

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------------------------

BÀI TẬP LỚN

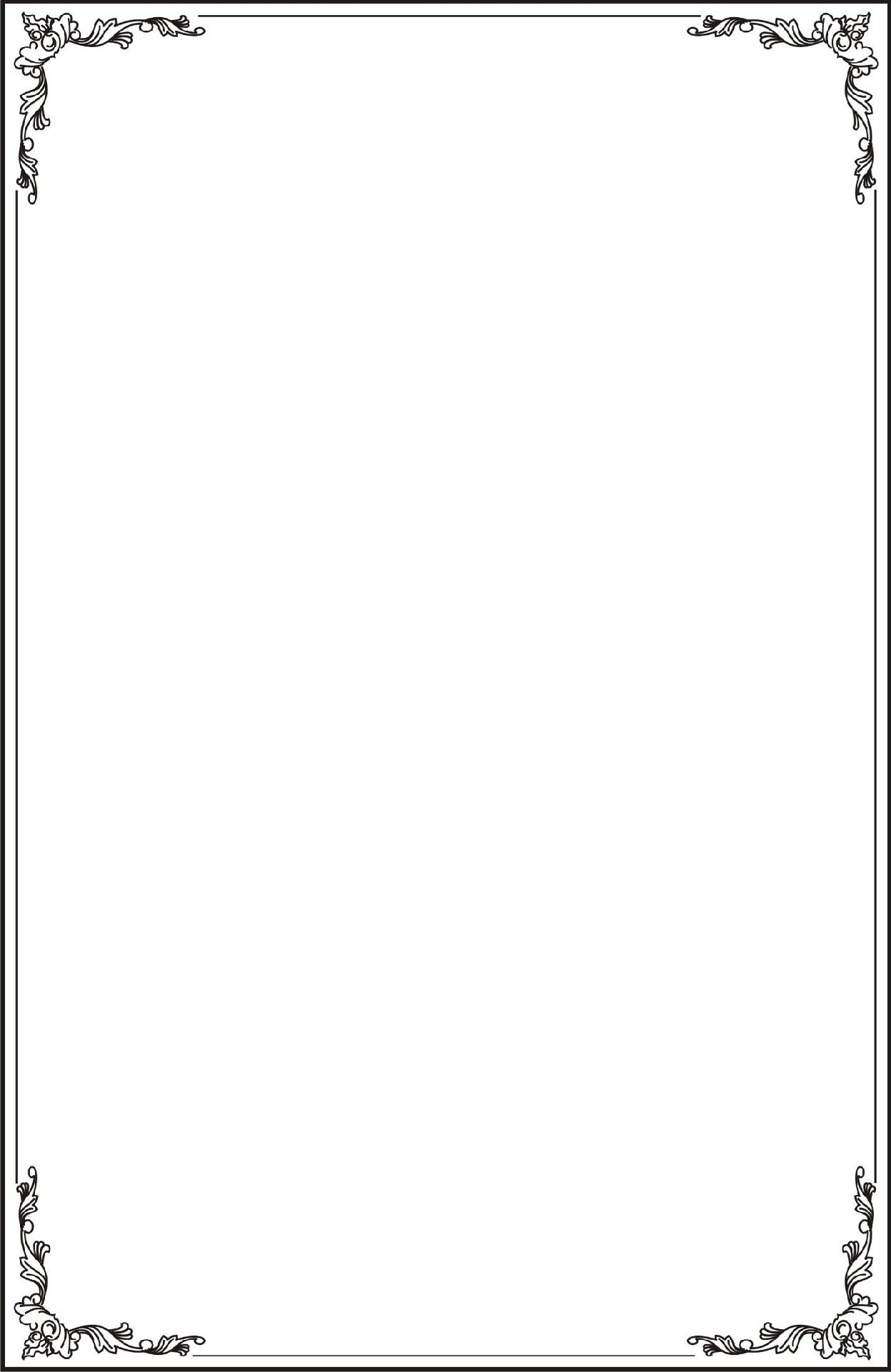
KIẾN TRÚC MÁY TÍNH

KIẾN TRÚC MÁY TÍNH – LỚP N02 – NHÓM 03

**Nhóm: LỚP N01 – NHÓM 01**

|  |  |  |
| --- | --- | --- |
| **Tên các thành viên** | | |
| **Tên thành viên** | **Mã số sinh viên** |  |
| Trần Đình Lâm Anh | B21DCCN013 |  |
| Nguyễn Việt Khiêm | B21DCCN468 |  |
| Nguyễn Thị Lan | B21DCCN818 |  |
| Trương Công Tuấn Thành | B21DCCN681 |  |
| Nguyễn Đăng Huy | B21DCCN062 |  |
| **Giảng viên hướng dẫn:** ThS. Đinh Xuân Trường | | |

**Hà Nội 2023**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------------

**BÀI TẬP LỚN**

**KIẾN TRÚC MÁY TÍNH**

**Nhóm: LỚP N01 – NHÓM 01**

|  |  |
| --- | --- |
| **Tên các thành viên** | |
| **Tên thành viên** | **Mã số sinh viên** | |
| Trần Đình Lâm Anh | B21DCCN013 | |
| Nguyễn Việt Khiêm | B21DCCN468 | |
| Nguyễn Thị Lan | B21DCCN818 | |
| Trương Công Tuấn Thành | B21DCCN681 | |
| Nguyễn Đăng Huy | B21DCCN062 | |
| **Giảng viên hướng dẫn:** ThS. Đinh Xuân Trường | |

**Hà Nội 2023**

**MỤC LỤC**

[Lời nói đầu 1](#_Toc137388455)

[PHẦN NGHIÊN CỨU LÀM VIỆC THEO NHÓM 2](#_Toc137388456)

[1. Nội dung chính của đề tài 2](#_Toc137388457)

[2. Miêu tả chương trình 2](#_Toc137388458)

[ Phần 1 : Lý thuyết mạch 2](#_Toc137388459)

[ Phần 2 : Code Assembly 6](#_Toc137388460)

[3. Miêu tả giao diện chương trình 11](#_Toc137388461)

[Tài liệu tham khảo 14](#_Toc137388462)

[Đánh giá về sự đóng góp của các thành viên 14](#_Toc137388463)

# Lời nói đầu

Đây là báo cáo phần nghiên cứu của đề tài “Mô phỏng hiển thị nhiều đèn led đơn nhấp nháy”. Với mục tiêu chính của đề tài là tạo ra một phần mềm mô phỏng đơn giản và hiệu quả để hiển thị quá trình nhấp nháy của đèn LED.

Các tính năng chính của mô phỏng bao gồm khả năng di chuyển, màu sắc và toạ độ hoạt động của đèn LED. Để thực hiện đề tài này, ta sử dụng một phần mềm mô phỏng phát triển bằng ngôn ngữ lập trình assembly. Kết quả của dự án này sẽ mang lại một công cụ mô phỏng đèn LED đơn nhấp nháy linh hoạt và dễ sử dụng. Trong quá trình thực hiện đề tài không tránh khỏi những sai sót, nhóm chúng em mong sẽ nhận được sự góp ý và đánh giá của thầy.

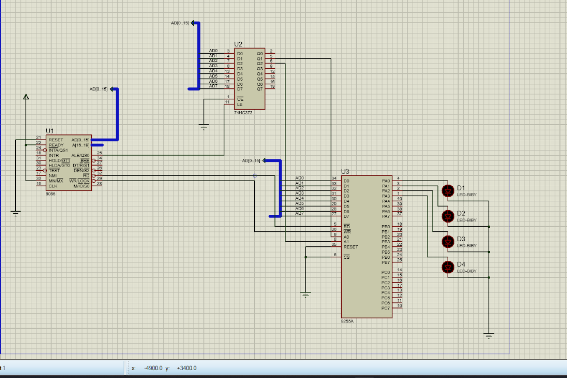
# PHẦN NGHIÊN CỨU LÀM VIỆC THEO NHÓM

## Nội dung chính của đề tài

* In ra chữ PTIT bằng các điểm ảnh giả lập cho việc đèn led đổi màu.
* Chữ PTIT di chuyển xung quanh màn hình.
* Khi di chuyển đến các cạnh của màn hình sẽ đổi hướng.

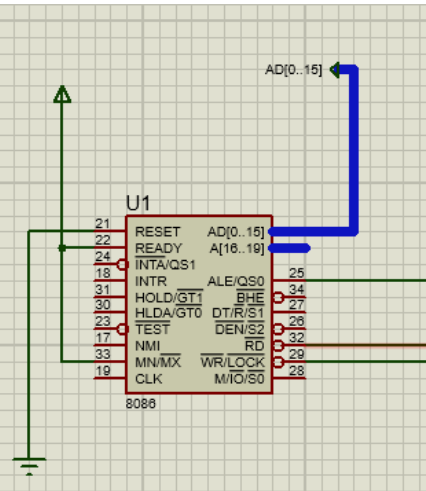
## Miêu tả chương trình

## Phần 1 : Lý thuyết mạch



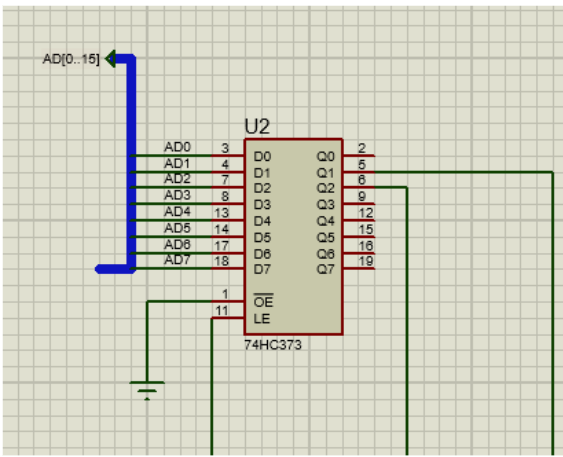
***Hình 1: Mạch***

* Vi xử lý 8086



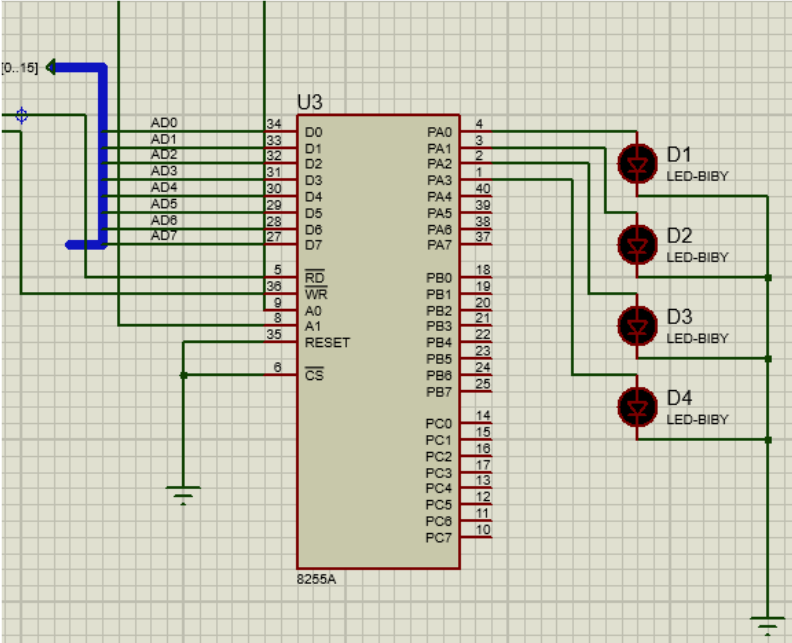
***Hình 2: Vi xử lý 8086***

* AD0-AD7: 8 chân dồn kênh cho phần thấp bus A và bus D ;
* A8-A15: 8 chân tín hiệu phân cao bus A( Ở đây chúng ta chỉ cần sử dụng đến 8 chân dồn kênh cho phần thấp bus A và bus D) như mạch ở trên thì chúng ta đặt nhãn cho cả 16 kênh dồn AD0-AD15 là AD[0..15]
* Chân chốt ALE thì sẽ đc nối với chân LE của IC 74HC373
* READY: Tín hiệu báo cho CPU biết tình trạng sẵn sàng của thiết bị ngoại vi hay bộ nhớ. Khi READY = 1, CPU có thể thực hiện đọc ghi ngay mà không cần chèn thêm các chu kỳ đợi; Khi thiết bị ngoại vi hay bộ nhớ chưa sẵn sàng, chúng gửi READY=0 báo cho CPU kéo dài lệnh đọc ghi bằng cách thêm các chu kỳ đợi.
* MN/MX : chân tín hiệu xác định chế độ làm việc của CPU ở chế độ
  + chúng ta nối 2 chân MN/MX và chân READY này với cả nguồn
* RD: Xung cho phép đọc (đảo) và  WR : Tín hiệu cho phép ghi chúng ta sẽ nối với 2 chân tương ứng của mạch 8255A
* Chân RESET để lặp lại chương trình thì sẽ nối với chân đất
* Vi mạch chốt dữ liệu IC 74HC373:



***Hình 3: Vi mạch chốt dữ liệu IC 74HC373***

* D0-D7 là tín hiệu dữ liệu của bus AD[0..15]
* Chân OE là cho phép dữ liệu đi ra chúng ta sẽ nối với chân đất
* Khi chân LE lên mức cao thì dữ liệu đầu ra  Q0 -> Q7 sẽ ứng  dữ liệu đầu vào tại thời điểm chốt (tạo ra 1 xung). Sau thời điểm chốt dữ liệu đầu ra sẽ không thay đổi khi dữ liệu đầu vào có thay đổi như thế nào đi nữa và chúng ta sẽ nối Q1 với A0 với Q2 với A1 là 2 chân dùng để chọn cổng của vi mạch 8225A
* Vi mạch 8225A:

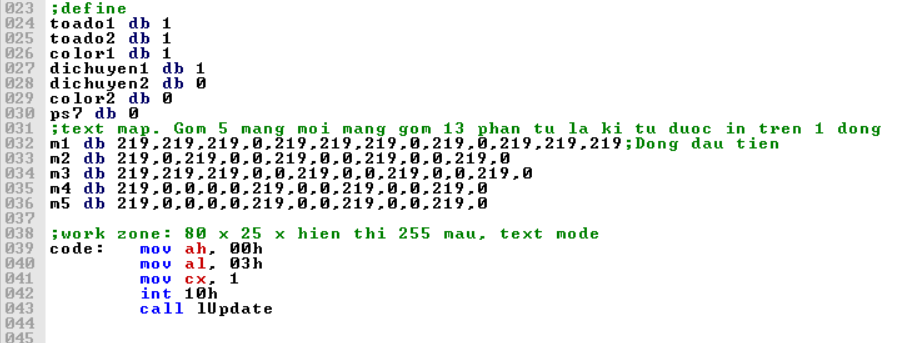


***Hình 4: Vi mạch 8225A***

* D0-D7 là tín hiệu dữ liệu của bus AD[0..15] chân chọn chip CS và chân RESET sẽ đc nối vào chân đất
* PA0-PA7 là các chân của cổng A
* PB0-PB7 là các chân của cổng B
* PC0-PC7 là các chân của cổng C
* Ở code của bài này chúng ta sẽ dùng cổng A và cổng C làm output còn cổng B là input và tất cả đều ở chế độ 0
* Ở mạch trên chúng ta mô phỏng nhấp nháy cho 4 đèn mỗi đèn sẽ đc nối với lần lượt các chân từ PA0-PA3

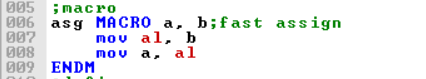
## Phần 2 : Code Assembly

* Dữ liệu đầu vào:
* Các biến phục vụ cho việc thay đổi trạng thái chữ PTIT và mô hình chữ PTIT, để in ra màn hình.
* Dữ liệu đầu ra:
* Màn hình đèn LED.
* Đèn LED theo từng khung hình với các màu sắc khác nhau (tổng cộng là 10 loại màu) và di chuyển
* Thay đổi màu sắc mỗi khung hình và trạng thái di chuyển thay đổi khi chạm rìa màn hình.



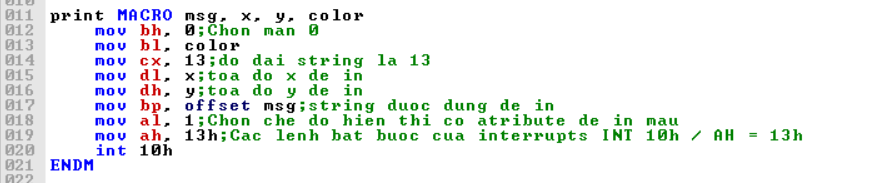
***Hình 5: Gán các giá trị, mảng***

* Macro asg:
* Macro dùng để gán nhanh 2 biến bất kì (Do hạn chế của việc phải gán thông qua ax)



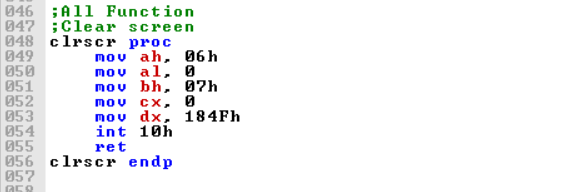
***Hình 6: Hàm macro asg***

* Macro print:
* Dùng để in string msg tại vị trí có tọa độ là (x, y) với màu ‘*color*’



***Hình 7: Macro print***

* Hàm clrscr:
* Được sử dụng để thực hiện xóa toàn bộ các kí tự hiện thời trên màn hình, chức năng này cho phép chúng ta tạo ra một khung trống hoàn toàn để in frame tiếp theo mà không có bất kỳ hiển thị nào từ frame trước đó. Kết quả là khi người ta nhìn vào, sẽ có cảm giác rằng các ký tự "PTIT" đang di chuyển thay vì chỉ đơn thuần là hiệu ứng in đè chồng lên nhau. Hiệu ứng này tạo ra một trải nghiệm hấp dẫn hơn và làm cho mắt người nhìn có cảm giác như đang chứng kiến sự chuyển động độc đáo của các ký tự.

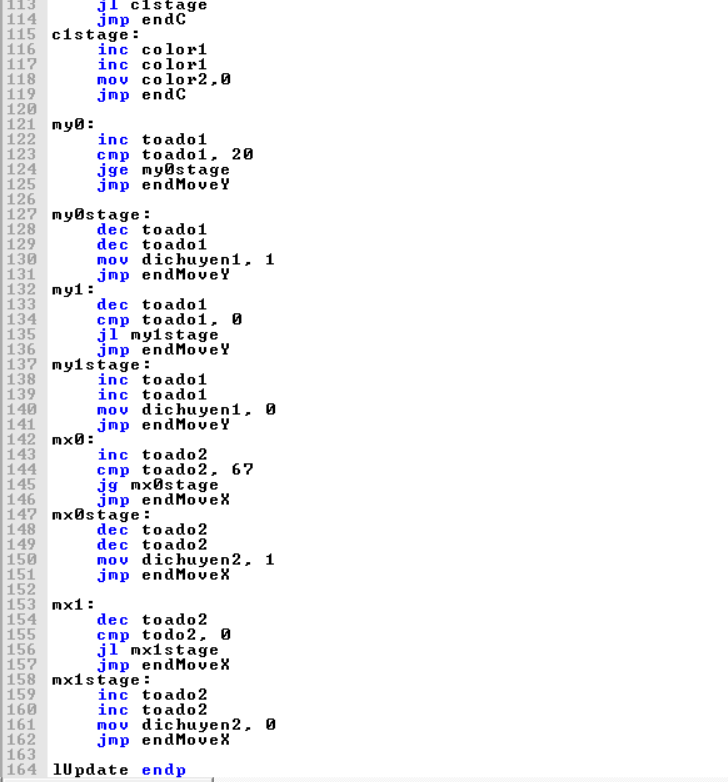


***Hình 8: Hàm clrscr***

* Hàm lUpdate
* Hàm chạy trong mỗi khung hình dùng để tính toán vị trí và màu sắc của chữ PTIT.
* Bên trong hàm lUpdate gồm mainLoop là phần được chạy trong mỗi khung hình còn phần trước đó để khởi tạo vị trí và màu sắc cho chương trình.
* Trong mainLoop, ban đầu sẽ xóa toàn bộ màn hình và gọi hàm prFull để in chữ PTIT với vị trí đã được tính toán. Sau đó tiếp tục tính toán để xử lí 3 phần vị trí có tọa độ là (x, y) và màu sắc của chữ PTIT.

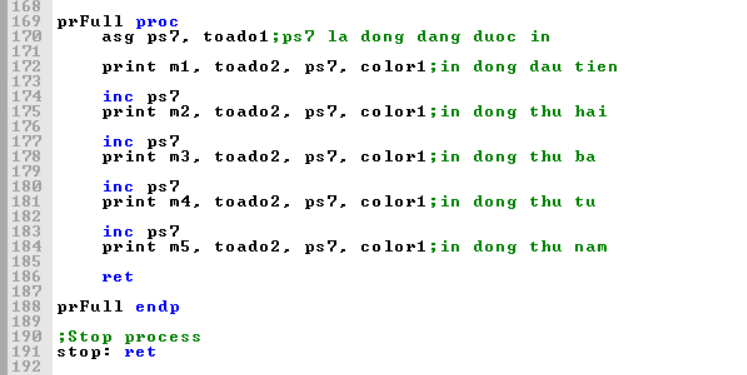


***Hình 9: Hàm IUpdate***



***Hình 10: Hàm IUpdate (tiếp)***

* Hàm c0, c0stage, c1, c1stage, my0, my0stage, my1, my1stage, mx0, mx0stage, mx1, mx1stage:
* c0: Tăng giá trị màu đang sử dụng hiện tại để đổi thành màu sắc tiếp theo (tối đa 10 loại màu) nếu đến màu cuối cùng, sẽ nhảy đến hàm c0stage.
* c0stage: Giảm giá trị màu đang sử dụng hiện tại xuống và thay đổi trạng thái tăng giá trị màu thành giảm giá trị màu để in ngược các loại màu so với ban đầu.
* c1: Giảm giá trị màu đang sử dụng hiện tại để đổi thành màu sắc tiếp theo (tối đa 10 loại màu) nếu đến màu cuối cùng, sẽ nhảy đến hàm c1stage.
* c1stage: Tăng giá trị màu đang sử dụng hiện tại xuống và thay đổi trạng thái giảm giá trị màu thành tăng giá trị màu để in ngược các loại màu so với ban đầu.
* Cách thực hiện và ý tưởng sẽ tương tự với các hàm *my0*, *my0stage*, *my1*, *my1stage*, *mx0*, *mx0stage*, *mx1*, *mx1stage*. Với sự khác biệt là điều kiện giá trị tọa độ x, y giới hạn sẽ khác với giá trị số lượng màu tối đa.
* Hàm prFull:
* Có công dụng để in chữ PTIT bắt đầu tại vị trí (*toado2*, *toado1*) với màu *color3* (toado1,2,3 được tính toán trước từ trong hàm lUpdate)
* Hàm lần lượt in từng hàng ngang của chữ PTIT.



***Hình 11: Hàm prFull***

* Giải thích về cách in ra được màu sắc và xóa các ký tự cũ trên màn hình:
* Thông thường chúng ta sẽ dùng ngắt 21h để gửi tín hiệu đến vi xử lý để yêu cầu một mệnh lệnh nào đó trong chương trình của chúng ta, nhưng với 21h sẽ có một số giới hạn.
* Thế nên để minh họa được chính xác nhất màu sắc của đèn LED ngoài đời thì bọn em đã sử dụng ngắt 10h, thì lệnh “INT 10h” được sử dụng để gọi ngắt mềm 10h và hiển thị ký tự muốn hiển thị với màu sắc chỉ định.
* Cũng như nhờ ngắt 10h, thanh ghi ah được thiết lập để chỉ định chức năng xóa màn hình (chức năng 06h). Thanh ghi al có thể có giá trị bất kỳ khi sử dụng chức năng này. Thanh ghi bh chứa mã màu nền (trong trường hợp này, màu đen được biểu diễn bằng giá trị 07h). Thanh ghi cx và dx được sử dụng để chỉ định vùng diện tích cần xóa, trong trường hợp này là toàn bộ màn hình.
* **Thứ tự thực hiện:**
* Bước 1: In ra màn hình phần giới thiệu trò chơi.
* Bước 2: Define các biến ps1, ps2, ps3, ps4, ps5, ps6, ps7. Khởi tạo text map để có mô hình in ra chữ PTIT.
* Bước 3: Chương trình nhảy vào hàm mainLoop.
  + Bước 3.1: Gọi hàm clrscr để xóa bỏ mọi thứ đã in trên màn hình.
  + Bước 3.2: Gọi hàm prFull để in ra chữ PTIT với các đặc điểm và thuộc tính hiện tại.
  + Bước 3.3: Kiểm tra trạng thái tăng hay giảm của bảng màu sắc, và tọa độ (x, y) của chữ PTIT bằng cmp với các biến trạng thái ps4, ps5, ps6 để sử dụng đúng chính xác hàm (*c0*, *c1*, *my0*, *my1*, *mx0*, *mx1*) thay đổi đặc điểm và thuộc tính của chữ PTIT.
  + Bước 3.4: Nhảy vào hàm mainLoop, quay trở lại bước 3.1.

## Miêu tả giao diện chương trình

* Khi bắt đầu khởi tạo chương trình, chữ “**PTIT**” sẽ di chuyển như sau:
* Xuất phát từ phía trên cùng bên trái của màn hình.
* Di chuyển theo chiều từ trên xuống dưới, và từ phải qua trái cùng lúc.
* Và chữ “PTIT” sẽ thay đổi chiều di chuyển khi:
* Chữ tiếp xúc với phía dưới của màn hình thì chiều di chuyển sẽ đổi thành từ phía dưới đi lên phía trên và ngược lại.
* Chữ tiếp xúc với phía bên phải của màn hình thì chiều di chuyển sẽ đổi thành từ bên phải sang bên trái và ngược lại.



***Hình 12: Led chạy và đang ở trạng thái màu đỏ***



***Hình 13: Led chạy và đang ở trạng thái màu xanh lá***

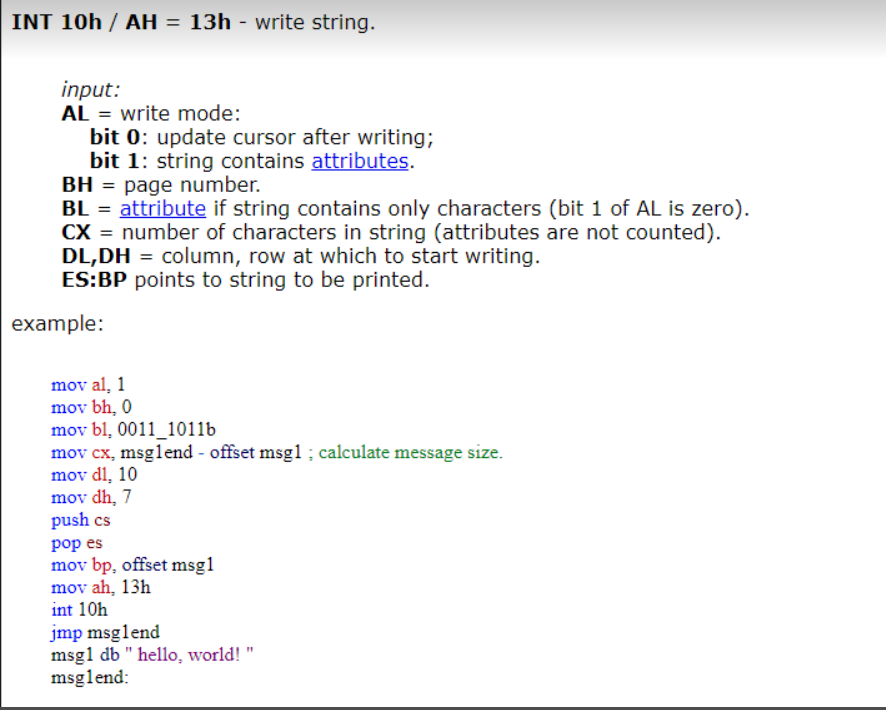


***Hình 14: Led chạy và đang ở trạng thái màu xanh dương***



***Hình 15: Led chạy và ở trạng thái màu xám***

# Tài liệu tham khảo



* <https://yassinebridi.github.io/asm-docs/8086_bios_and_dos_interrupts.html#int10h_13h>
* Phần lý thuyết miêu tả chương trình: https://youtu.be/GCHmyCTLz5s

# Đánh giá về sự đóng góp của các thành viên

|  |  |  |
| --- | --- | --- |
| **Tên thành viên** | **Mã số sinh viên** | **Đóng góp** |
| Trần Đình Lâm Anh | B21DCCN013 | 20% |
| Nguyễn Việt Khiêm | B21DCCN468 | 20% |
| Nguyễn Thị Lan | B21DCCN818 | 20% |
| Trương Công Tuấn Thành | B21DCCN681 | 20% |
| Nguyễn Đăng Huy | B21DCCN062 | 20% |