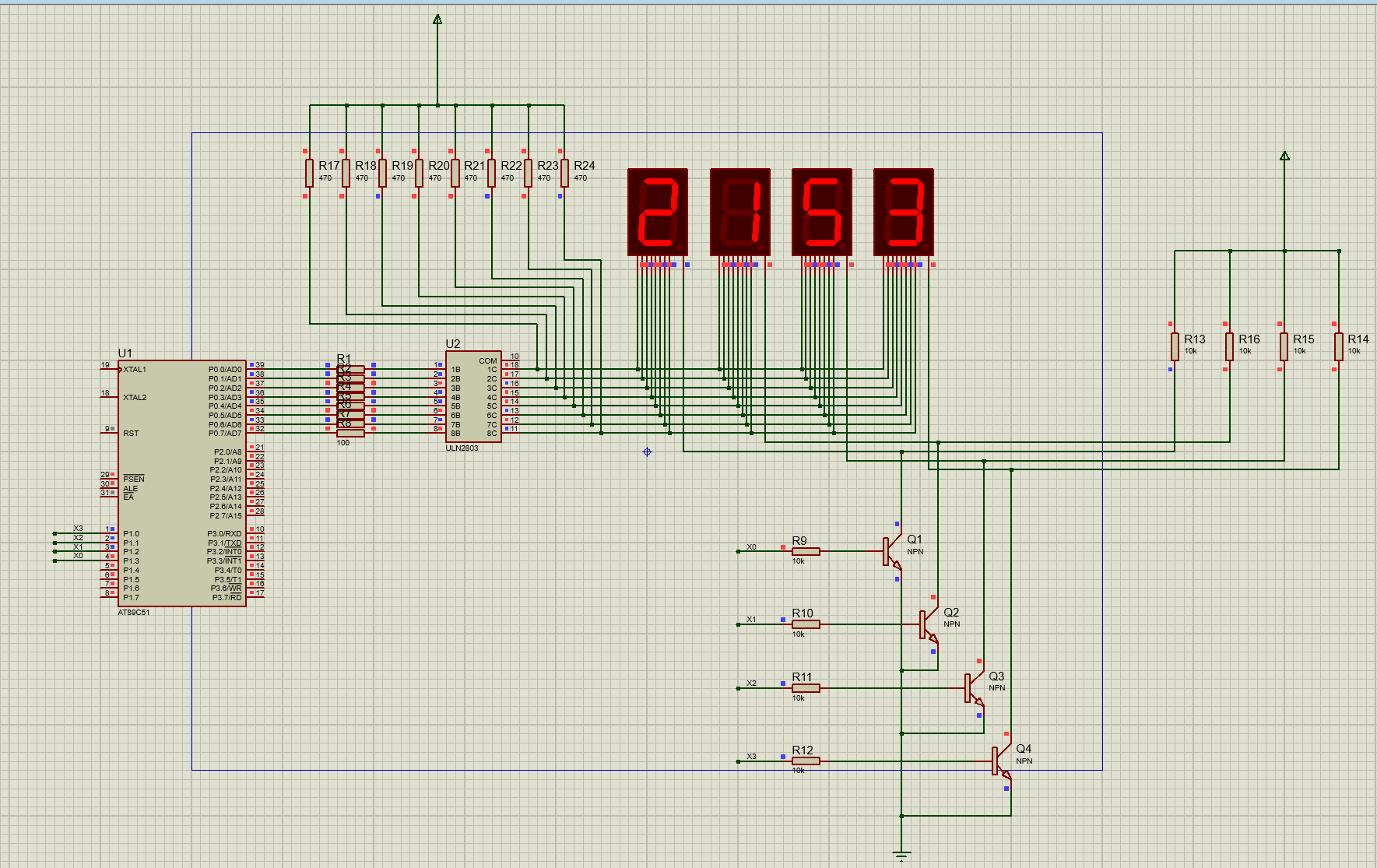
****

Hình 3.3. Hình ảnh code chương trình

3.4.2. Bài tập thực hành

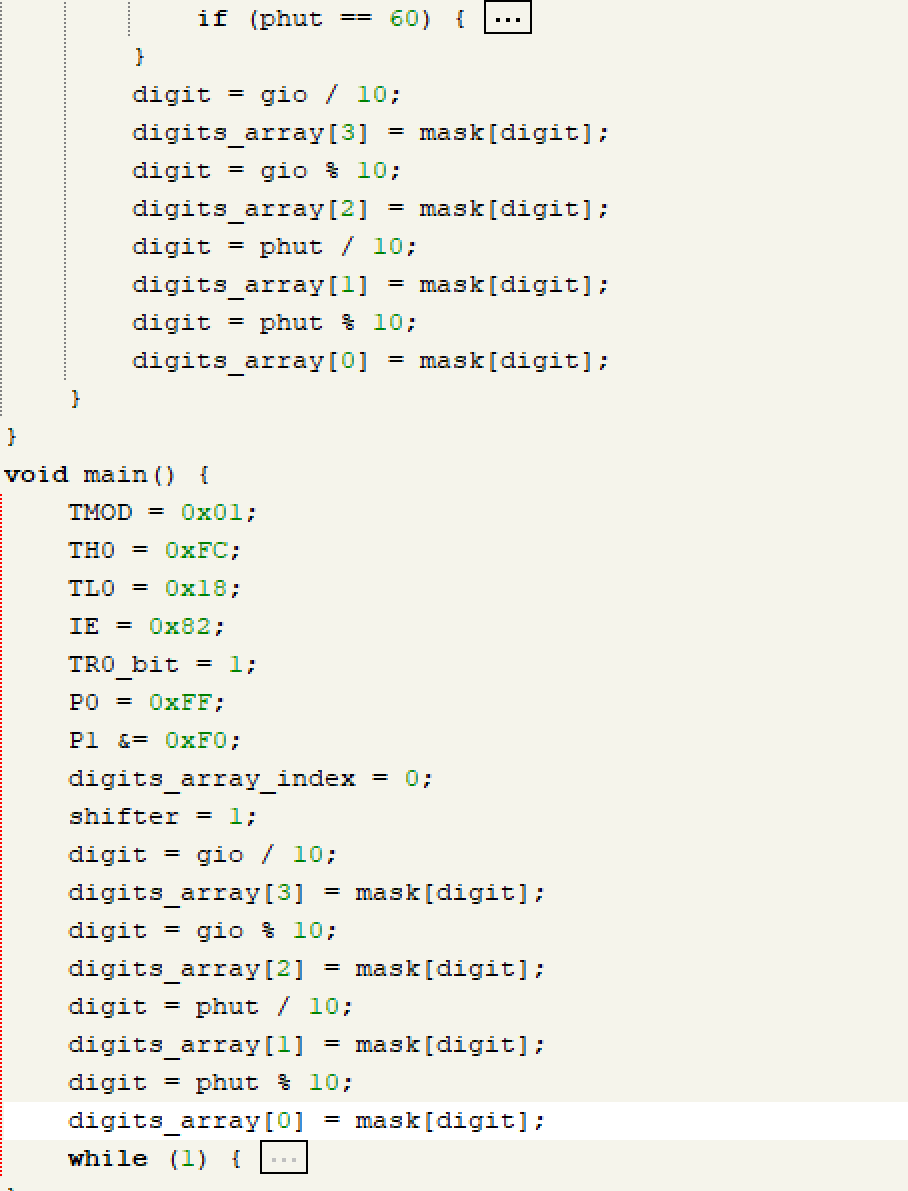
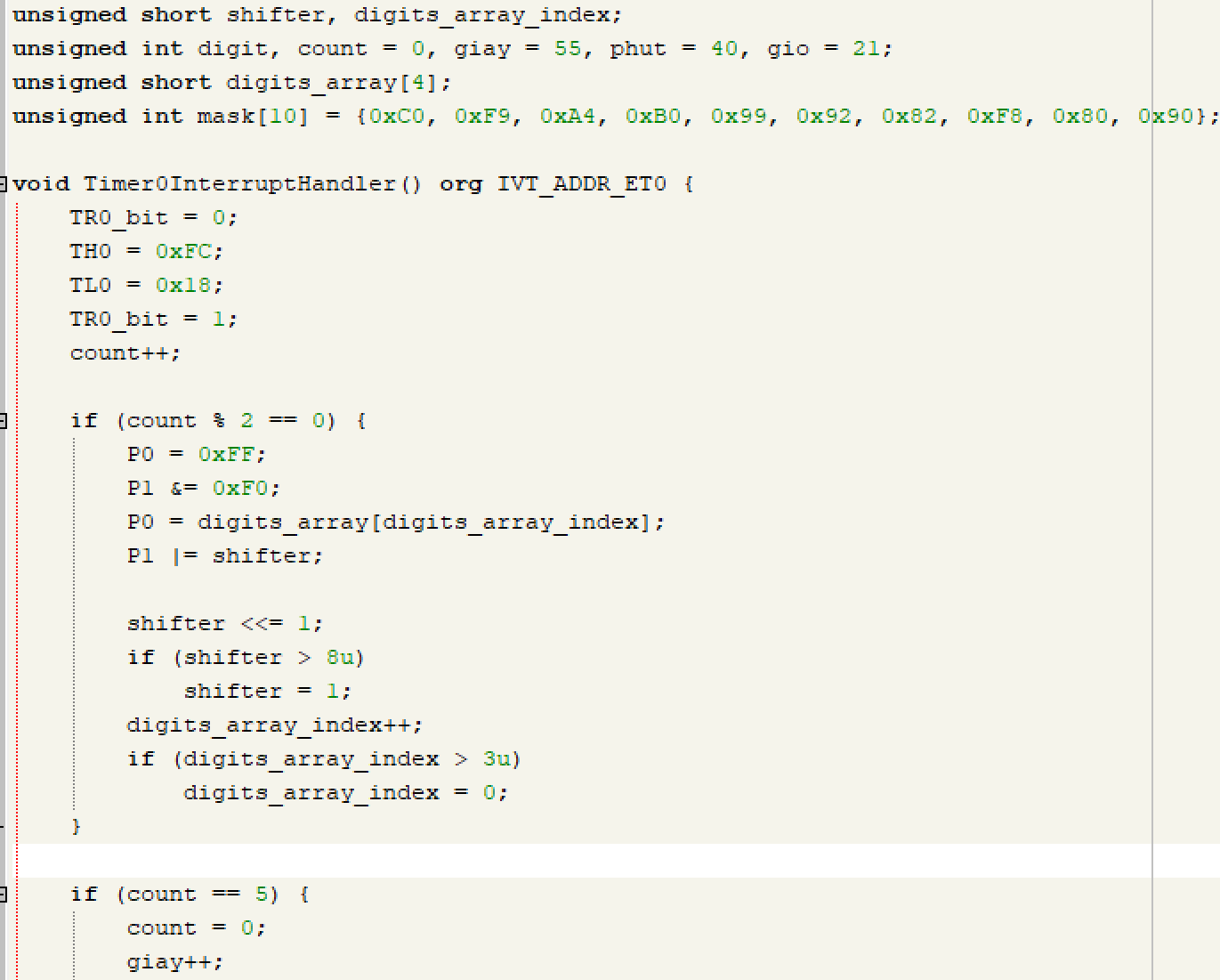
1. Xây dựng đồng hồ hiển thị giờ và phút lên 04 LED 7 thanh. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**



Hình 3.4. Hình ảnh mô phỏng trên proteus

**- Code chương trình**

****

Hình 3.5. Hình ảnh code chương trình

BÀI 4: GIAO TIẾP VỚI CẢM BIẾN NHIỆT ĐỘ

4.1. Mục đích

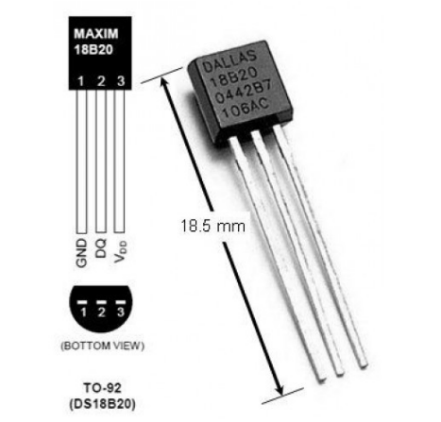
- Tìm hiểu cảm biến nhiệt độ DS1820;

- Lập trình ứng dụng đo và hiển thị nhiệt độ lên LCD.

4.2. Cơ sở lý thuyết

4.2.1. Cảm biến nhiệt độ DS1820

Cảm biến nhiệt độ DS18B20 là cảm biến nhiệt độ chuyên dùng để đo nhiệt độ môi trường với đầu ra dạng số khác với một số loại cảm biến analog như: LM35, LM335



Hình 4.1. Hình ảnh cảm biến nhiệt độ DS18B20

Phương pháp đọc trạng giá trị của cảm biến DS1820 theo chuẩn giao tiếp 1 dây(one wire)

Chuẩn one wire theo phương thức giao tiếp master/slave.

Trong mikroc hỗ trợ các hàm để thực hiện việc giao tiếp DS1820 với vi điều khiển.

4.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

4.3.1. Thiết bị thực hành thí nghiệm

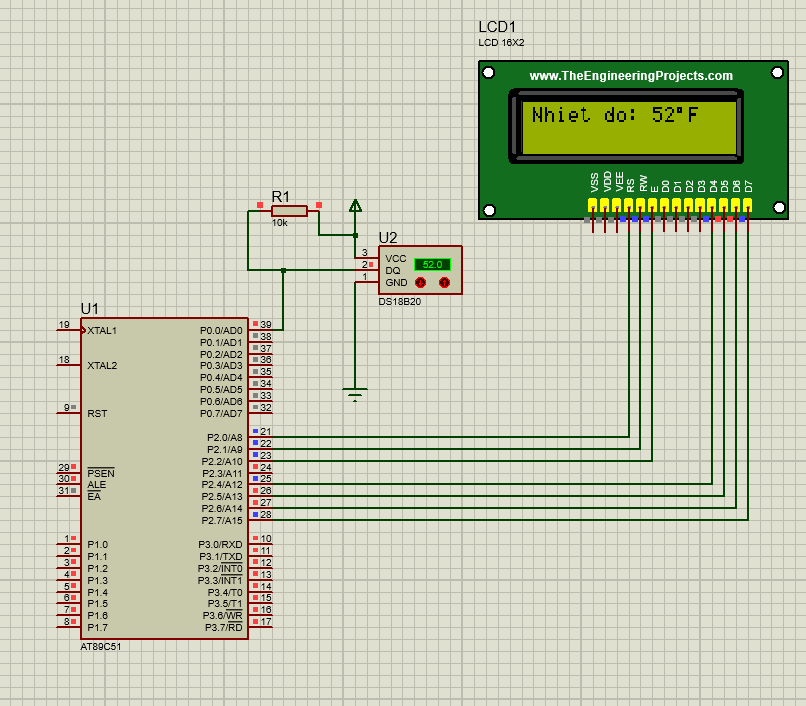
- Máy tính cài đặt phần mềm MikroC for 8051.

4.4. Các bước tiến hành thí nghiệm

4.4.1. Lập trình giao tiếp với cảm biến nhiệt độ DS1820

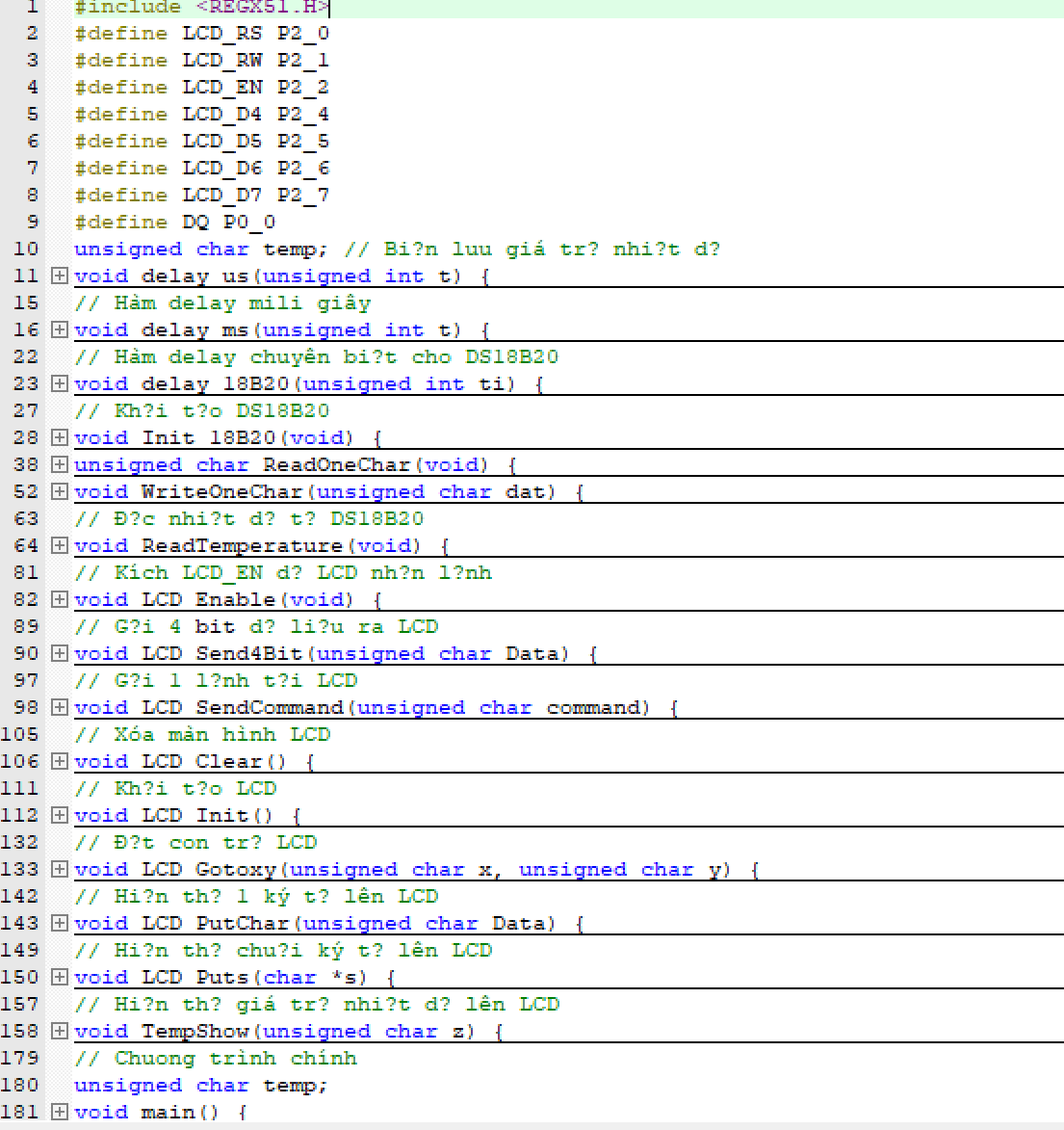
Hiển thị nhiệt độ đo được từ cảm biến DS1820 lên LCD 16x2.

**- Mô phỏng trên Proteus**



Hình 4.2. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**

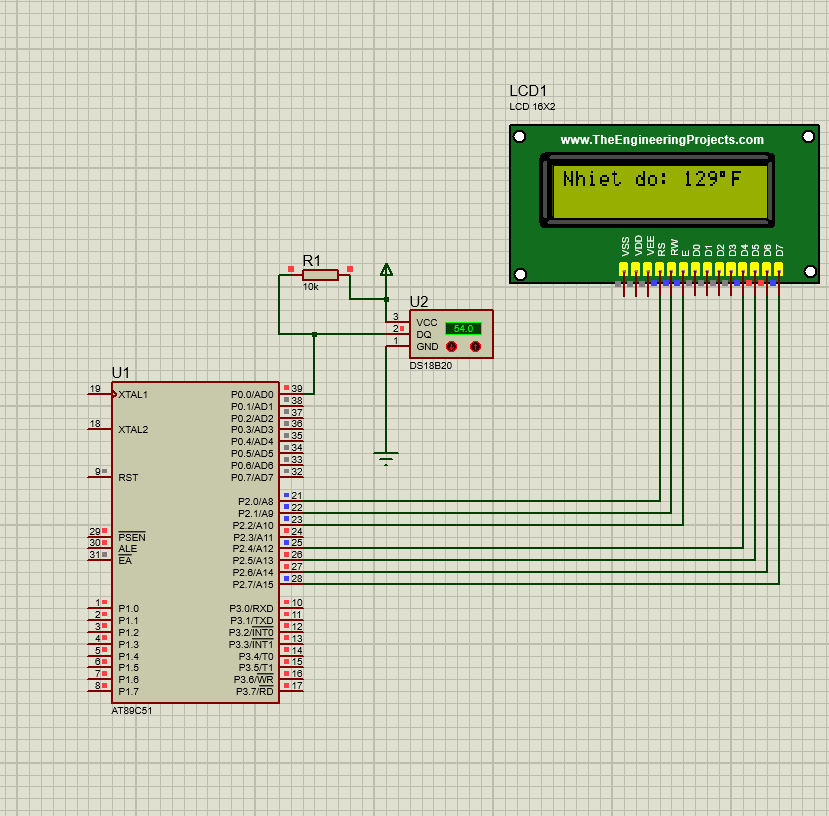
****

Hình ảnh 4.3. Hình ảnh code chương trình

4.4.2. Bài tập thực hành

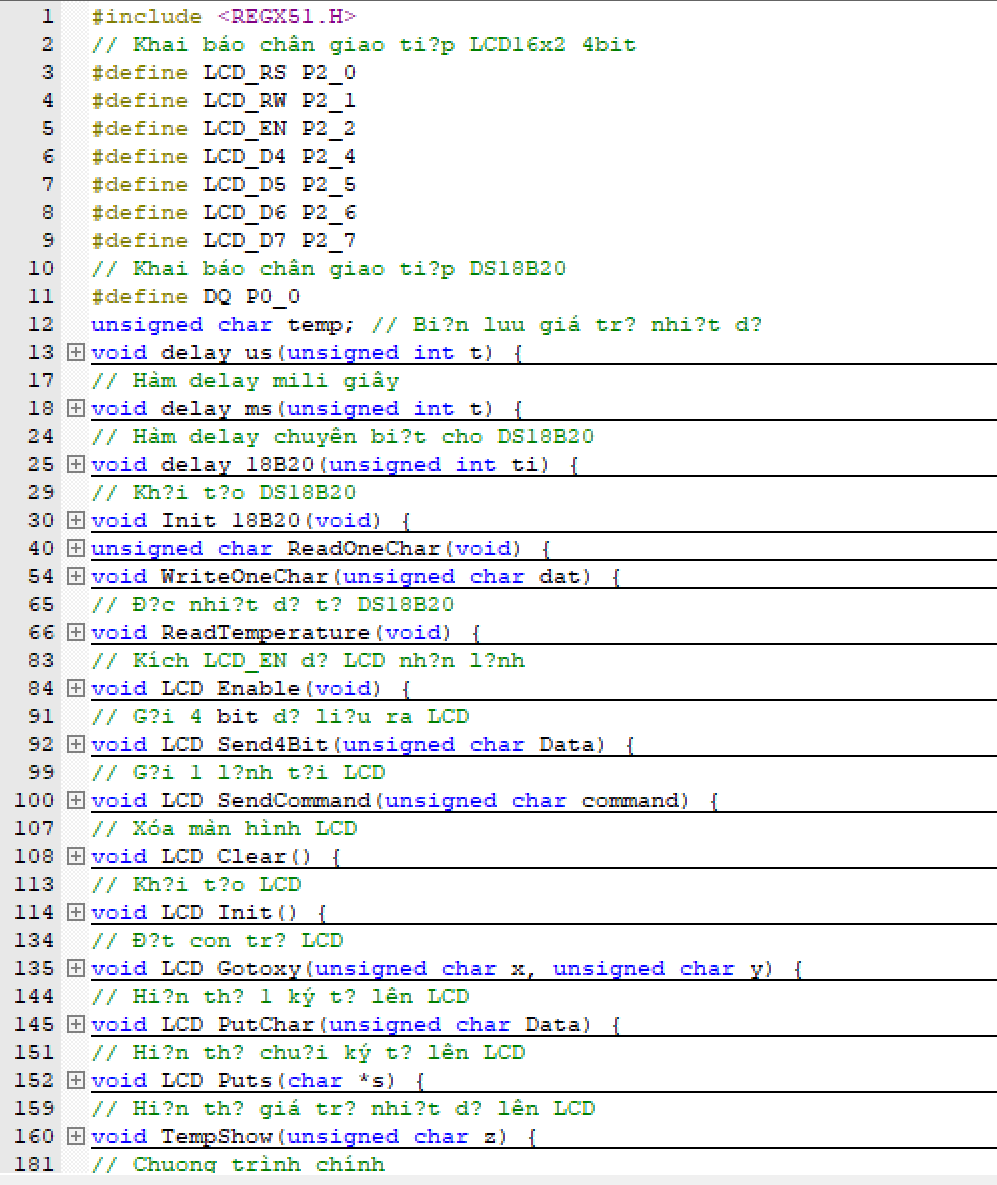
1. Viết lại chương trình trên đổi nhiệt độ C -> F và ngược lại. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**



Hình 4.4. Hình ảnh mô phỏng trên Proteus

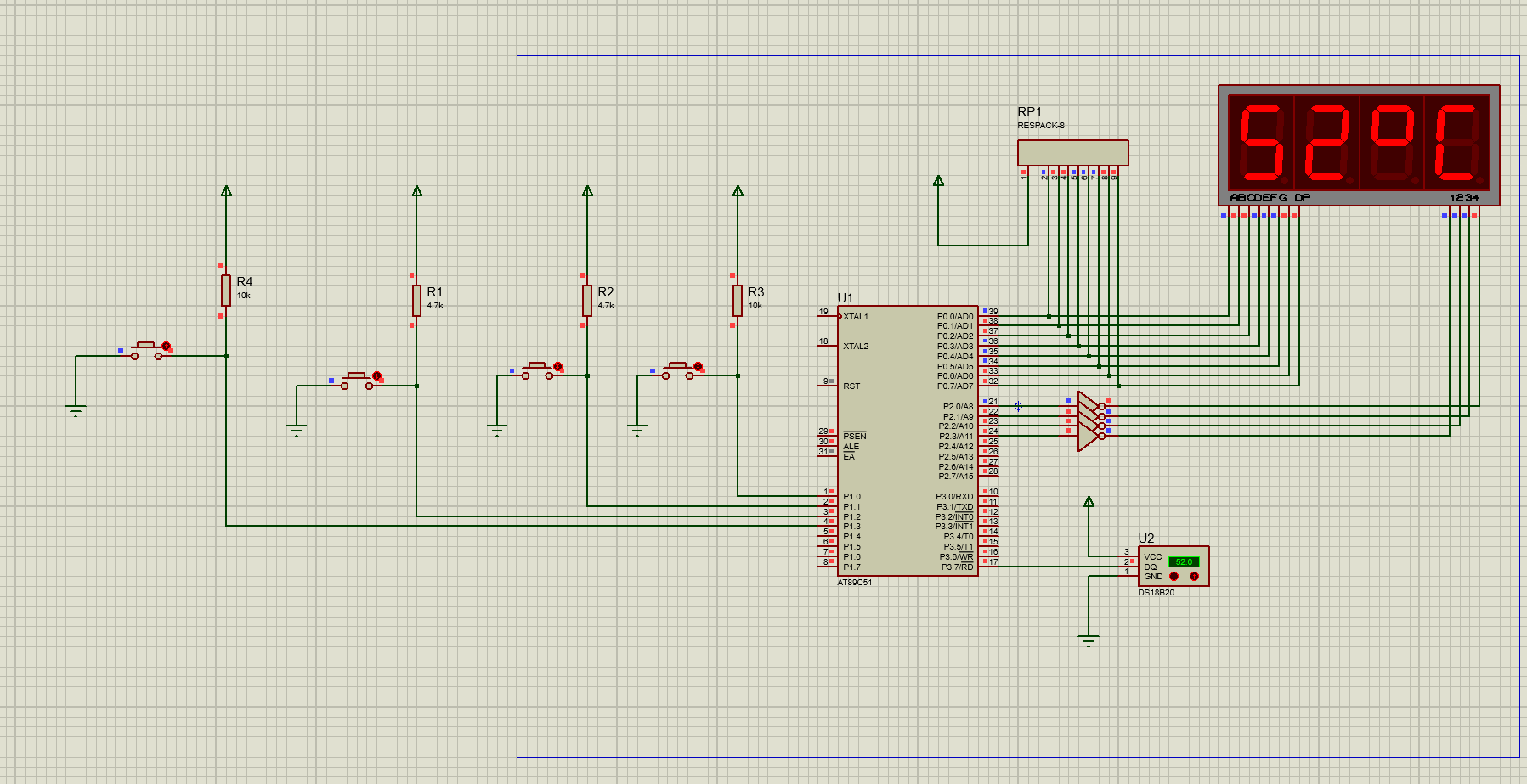
**- Code chương trình**

****

Hình 4.5. Hình ảnh code chương trình

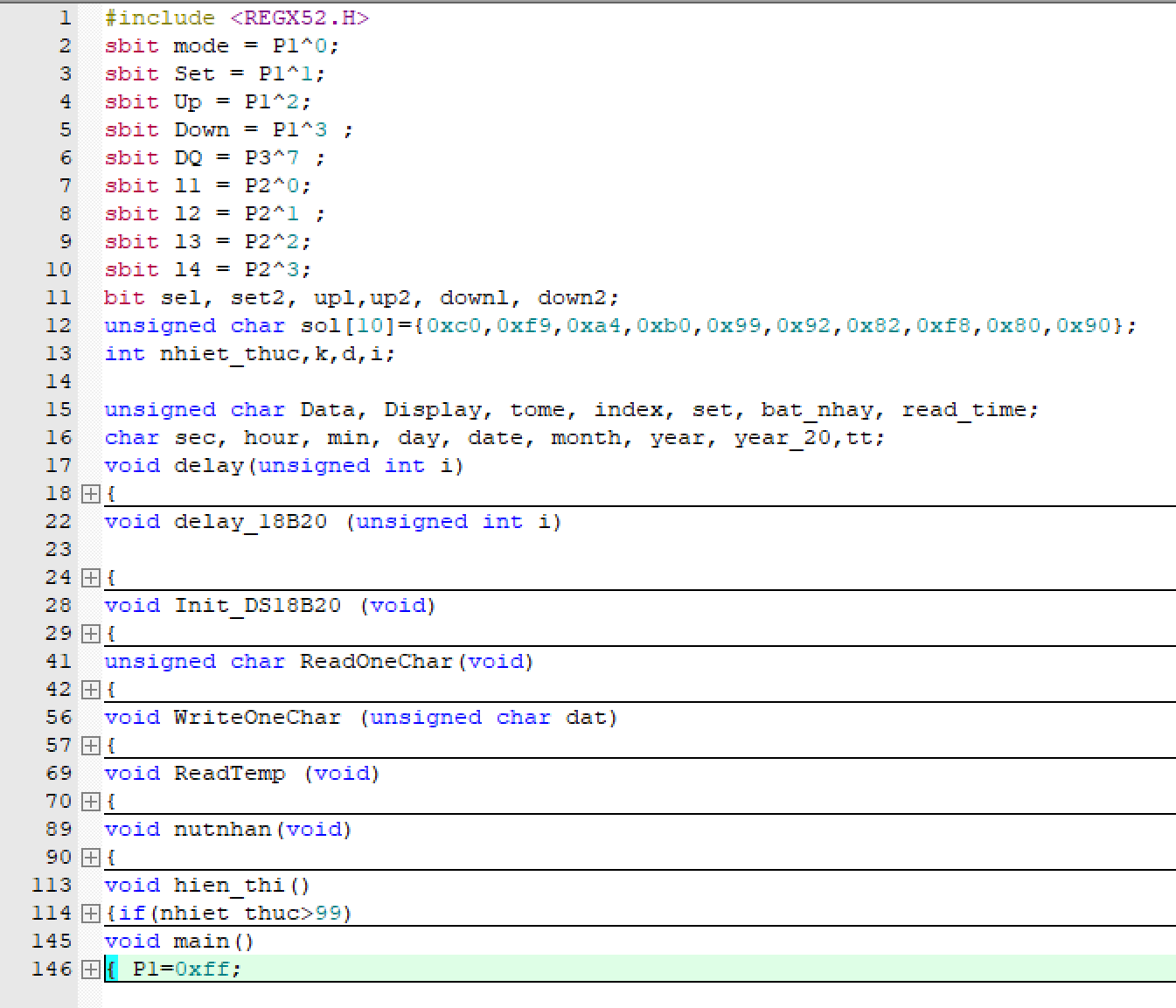
2. Hiển thị nhiệt độ đo được từ cảm biến DS1820 lên các đèn LED 7 thanh. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng.

**- Mô phỏng trên Proteus**



Hình 4.6. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**



Hình 2.7. Hình ảnh code chương trình

BÀI 5: LẬP TRÌNH SỬ DỤNG NGẮT NGOÀI

5.1. Mục đích

- Tìm hiểu ngắt ngoài của vi điều khiển 8051

- Lập trình ứng dụng đếm số lần bấm nút và hiển thị lên LCD

5.2. Cơ sở lý thuyết

5.2.1. Ngắt ngoài của 8051

Khái niệm ngắt: Ngắt là sự đáp ứng các sự kiện bên trong hoặc bên ngoài nhằm thông báo cho bộ vi điều khiển biết thiết bị đang cần được phục vụ.

Ngắt ngoài được sử dụng trong các ứng dụng như:

- Đếm trạng thái xuất hiện sự kiện

- Đo lường tốc độ động cơ dùng encoder

Trình tự thực hiện ngắt của vi điều khiển:

B1: Kết thúc lệnh hiện tại trong chương trình chính của vi điều khiển và lưu địa chỉ của lệnh kế tiếp vào ngăn xếp.

B2: Lưu lại trạng thái hiện hành của tất cả các ngắt vào bên trong.

B3: Nhảy tới một vị trí cố định trong bộ nhớ gọi là bảng vecto ngắt, nơi lưu địa chỉ của trình phục vụ ngắt

B4: Nhận địa chỉ từ bảng vecto ngắt rồi nhảy tới địa chỉ đó và bắt đầu thực hiện trình phục vụ ngắt cho tới lệnh cuối cùng.

B5: Kết thúc trình phục vụ ngắt và vđk trở về đúng vị trí mà nó bị ngắt lúc trước sau nạp địa chỉ lệnh cần làm kế tiếp từ ngăn xếp và thực hiện lệnh này.

5.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

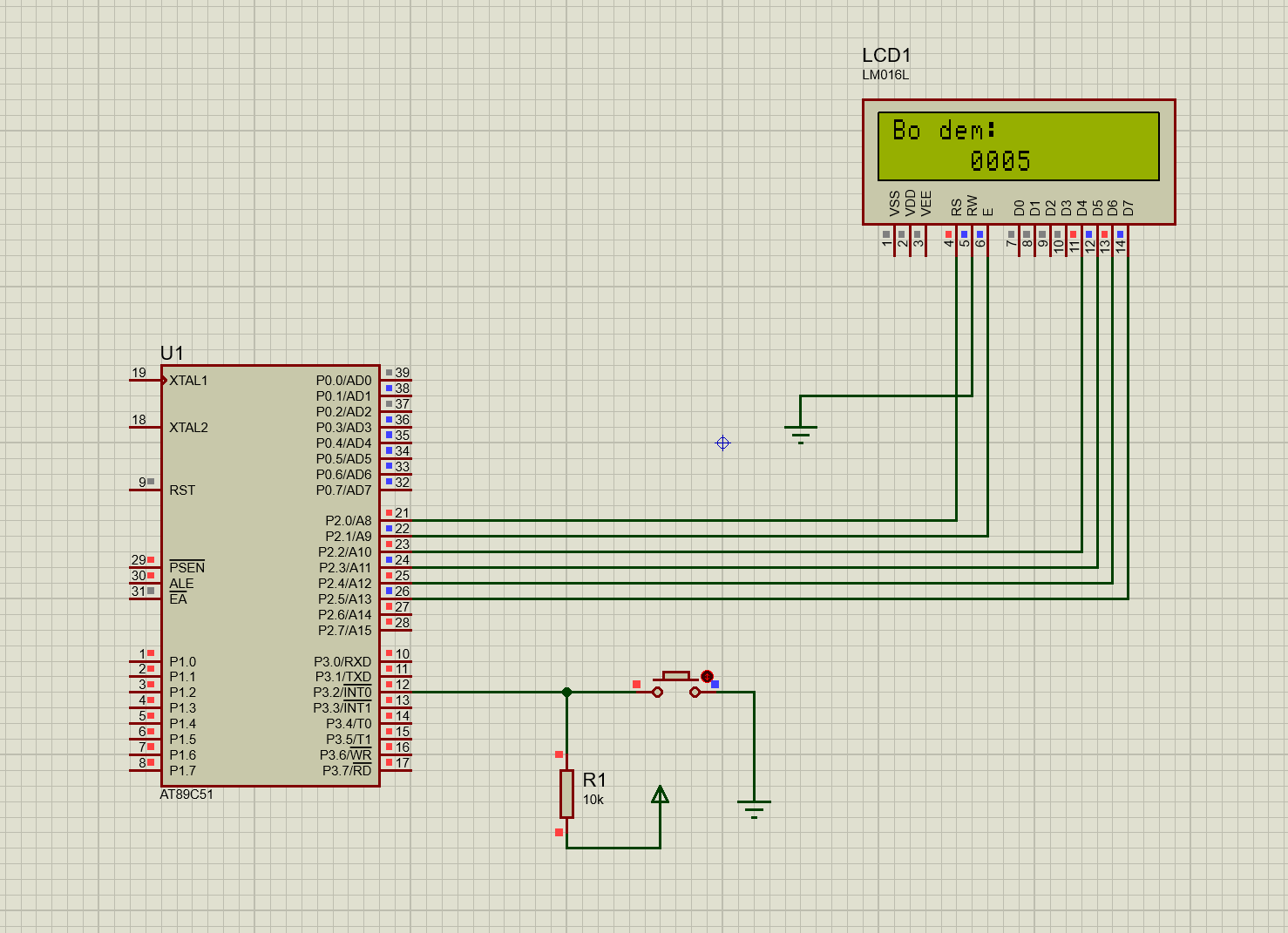
5.3.1. Thiết bị thực hành thí nghiệm

- Máy tính cài đặt phần mềm MikroC for 8051.

5.4. Các bước tiến hành thí nghiệm

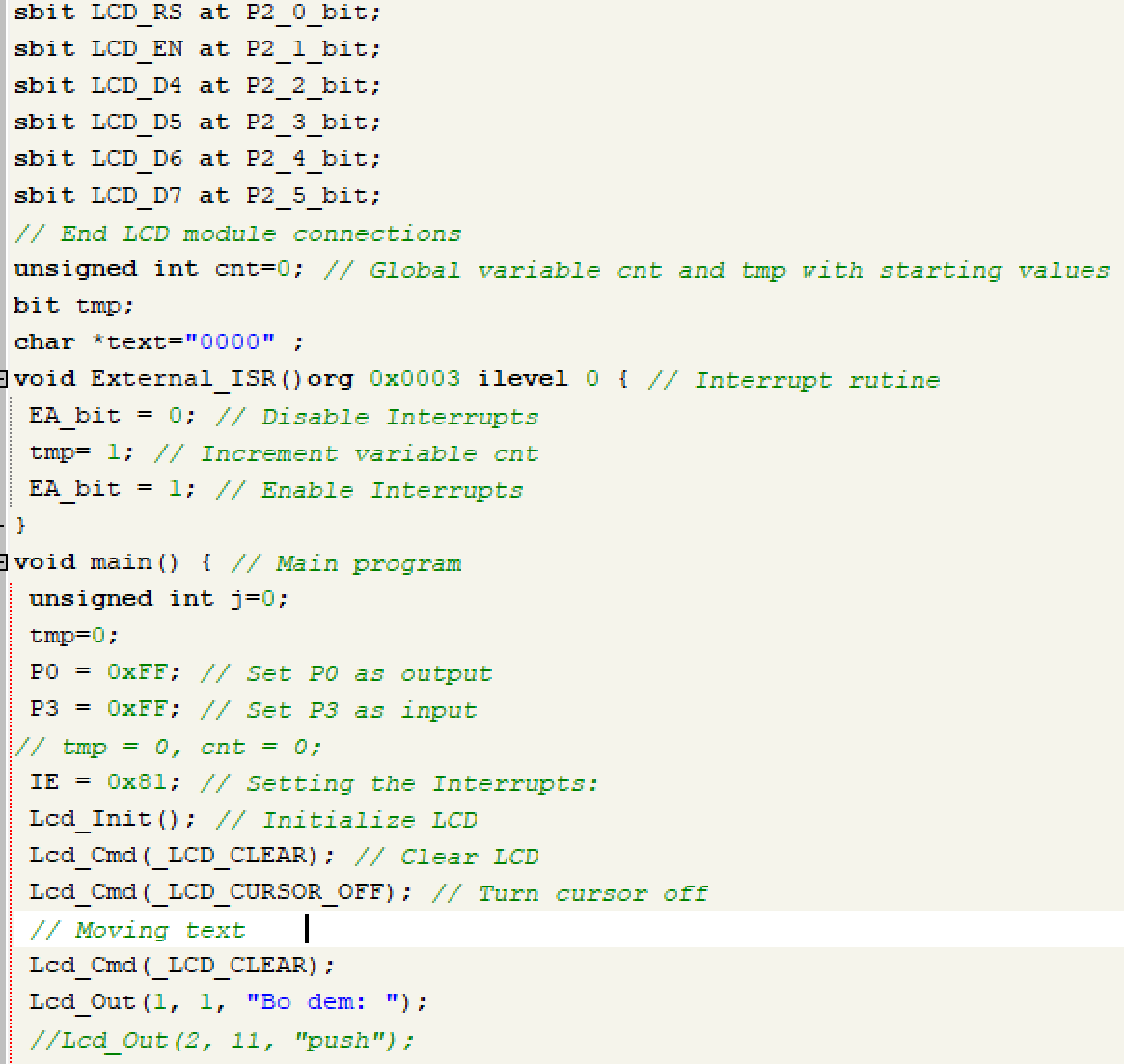
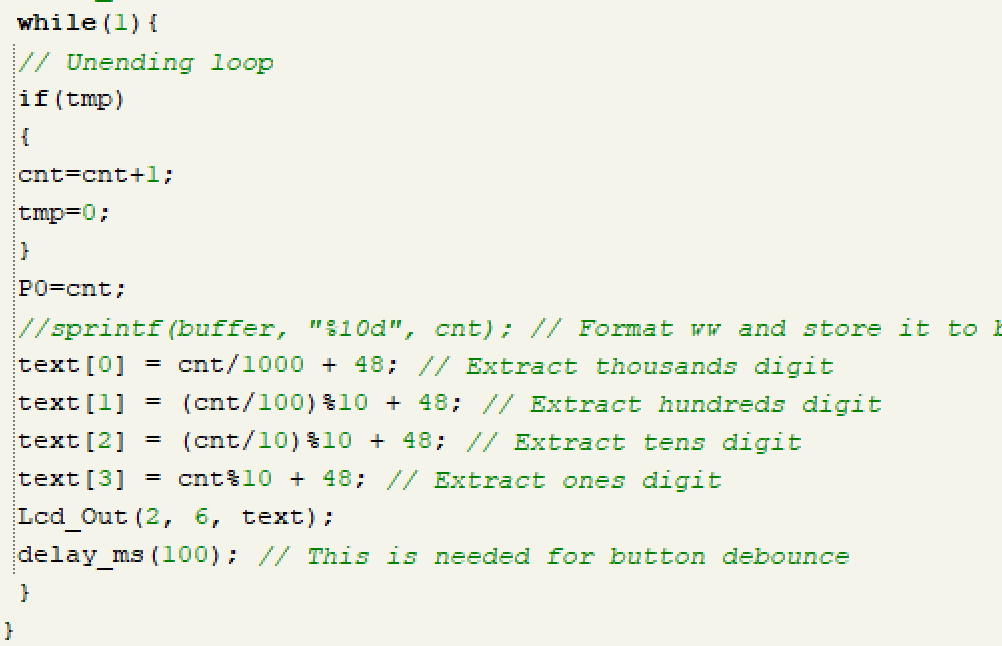
5.4.1. Hiển thị số lần ngắt ngoài trên chân P3.2 lên LCD 16x2

**- Mô phỏng trên Proteus**

****

Hình 5.1. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**

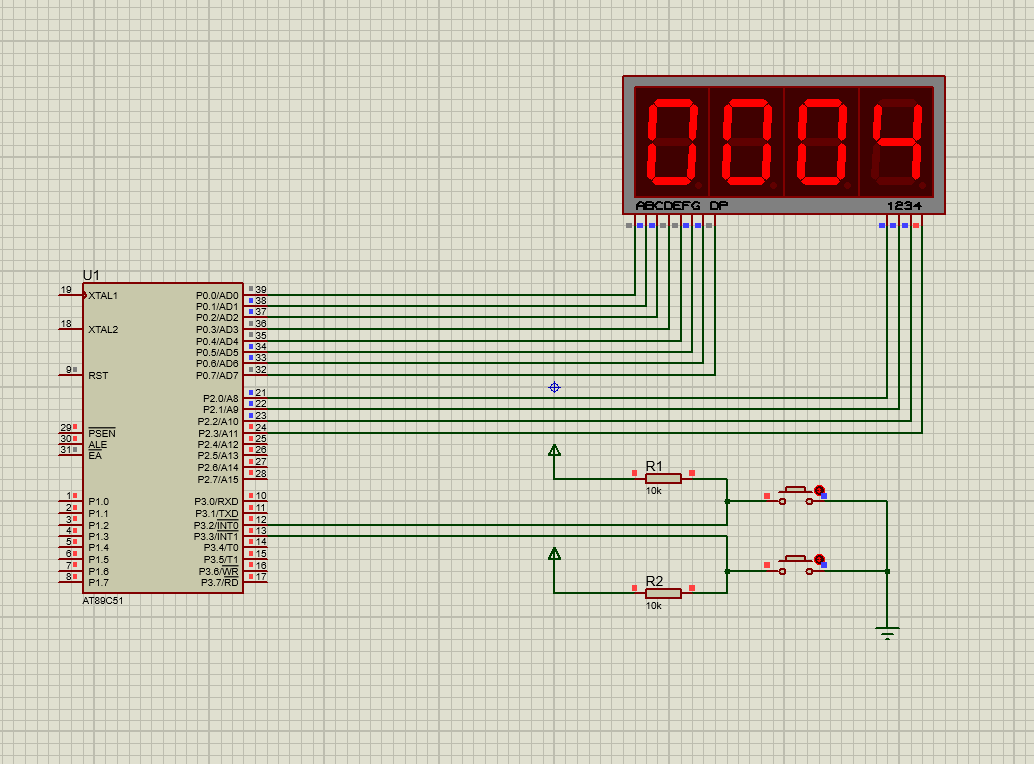
** **

Hình 5.2. Hình ảnh code chương trình

5.4.2. Bài tập thực hành

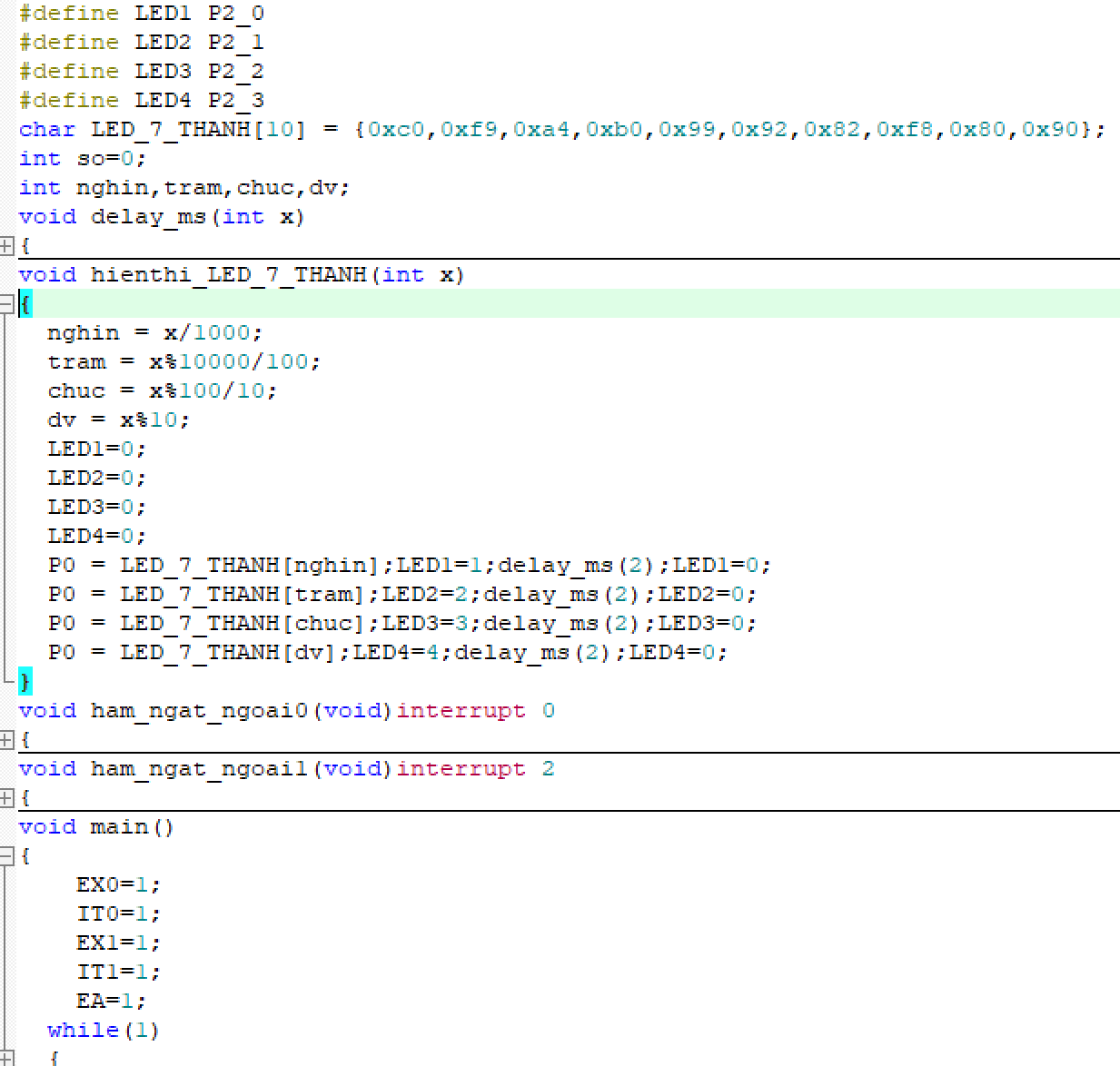
Viết chương trình đếm và hiển thị số lần ngắt ngoài trên chân P3.2 và hiển thị lên các đèn LED 7 thanh. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng.

**- Mô phỏng trên Proteus**



Hình 5.3. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**



Hình 5.4. Hình ảnh code chương trình

BÀI 6: HỆ ĐIỀU HÀNH THỜI GIAN THỰC

6.1. Mục đích

- Tìm hiểu về hệ điều hành thời gian thực;

- Lập trình ứng dụng với hệ điều hành thời gian thực.

6.2. Cơ sở lý thuyết

6.2.1. Hệ điều hành thời gian thực

- Khái niệm hệ điều hành thời gian thực

Hệ thống thời gian thực là hệ thống mà sự hoạt động tin cậy của nó không chỉ phụ thuộc vào sự chính xác của kết quả, mà còn phụ thuộc vào thời điểm đưa ra kết quả, hệ thống có lỗi khi yêu cầu về thời gian không được thoả mãn.

Một RTOS thường có tính mềm dẻo và có tính cấu trúc. Nó cho phép tích hợp thêm các dịch vụ gia tăng theo vòng tròn đồng tâm. Vòng trong cùng hay nhân cung cấp những đặc tính quan trọng nhất của hệ điều hành thời gian thực. Các đặc điểm khác có thể được thêm vào như một vòng ngoài khi cần thiết. Nhân nhỏ của một RTOS thích hợp cho một ứng dụng bộ xử lý nhỏ, trong khi những vòng ngoài có thể

giúp đỡ xây dựng hệ thống thời gian thực lớn. Các RTOS thường cung cấp các mức xử lý ưu tiên. Các công việc ưu tiên cao hơn sẽ được thực hiện trước. Ngoài các chức năng của hệ điều hành như trên, hệ điều hành thời gian thực có thể hỗ trợ thêm các chức năng sau:

+ Lập lịch phân chia thời gian sử dụng tài nguyên, đặt mức ưu tiên các tác vụ.

+ Truyền thông và đồng bộ giữa các tác vụ hoặc giữa tác vụ và ngắt.

+ Phân phối bộ nhớ động.

+ Quản lý các thiết bị vào ra.

Nhiều hệ điều hành không thời gian thực cũng cung cấp nhân của tương tựnhưng điểm khác biệt lớn nhất của hệ điều hành thời gian thực và hệ điều hành không thời gian thực nói chung là tính tiền định (deterministic). Thời gian tiền định tức là các dịch vụ của hệ điều hành thực hiện chỉ được yêu cầu một khoảng thời gian nhất định, muốn tiền định tức là phải tính toán chính xác theo toán học. Các đáp ứng về thời gian là nghiêm ngặt trong hệ điều hành thời gian thực, không thể có thành phần thời gian ngẫu nhiên. Với một phần thời gian ngẫu nhiên có thể tạo ra trễ ngẫu nhiên, từ đó gây ra các đáp ứng deadline không thoả mãn.

+ Thực hiện hệ điều hành tiny Rtos cho 8051

Gồm có RTX Full và RTX tiny

Được tích hợp trong gói phần mềm KeilC C51

6.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

6.3.1. Thiết bị thực hành thí nghiệm

- Máy tính cài đặt phần mềm KeilC 8051.

6.4. Các bước tiến hành thí nghiệm

Nội dung thực hiện:

On/Off 4 Led với 4 task khác nhau

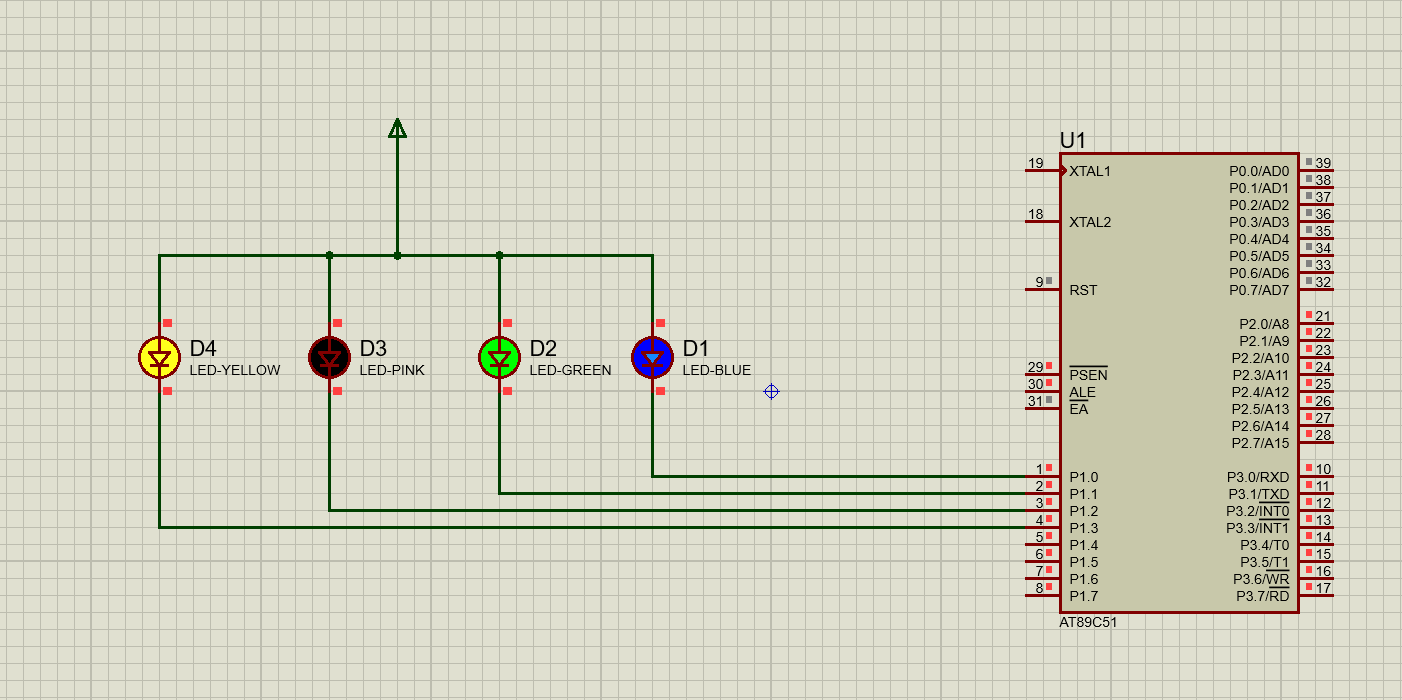
- Task 1: nhấp nháy led 1 với chu kỳ 1000ms

- Task 2: nhấp nháy led 3 với chu kỳ 600ms

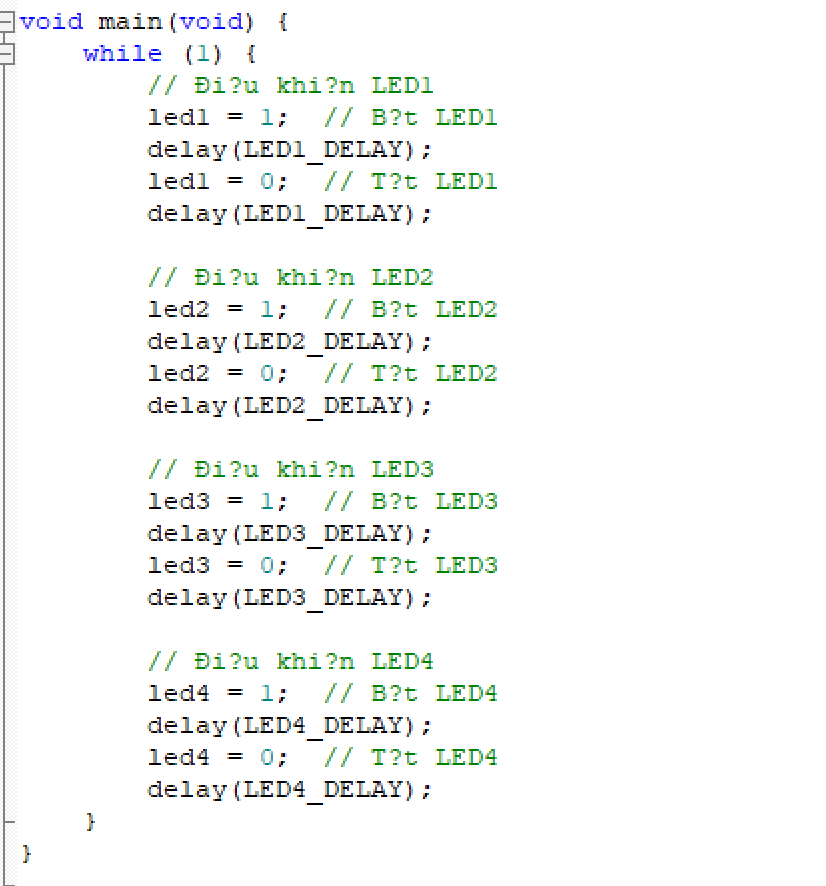
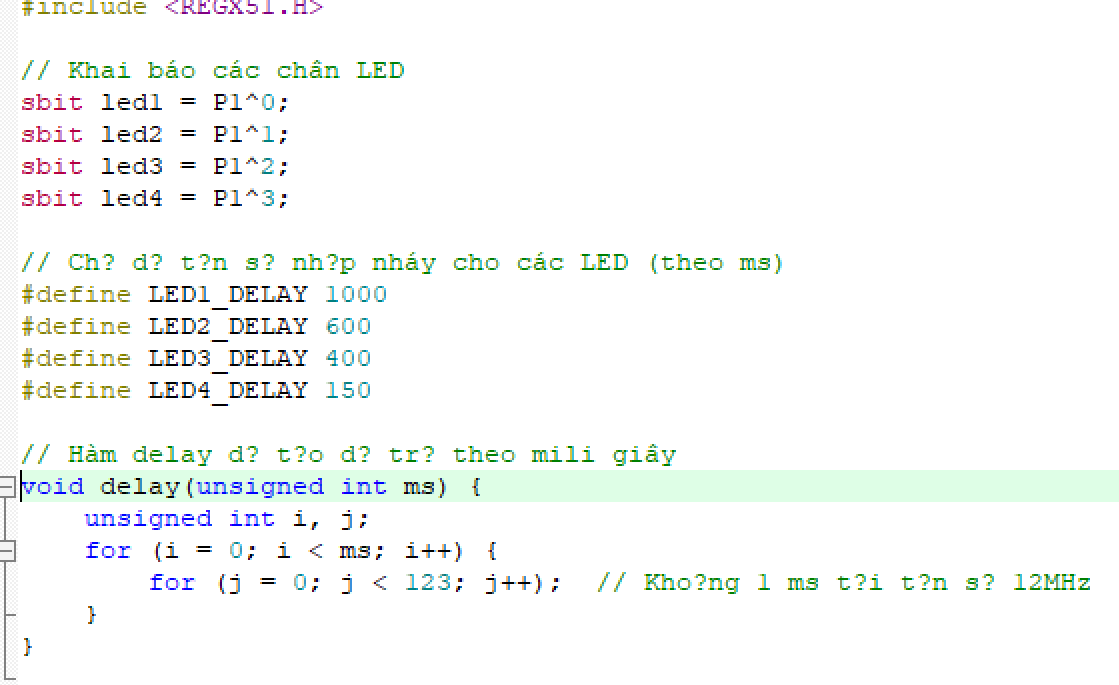
- Task 3: nhấp nháy led 3 với chu kỳ 400ms

- Task 4: nhấp nháy led 1 với chu kỳ 150ms

**- Mô phỏng trên proteus**



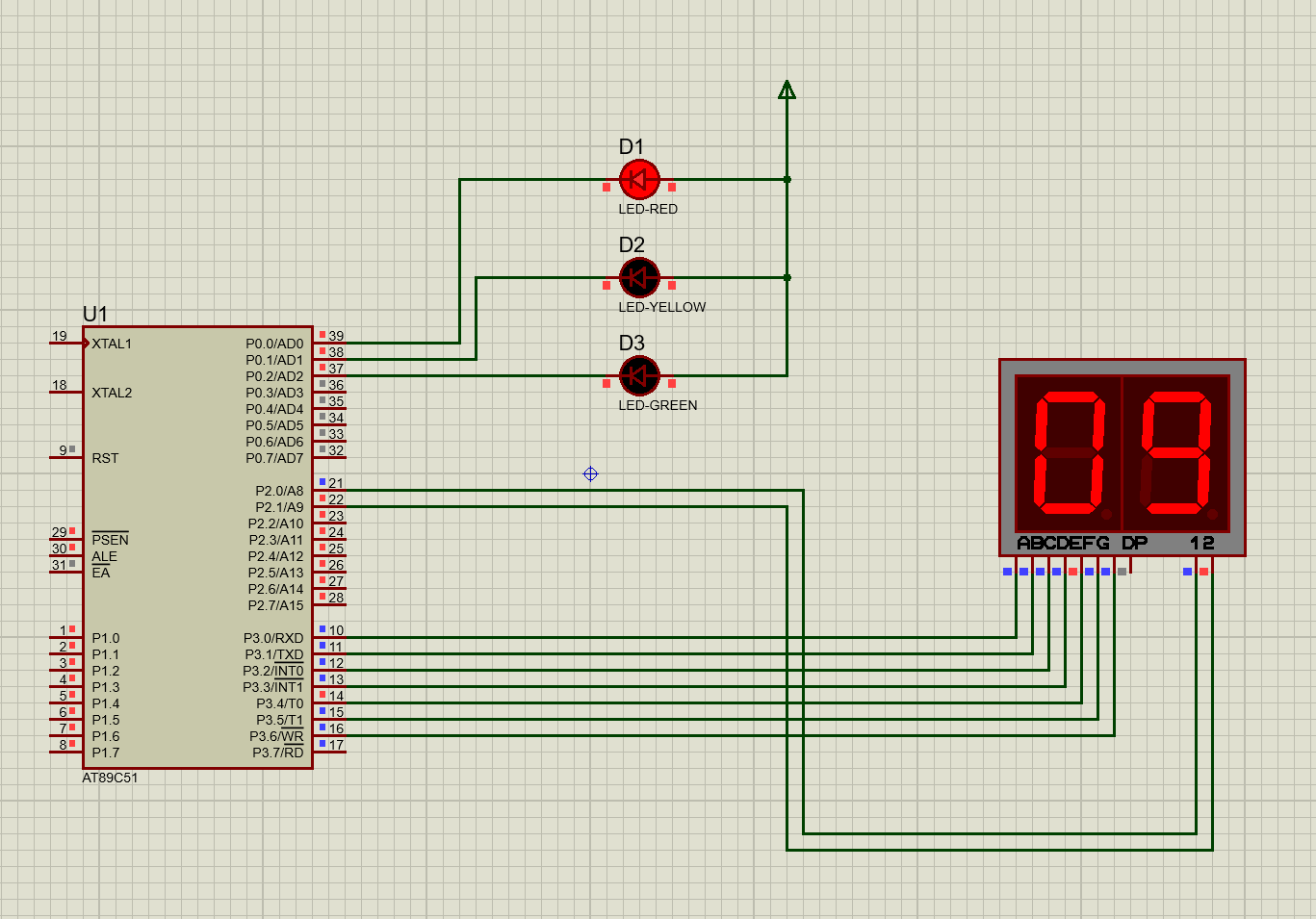
Hình 6.1. Hình ảnh mô phỏng trên Proteus



Hình 6.2. Hình ảnh code chương trình

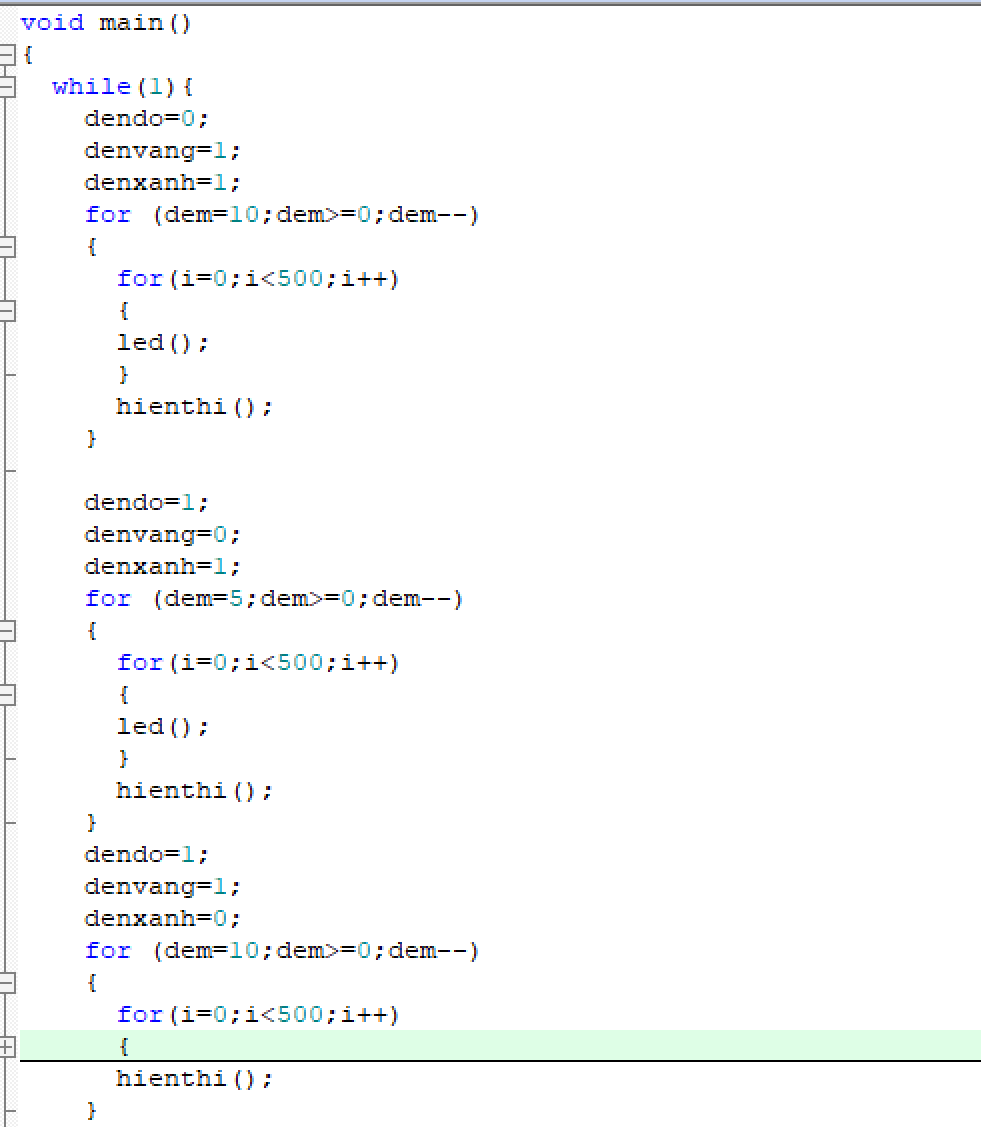
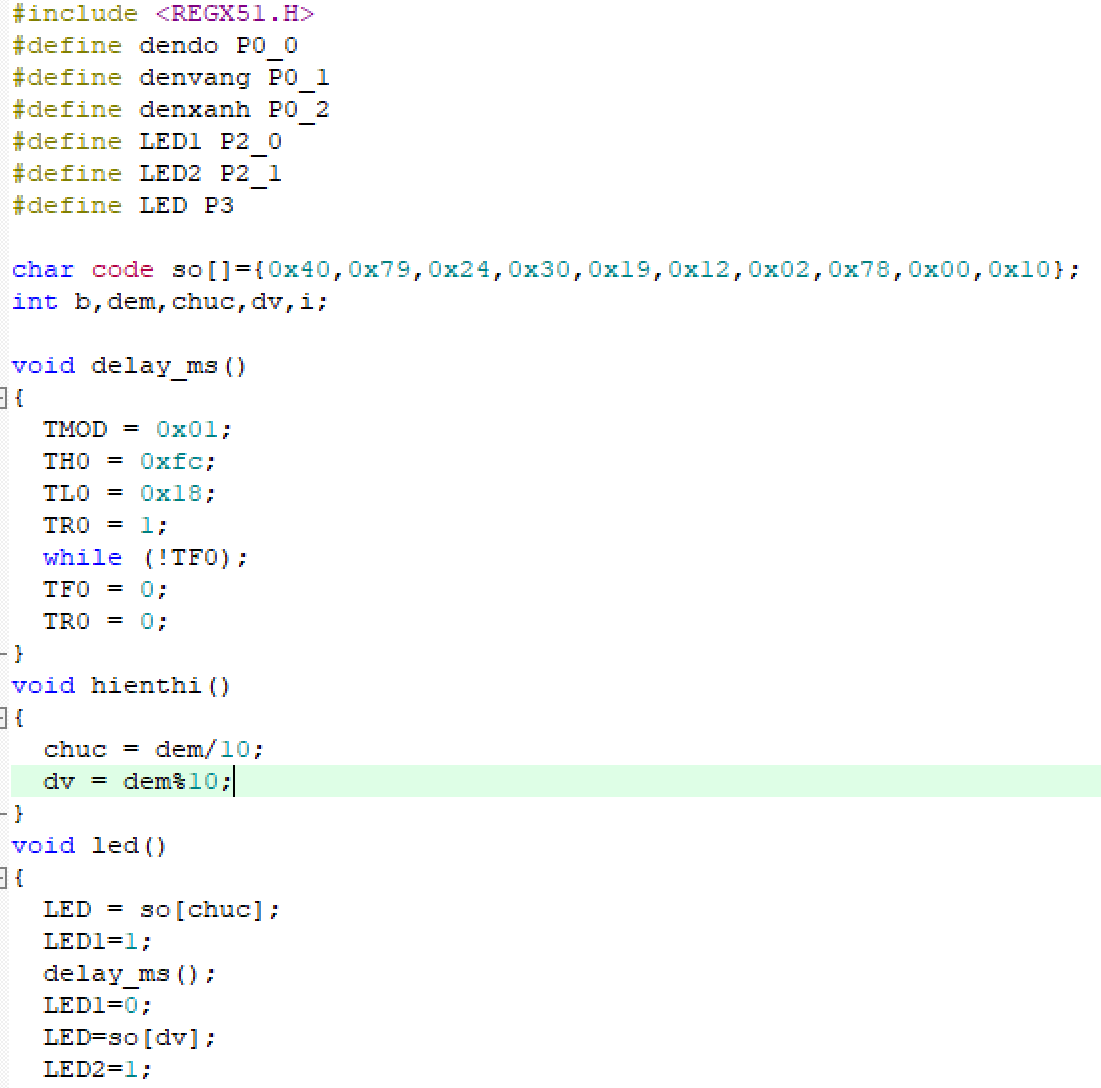
Viết chương trình đèn giao thông. Mô phỏng trên proteus và kiểm tra trên mạch phần cứng

**- Mô phỏng trên Proteus**



Hình 6.1. Hình ảnh mô phỏng trên Proteus

**- Code chương trình**

****

Hình 6.2. Hình ảnh code chương trình