

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



ĐỒ ÁN THIẾT KẾ LUẬN LÝ

Báo cáo

Mô phỏng hệ thống đèn giao thông bằng kit phát triển BKIT PIC

GVHD: ThS. Phan Đình Thế Duy
SV: Lê Khắc Minh Đăng - 1810109

TP. HỒ CHÍ MINH, THÁNG 7/2020



Tóm tắt đề tài

Đề tài được giao bởi giảng viên hướng dẫn - Thầy Phan Định Thế Duy - và thực hiện nhằm ứng dụng các kiến thức đã được học trong các môn học *Hệ thống số*, *Kỹ thuật lập trình*, *Linh kiện & mạch điện tử* và *Thực tập phần cứng máy tính* để thực hiện việc thiết kế một sản phẩm có tính thực tế là hệ thống điều khiển đèn giao thông ở ngã tư với các tính năng thiết thực.

Quá trình thực hiện đề tài bao gồm ba giai đoạn chính. Giai đoạn đầu tiên là thiết kế bản bố trí linh kiện trên PCB dựa trên sơ đồ nguyên lý được cung cấp sẵn sử dụng phần mềm *Altium Designer*. Giai đoạn thứ hai là hàn tay các linh kiện lên mạch PCB và kiểm tra hoạt động của mạch. Giai đoạn cuối cùng bao gồm việc học về lập trình vi điều khiển PIC trên phần mềm *MPLAB X IDE*, nghiên cứu các mã nguồn mẫu và sau đó hiện thực đề tài theo phương pháp thiết kế dùng biểu đồ trạng thái với các tính năng, cải tiến tùy chọn.

Về phần cứng, sản phẩm bao gồm kit phát triển BKIT PIC đã được hàn và hoạt động tốt và một mạch PCB được làm bằng tay dùng để mô phỏng hệ thống đèn giao thông. Về phần mềm, sản phẩm là một chương trình được nạp vào kit phát triển để điều khiển hệ thống theo ý tưởng của thiết kế.



Mục lục

1 Giới thiệu đề tài	5
1.1 Tên đề tài	5
1.2 Yêu cầu của đề tài	5
1.3 Ý nghĩa thực tiễn	5
2 Cơ sở lý thuyết	6
2.1 Tổng quan về vi điều khiển PIC	6
2.2 Giới thiệu các module trong thiết kế	7
2.3 Phương pháp thiết kế bằng biểu đồ trạng thái	8
3 Xây dựng mạch BKIT PIC và mạch mô phỏng	9
3.1 Kit phát triển BKIT PIC	9
3.2 Mạch mô phỏng	9
4 Các tính năng của hệ thống điều khiển	11
4.1 Các chế độ hoạt động	11
4.2 Xác lập thời gian đếm ngược	11
4.3 Tính năng khác	11
5 Thiết kế chương trình	12
5.1 Các module được sử dụng	12
5.2 Thiết kế chương trình chính	12
6 Cách thức vận hành hệ thống	14
6.1 Nạp chương trình	14
6.2 Chọn chế độ hoạt động	14
6.3 Xác lập thời gian đếm ngược	16
7 Sản phẩm hoàn chỉnh	17
8 Tổng kết	18



8.1	Kết quả đạt được	18
8.2	Vấn đề còn tồn tại	18
8.3	Hướng phát triển	19



Danh sách hình vẽ

1	Kit phát triển BKIT PIC	9
2	Bản thiết kế mạch mô phỏng	10
3	Cấu trúc chương trình	12
4	Biểu đồ trạng thái của chương trình chính	13
5	Màn hình LCD và mạch mô phỏng ở trạng thái đếm tự động	15
6	Màn hình LCD và mạch mô phỏng ở trạng thái chuyển bằng tay	15
7	Màn hình LCD và mạch mô phỏng ở trạng thái chờ	16
8	Màn hình LCD khi đang chỉnh thời gian xanh cho đèn 1	16
9	Sản phẩm cuối cùng	17



1 Giới thiệu đề tài

1.1 Tên đề tài

Mô phỏng hệ thống đèn giao thông bằng kit phát triển BKIT PIC.

1.2 Yêu cầu của đề tài

- Thiết kế bản bố trí linh kiện của kit phát triển BKIT PIC dựa trên sơ đồ nguyên lý có sẵn bằng phần mềm Altium Designer.
- Hàn tay các linh kiện lên mạch và kiểm tra hoạt động của mạch.
- Tìm hiểu về vi điều khiển PIC và cách lập trình kit phát triển bằng phần mềm MPLAB X IDE thông qua bài giảng và các mã nguồn mẫu được cung cấp.
- Dùng các kiến thức đã tìm hiểu để thiết kế và hiện thực chương trình điều khiển hệ thống đèn giao thông với các tính năng tùy chọn.
- Dùng kiến thức đã học ở môn học *Thực tập phần cứng máy tính* để làm một bản mạch nhỏ mô phỏng hệ thống đèn giao thông ở ngã tư.

1.3 Ý nghĩa thực tiễn

Lập trình hệ thống nhúng là một kỹ năng thiết yếu không thể thiếu của ngành Kỹ thuật máy tính. Ngày nay, hệ thống nhúng ngày càng phát triển mạnh trong tất cả các lĩnh vực từ điện toán đến công nghiệp, y tế, giáo dục, giao thông vận tải, v.v. Do đó, đề tài đồ án này giúp cho sinh viên làm quen với việc lập trình hệ thống nhúng cho các hệ thống thực tế.

Về mặt kiến thức, đề tài áp dụng kiến thức của nhiều môn học trước như *Hệ thống số*, *Kỹ thuật lập trình*, *Linh kiện & mạch điện tử*, *Thực tập phần cứng máy tính* để hiện thực tạo ra sản phẩm, củng cố lại kiến thức cho sinh viên.

Về mặt thực tiễn, hệ thống đèn giao thông là một hệ thống điều khiển vô cùng phổ biến và quan trọng trong giao thông vận tải. Việc hiện thực mô phỏng hệ thống này trên kit phát triển giúp đưa ra các tính năng mới mẻ, đơn giản mà hiệu quả trong việc hệ thống đèn giao thông, có thể ứng dụng được trong thực tế.



2 Cơ sở lý thuyết

2.1 Tổng quan về vi điều khiển PIC

PIC là một dòng vi điều khiển được phát triển bởi *Microchip Technology*, kế thừa từ PIC1650 nguyên bản được phát triển bởi General Instrument's Microelectronics Division. Ban đầu PIC viết tắt cho *Peripheral Interface Controller*, nghĩa là bộ điều khiển giao diện ngoại vi, và hiện tại được mở rộng thành *Programmable Intelligent Computer* - máy tính thông minh khả lập trình. Khả năng phần cứng của PIC giao động từ 6, 8 đến 144 chân SMD, với các chân I/O, module ADC và DAC, và các cổng giao tiếp như UART, I2C, CAN và thậm chí là USB.[1]

Các tính năng phần cứng của PIC có thể kể đến như:

- Bộ nhớ Flash là bộ nhớ chương trình
- SRAM là bộ nhớ dữ liệu
- Bộ nhớ EEPROM khả lập trình
- Chế độ tiết kiệm năng lượng
- Bộ định thời giám sát (Watchdog timer)
- Nhiều chế độ dao động thạch anh nội, hoặc thạch anh ngoại.
- Các chân GPIO
- Bộ định thời 8/16/32 bit
- Module ADC, PWM
- Các giao diện giao tiếp và truyền dữ liệu.
- V.v.

Phần mềm dùng để lập trình và phát triển vi điều khiển PIC là một gói các phần mềm IDE miễn phí MBLAB X. Chương trình từ phần mềm được nạp vào vi điều khiển và gỡ lỗi bằng các dòng mạch nạp MPLAB và PICkit.



2.2 Giới thiệu các module trong thiết kế

Sau đây là khái niệm về các module được sử dụng trong thiết kế hệ thống này, các khái niệm được tóm tắt từ *Hướng dẫn sử dụng mạch BKIT PIC* của BKIT Hardware Club.[2]

Các nguồn ngắn và bộ định thời:

Dòng PIC18F4620 trên kit có nhiều nguồn ngắn và 2 mức ưu tiên ngắn (ưu tiên thấp và ưu tiên cao). Vector ngắn có mức ưu tiên cao có địa chỉ 0x08 còn ngắn có mức ưu tiên thấp có địa chỉ 0x18. Khi hàm phục vụ ngắn quang cho ngắn ưu tiên thấp đang xảy ra, ngắn ưu tiên cao xảy ra sẽ tạm dừng ngắn ưu tiên thấp và phục vụ cho ngắn ưu tiên cao.

Khi các nguồn ngắn có cùng độ ưu tiên, chúng sẽ cùng nhảy đến 1 địa chỉ ngắn. Hàm phục vụ ngắn quang cần phải kiểm tra tất cả các cờ để xác định nguồn ngắn nào đang gây ra ngắn. Cờ ngắn cần được xóa để tránh hiện tượng ngắn đê quy, vi điều khiển sẽ lặp vô tận trong hàm ngắn cho đến khi tràn stack.

Bộ định thời timer3 có 2 chế độ 16 bit hoặc 8 bit. Clock cấp cho timer3 có thể là clock nội hoặc clock ngoại (lấy từ T3CKI). Ngoài ra timer3 còn có bộ Prescaler để chia tần số clock.

Khi hoạt động ở chế độ 16 bit, giá trị của bộ định thời (counter)/ bộ đếm timer3 được ghi vào 2 thanh ghi TMR3H và TMR3L. Ngược lại, ở chế độ 8 bit, giá trị đếm được lưu trong thanh ghi TMR3L.

Khi timer3 đếm tràn từ FF:FF (chế độ 16 bit) hoặc FF (chế độ 8 bit) lên 0, cờ TMR3IF sẽ được bật lên 1 và gây ra ngắn nếu các bit cho phép ngắn (ngắn toàn cục, ngắn timer3) được set lên 1.

Ma trận phím:

Ma trận phím là cách kết nối các phím theo hàng và cột. Cách kết nối như vậy sẽ tiết kiệm được tài nguyên của vi điều khiển. Trong ma trận phím 4x4, ta có 16 phím và 8 tín hiệu điều khiển. Nếu măc 16 phím này như các phím đơn, ta phải cần đến 16 tín hiệu điều khiển.

Để xác định vị trí của phím nào được nhấn, ta sẽ duyệt qua tất cả các hàng và xét các phím hàng đó. Quy ước tín hiệu DEACTIVE là mức 1 và tín hiệu ACTIVE là mức 0, từng bước giải mã ma trận phím như sau:

- Cho tín hiệu các cột là DEACTIVE
- Cài đặt tín hiệu tại các cột là INPUT
- Cho tín hiệu các hàng là DEACTIVE



- Cài đặt tín hiệu các hàng là OUTPUT
- Lặp qua các hàng, cho tín hiệu hàng đang xét là ACTIVE và xét các cột trong hàng đang cho ACTIVE, cột nào có tín hiệu ACTIVE là phím tương ứng được nhấn.
- Kết thúc vòng lặp

Màn hình LCD 16x2

Màn hình LCD 16x2 dùng để hiển thị ký tự. Bên trong nó bao gồm các thành phần bộ nhớ DDRAM, CGRAM và CGROM.

DDRAM lưu trữ mã ký tự hiển thị ra màn hình. Mã này giống với mã ASCII. Có tất cả 80 ô nhớ DDRAM. Vùng hiển thị tương ứng với cửa sổ gồm 16 ô nhớ hàng đầu tiên và 16 ô nhớ hàng thứ hai. Chúng ta có thể tạo hiệu ứng dịch chữ bằng cách sử dụng lệnh dịch, khi đó cửa sổ hiển thị sẽ dịch đem lại hiệu ứng dịch chữ.

CGRAM lưu trữ tám mẫu ký tự do người dùng định nghĩa.

CGROM lưu trữ các ký tự có thể hiển thị trên LCD.

LCD có 2 chế độ 8 bit và 4 bit. Ở chế độ 8 bit, ta dùng toàn bộ 8 chân D0-D7 để giao tiếp. Ở chế độ 4 bit, ta chỉ dùng 4 bit cao D4-D7 để giao tiếp với LCD. Dữ liệu gửi cho LCD ở chế độ này bao gồm 4 bit cao gửi trước, sau đó sẽ đến 4 bit thấp.

2.3 Phương pháp thiết kế bằng biểu đồ trạng thái

Biểu đồ trạng thái là phương pháp tốt nhất để thiết kế các hệ thống tự động có tính tuần tự. Ý tưởng của phương pháp này là mô hình hóa hoạt động của hệ thống bởi một biểu đồ các trạng thái và các cạnh chuyển trạng thái, với mỗi trạng thái có ngõ ra tương ứng và mỗi cạnh chuyển trạng thái ứng với một hoặc nhiều tín hiệu ngõ vào.

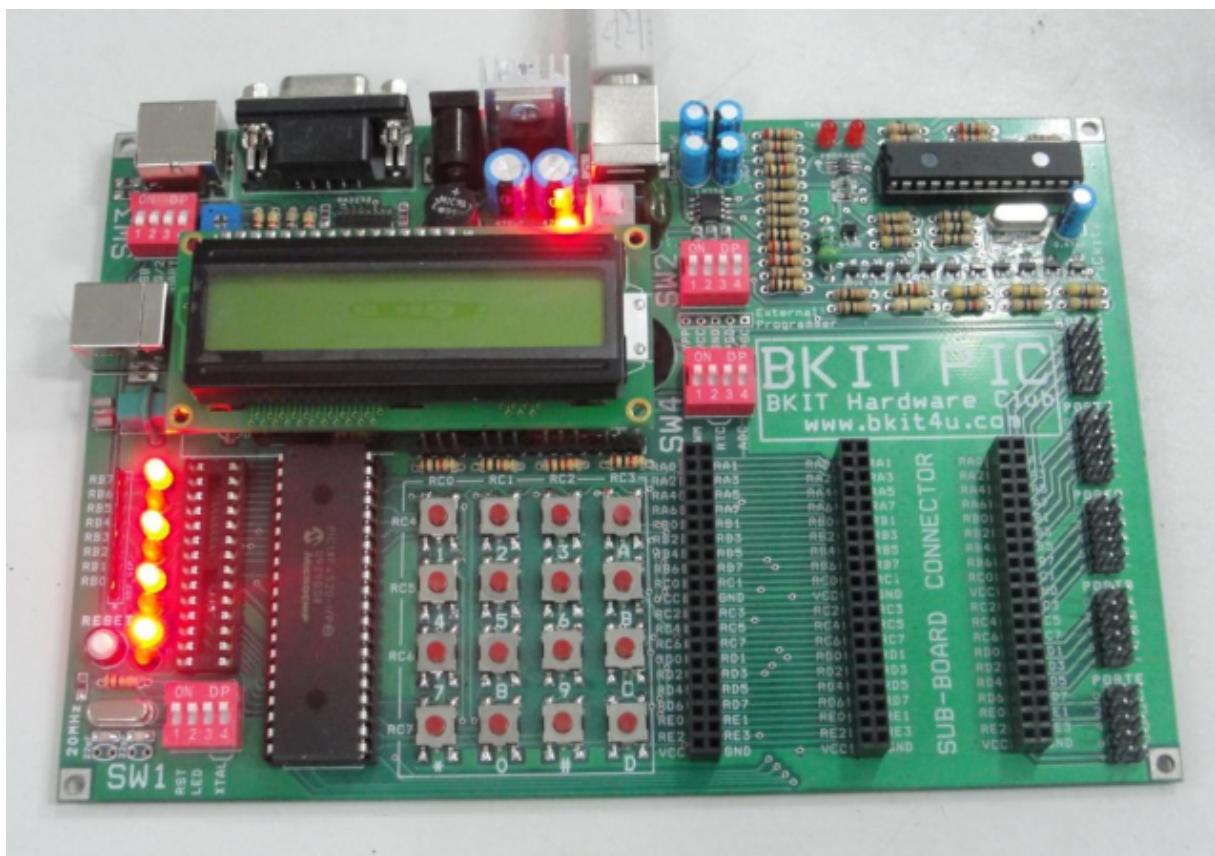
Việc hiện thực từ biểu đồ trạng thái sang chương trình gồm 3 bước:

1. Xây dựng phần khung chương trình có đầy đủ các trạng thái với cấu trúc **switch**.
2. Hiện thực các hành động hay hoạt động của hệ thống ở từng trạng thái. Hành động hay hoạt động ở từng trạng thái thường gắn liền với tên của trạng thái.
3. Hiện thực điều kiện chuyển trạng thái.

3 Xây dựng mạch BKIT PIC và mạch mô phỏng

3.1 Kit phát triển BKIT PIC

Kit phát triển BKIT PIC được thiết kế và phát triển bởi *BKIT Hardware Club*, dựa trên vi điều khiển **PIC18F4620**[3]. Trên kit được tích hợp sẵn nhiều cổng kết nối, bộ nguồn, mạch nạp *PICKit2* và nhiều module khác như màn hình LCD 16x2, dãy LED đơn, ma trận phím 4x4, ADC, PWM, giao tiếp UART, USB, PS/2, v.v. Các chân tín hiệu của PIC cũng được đưa ra các rào chân để thuận tiện cho việc sử dụng.

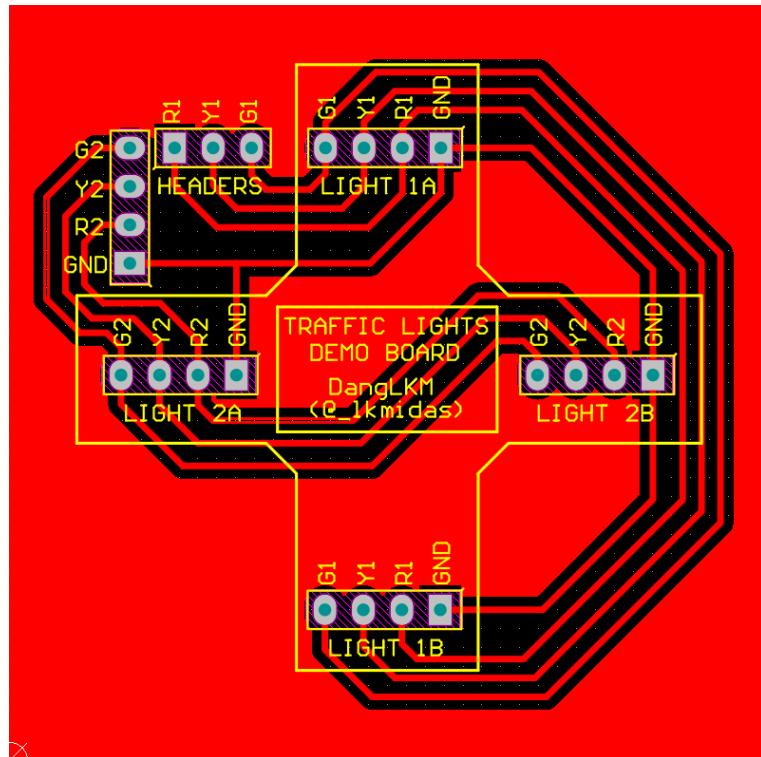


Hình 1: Kit phát triển BKIT PIC

3.2 Mạch mô phỏng

Mạch mô phỏng là một mạch PCB một mặt với các rào chân dùng để cắm dây và các module đèn giao thông[4]. Mạch được làm bằng tay theo phương pháp ủi thủ công đã được học trong môn *Thực tập phần cứng máy tính*. Các bước thực hiện như sau:

- Thiết kế sơ đồ nguyên lý và sắp xếp linh kiện bằng phần mềm Altium Designer. Trong các phần bên dưới của bài cáo cáo, "đèn 1" và "đèn 2" sẽ dùng để chỉ cắp các đèn A và B nằm trên hai đường vuông góc nhau giống như trong hình sau:



Hình 2: Bản thiết kế mạch mô phỏng

- In bản thiết kế PCB ra giấy thuốc với định dạng *Monochrome* (hai màu đen trắng).
- Cưa tấm đồng một mặt theo kích thước tương ứng với thiết kế.
- Chuyển lớp mực trên giấy thuốc sang tấm đồng bằng bàn ủi điện, ủi mạnh tay tờ giấy trên tấm đồng trong khoảng 10 đến 15 phút.
- Gỡ tờ giấy ra và ngâm tấm đồng trong dung dịch sắt(III) clorua. Phần đồng không được bao phủ bởi mực sẽ bị dung dịch hòa tan trong vài phút.
- Dùng bùi nhùi sắt chà lớp mực ra khỏi tấm đồng và khoan các lỗ tương ứng.
- Hàn các rào chân lên mạch.



4 Các tính năng của hệ thống điều khiển

4.1 Các chế độ hoạt động

Hệ thống có thể hoạt động ở 3 chế độ khác nhau tùy chỉnh, bao gồm: chế độ tự động đếm ngược, chế độ chuyển đèn bằng tay và chế độ chờ.

1. Trong chế độ tự động đếm ngược, các đèn sẽ chuyển màu khi thời gian đếm đến 0. Thời gian xanh, vàng của các đèn trên hai đường vuông góc có thể khác nhau, và thời gian đỏ của đèn này luôn bằng tổng thời gian vàng và xanh của đèn kia. Đây là chế độ mặc định của hệ thống khi khởi động.
2. Trong chế độ chuyển đèn bằng tay, các đèn sẽ chỉ chuyển màu khi ta nhấn phím chuyển. Hai chế độ vừa nêu có thể chuyển qua lại tương ứng bởi một nút nhấn.
3. Trong chế độ chờ, cả bốn đèn ở bốn góc ngã tư sẽ chỉ chớp đèn vàng. Từ 2 chế độ trên có thể chuyển sang chế độ này bởi một nút nhấn khác.

4.2 Xác lập thời gian đếm ngược

Thời gian đếm ngược của các đèn trong chế độ tự động đếm ngược có thể được thay đổi theo hai cách: dựa theo hai bộ thời gian được xác lập sẵn hoặc cài đặt thời gian tùy ý.

1. Trong hệ thống có xác lập hai bộ thời gian, chúng có thể được tùy chọn bằng hai nút nhấn. Chức năng này dùng để giả lập một tính năng trong thực tế, đó là khả năng thay đổi thời gian đếm tùy thuộc vào một điều kiện đặc biệt nào đó, chẳng hạn như vào giờ cao điểm, giờ khuya hay vào lúc một chiều có nhiều xe hơn hẳn chiều còn lại, v.v.
2. Ngoài ra, thời gian đếm có thể được cài đặt lại một cách tùy ý. Việc cài đặt thời gian sẽ được nhập vào từ các phím số, mỗi giá trị thời gian là một số tự nhiên hai chữ số. Ngoài ra, thời gian đếm của đèn xanh được quy định trong khoảng từ 10 đến 99 giây và đèn vàng là từ 3 đến 10 giây.

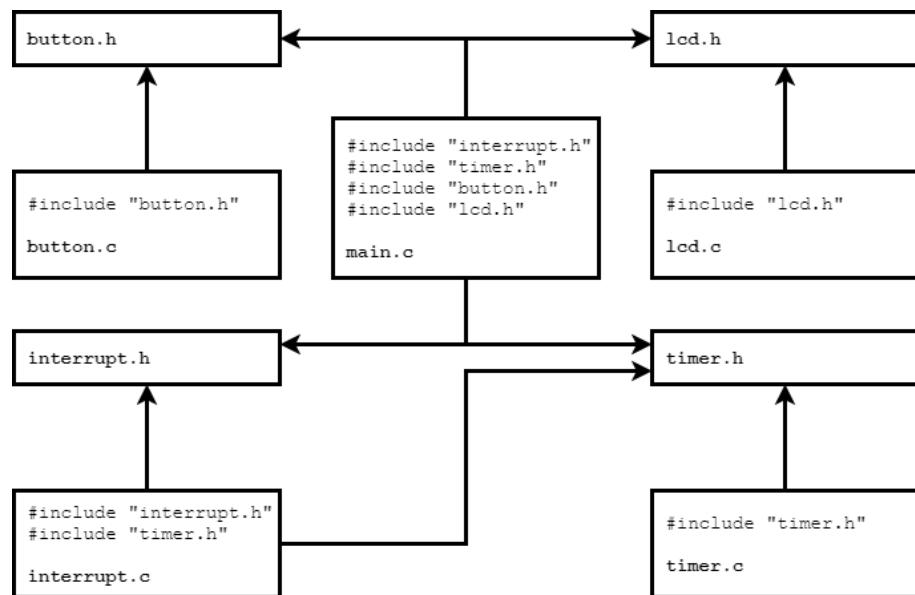
4.3 Tính năng khác

Ngoài các tính năng cơ bản nêu trên, hệ thống còn có tính năng chuyển đổi ngay lập tức từ xanh sang đỏ, đỏ sang xanh bằng một nút nhấn. Chức năng này dùng để giả lập trường hợp thực tế khi có sự cố hoặc có xe ưu tiên tiến đến ngã tư khi đèn đỏ. Trong trường hợp thực tế, nút nhấn điều khiển sẽ có thể được thay bằng một cảm biến hay máy ghi hình.

5 Thiết kế chương trình

5.1 Các module được sử dụng

Các module của kit BKIT PIC được sử dụng trong chương trình bao gồm: bộ ngắt, bộ định thời, ma trận phím 4x4 và màn hình LCD. Các module trên được hiện thực trong các file `interrupt.c`, `timer.c`, `button.c` và `lcd.c`. Các file trên đã được cung cấp sẵn trong các bài giảng, tuy nhiên các file được sử dụng trong đề tài này có được chỉnh sửa về phong cách lập trình và cách đặt tên biến, tên hàm. Cấu trúc chương trình có thể được tóm tắt bằng sơ đồ sau:



Hình 3: Cấu trúc chương trình

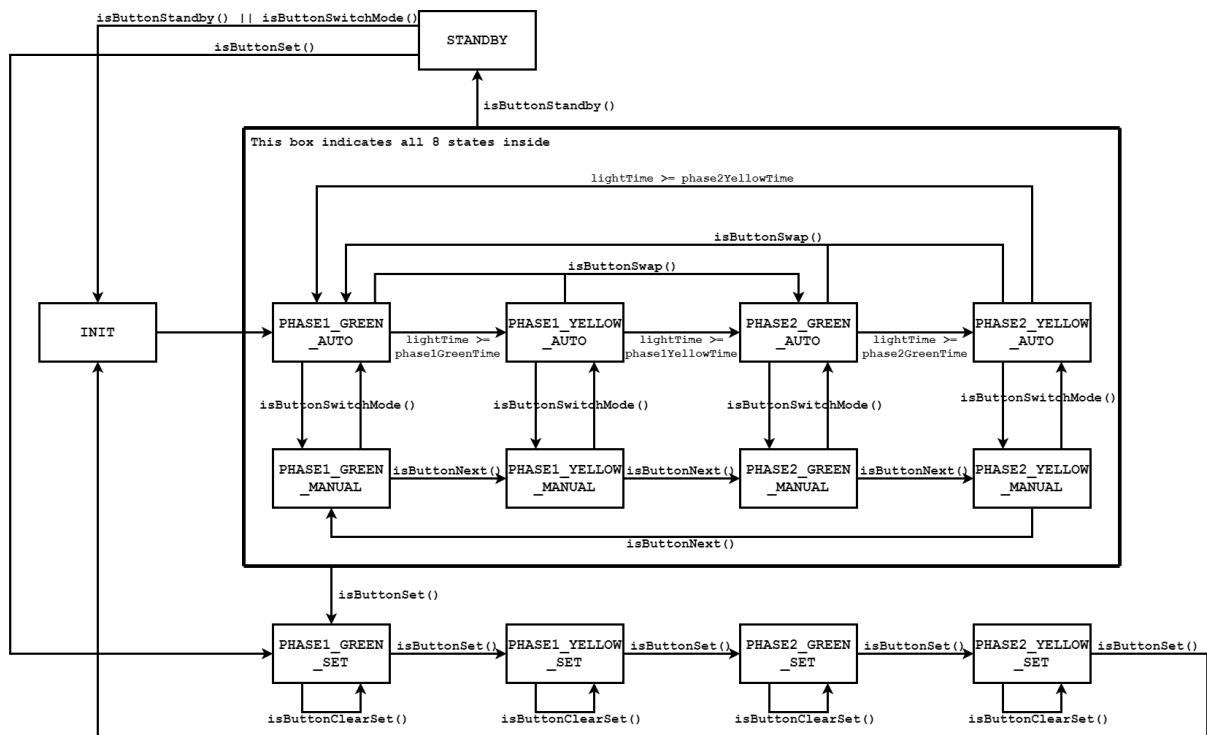
5.2 Thiết kế chương trình chính

Chương trình chính được hiện thực trong file `main.c` bằng phương pháp thiết kế dựa trên mô hình biểu đồ trạng thái. Mô hình thiết kế này giúp đảm bảo hệ thống hoạt động ổn định và không xảy ra các hành vi bất định ngoài ý muốn. Biểu đồ trạng thái của thiết kế hệ thống đèn giao thông bao gồm các trạng thái sau:

1. INIT: trạng thái khởi tạo.
2. STANDBY: trạng thái của chế độ chờ.
3. PHASE_1_GREEN_AUTO: trạng thái đèn 1 xanh của chế độ đếm tự động.

4. PHASE_1_YELLOW_AUTO: trạng thái đèn 1 vàng của chế độ đếm tự động.
 5. PHASE_2_GREEN_AUTO: trạng thái đèn 2 xanh của chế độ đếm tự động.
 6. PHASE_2_YELLOW_AUTO: trạng thái đèn 2 vàng của chế độ đếm tự động.
 7. PHASE_1_GREEN_MANUAL: trạng thái đèn 1 xanh của chế độ chuyển bằng tay.
 8. PHASE_1_YELLOW_MANUAL: trạng thái đèn 1 vàng của chế độ chuyển bằng tay.
 9. PHASE_2_GREEN_MANUAL: trạng thái đèn 2 xanh của chế độ chuyển bằng tay.
 10. PHASE_2_YELLOW_MANUAL: trạng thái đèn 2 vàng của chế độ chuyển bằng tay.
 11. PHASE_1_GREEN_SET: trạng thái xác lập thời gian xanh của đèn 1.
 12. PHASE_1_YELLOW_SET: trạng thái xác lập thời gian vàng của đèn 1.
 13. PHASE_2_GREEN_SET: trạng thái xác lập thời gian xanh của đèn 2.
 14. PHASE_2_YELLOW_SET: trạng thái xác lập thời gian vàng của đèn 2.

Dưới đây là biểu đồ trạng thái chi tiết của thiết kế gồm các trạng thái và các tín hiệu chuyển trạng thái tương ứng:



Hình 4: Biểu đồ trang thái của chương trình chính



6 Cách thức vận hành hệ thống

6.1 Nạp chương trình

Để có thể nạp chương trình, đầu tiên ta cần gạt các công tắc ở SW2 và các công tắc XTAL, RST về ON. Ta cũng cần gạt công tắc LED về OFF và LCD về ON để sử dụng màn hình LCD thay vì dãy LED ở PORTB. Sau đó ta dùng dây nối để kết nối mạch đèn mô phỏng với kit thông qua các rào chân ở PORTD. Cách kết nối là các chân 2, 4, 6 ứng với đèn xanh, vàng, đỏ của đèn 1 và 3, 5, 7 ứng với xanh, vàng, đỏ của đèn 2.

Sau khi đã chuẩn bị xong, việc nạp chương trình sẽ được thực hiện bởi phần mềm MPLAB X IDE thông qua kết nối USB với kit.

6.2 Chọn chế độ hoạt động

Sau khi khởi động, hệ thống sẽ vào chế độ đếm tự động, thời gian đếm được hiện trên màn hình LCD. Trong chế độ này, ta có thể sử dụng các phím sau:

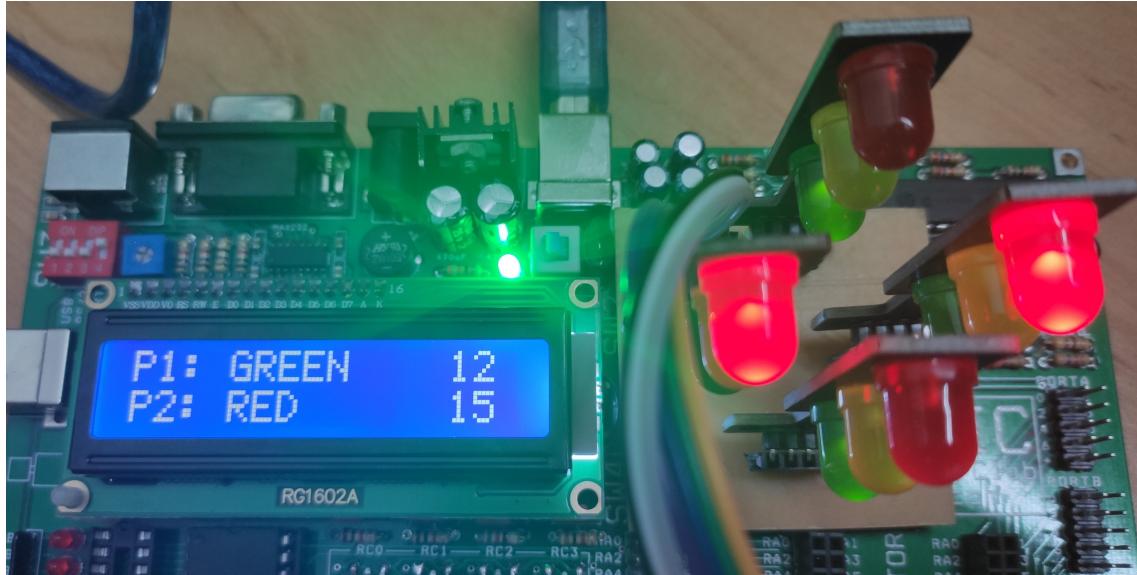
- Phím A để chuyển sang chế độ chuyển đèn bằng tay.
- Phím # để chuyển sang chế độ chờ.
- Phím C và D để chọn một trong hai bộ thời gian cài sẵn.
- Phím * để tùy chỉnh thời gian.
- Phím 0 để ngay lập tức chuyển đèn từ xanh sang đỏ, đỏ sang xanh.

Trong chế độ chuyển đèn bằng tay, thay vì các số đếm ngược, chữ M viết tắt cho *Manual* sẽ hiện trên màn hình LCD. Các phím sau được sử dụng:

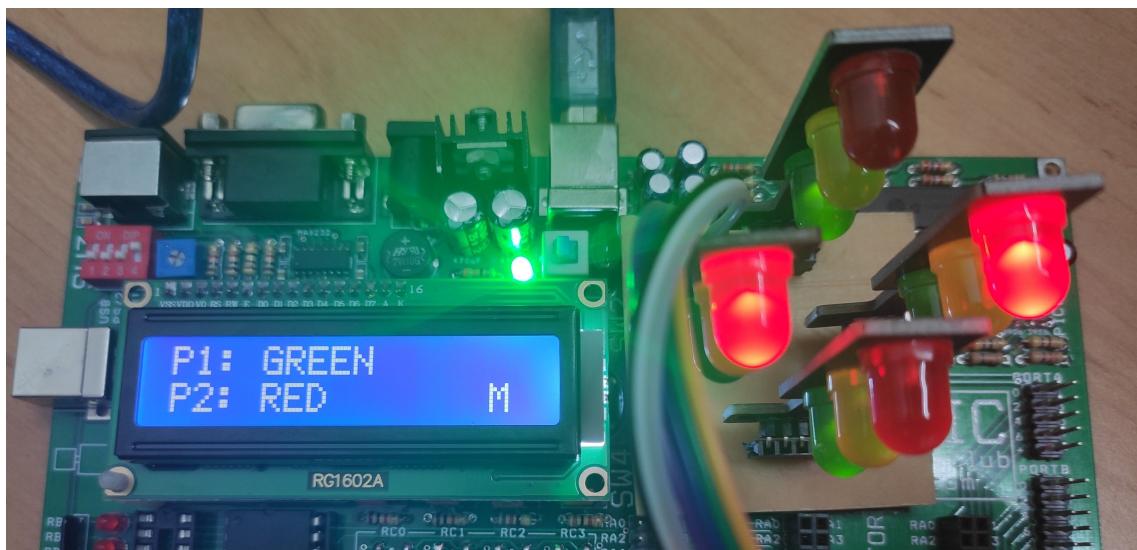
- Phím A để chuyển về chế độ đếm tự động.
- Phím B để chuyển đèn.
- Phím # để chuyển sang chế độ chờ.

Trong chế độ chờ, ta chỉ có thể dùng phím * để tùy chỉnh thời gian cho chế độ đếm tự động hoặc các phím #, A để về chế độ đếm tự động.

Đây là một vài hình chụp minh họa:



Hình 5: Màn hình LCD và mạch mô phỏng ở trạng thái đếm tự động



Hình 6: Màn hình LCD và mạch mô phỏng ở trạng thái chuyển bằng tay



Hình 7: Màn hình LCD và mạch mô phỏng ở trạng thái chờ

6.3 Xác lập thời gian đếm ngược

Trong chế độ đếm tự động, như đã nói ở trên, ta có thể chọn một trong hai bộ thời gian sẵn có với các phím C và D, thời gian cụ thể của từng bộ theo thứ tự đèn 1 xanh - đèn 1 vàng - đèn 2 xanh - đèn 2 vàng lần lượt là 35 - 3 - 10 - 3 và 25 - 3 - 20 - 3.

Phím * được dùng để vào trạng thái tùy chỉnh thời gian. Thời gian sẽ được chỉnh theo thứ tự tương tự như trên: đèn 1 xanh - đèn 1 vàng - đèn 2 xanh - đèn 2 vàng. Việc nhập số được thực hiện thông qua 10 phím số từ 0 đến 9. Trên màn hình LCD sẽ hiển thị một con trỏ nằm ngang (_) nhấp nháy ứng với chữ số cần nhập. Sau khi nhập đủ 2 chữ số, con số sẽ nhấp nháy chờ xác nhận từ người dùng. Nếu đã nhập số chính xác, ta có thể nhấn phím * để nhập số tiếp theo, hoặc ngược lại, ta có thể nhấn phím A để nhập lại số mới.

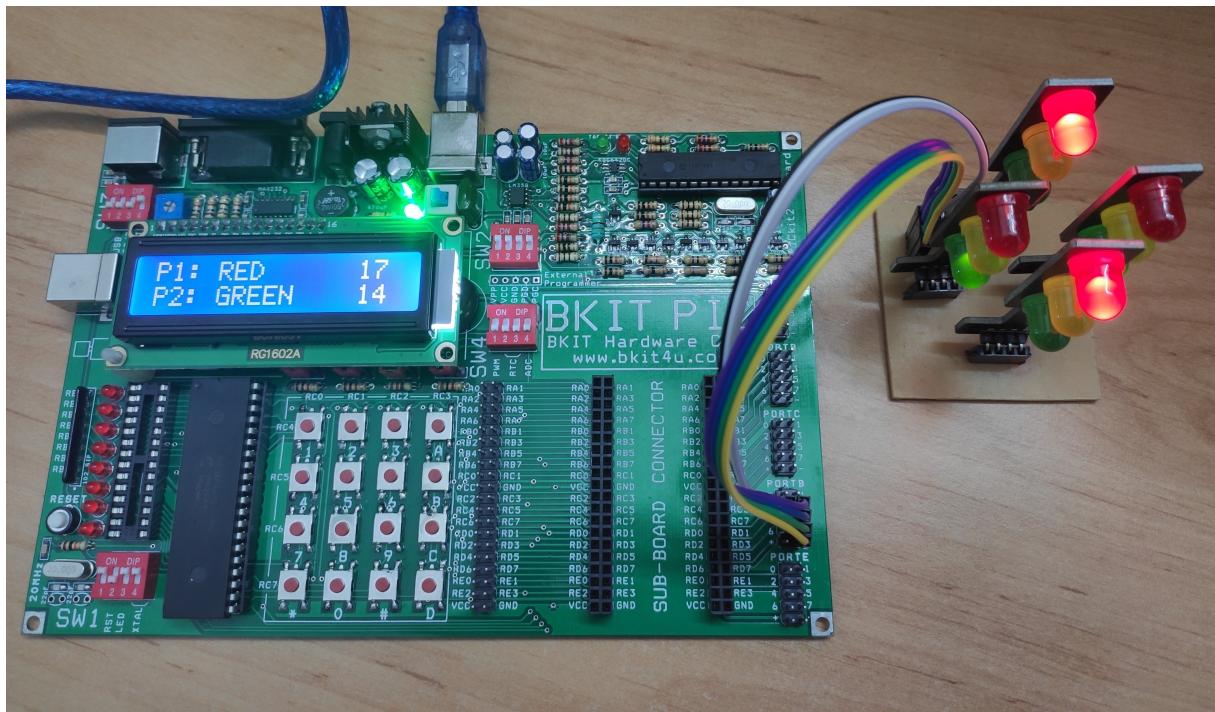
Bên dưới là hình chụp minh họa:



Hình 8: Màn hình LCD khi đang chỉnh thời gian xanh cho đèn 1

7 Sản phẩm hoàn chỉnh

Hình ảnh chụp toàn bộ sản phẩm hoàn chỉnh:



Hình 9: Sản phẩm cuối cùng

Đoạn video ngắn giới thiệu toàn bộ chức năng của hệ thống có thể được tìm thấy trong đĩa DVD đính kèm hoặc truy cập tại link <https://youtu.be/Ua0uD4EGTHY> (video được lồng tiếng bằng tiếng Anh).

Các file thiết kế, file mã nguồn cũng có thể được tìm thấy trong đĩa DVD hoặc qua đường link <https://github.com/LKMDang/Traffic-light-system-on-BKIT-PIC>.

Ngoài ra, trong đĩa DVD còn có phiên bản mềm của bài báo cáo này.



8 Tổng kết

8.1 Kết quả đạt được

Phần cứng:

- Tìm hiểu và thành thạo hơn về phần mềm thiết kế mạch *Altium Designer*.
- Thiết kế thành công bản sắp xếp linh kiện của kit phát triển BKIT PIC với *Altium Designer*.
- Thành công trong việc hàn các linh kiện lên mạch BKIT PIC.
- Áp dụng được các kiến thức đã học để tự làm mạch mô phỏng đèn giao thông.

Phần mềm:

- Tìm hiểu và làm quen với việc lập trình vi điều khiển PIC trên *MPLAB X IDE* nói riêng, và lập trình hệ thống nhúng nói chung.
- Học được các kỹ thuật và phương pháp thiết kế và lập trình hệ thống nhúng, đặc biệt là phương pháp thiết kế theo biểu đồ trạng thái.
- Đọc qua và hiểu được các mã nguồn được cung cấp, đồng thời chỉnh sửa để phù hợp với phong cách lập trình của riêng mình.
- Lập trình thành công chương trình điều khiển hệ thống đèn giao thông áp dụng những kiến thức đã học và tự tìm hiểu.

8.2 Vấn đề còn tồn tại

- Mạch BKIT PIC hiện nay hoạt động chưa ổn định, có thể do nhiều ở đường dây nguồn.
- Chưa tận dụng được tối đa các tính năng của kit phát triển do chưa đủ kiến thức để lập trình giao tiếp với các module đó.
- Vì lý do trên, chương trình hiện tại tuy hoàn chỉnh và vận hành ổn định nhưng còn khá đơn giản, ít các tính năng nâng cao.



8.3 Hướng phát triển

- Tiếp tục tìm hiểu sâu và thực hành nhiều hơn với các phần mềm *Altium Designer* và *MPLAB X IDE*.
- Thiết kế lại mạch BKIT PIC với cách sắp xếp linh kiện chuẩn hơn trong việc khử nhiễu và kiểm tra lại độ ổn định của mạch.
- Mở rộng các tính năng cho mạch mô phỏng, tận dụng tối đa các module trên kit như giao tiếp UART, PS/2, RTC, v.v để hiện thực các chức năng nâng cao hơn.



Tài liệu

- [1] *PIC microcontrollers*, từ Wikipedia. Truy cập ngày 14/7 tại https://en.wikipedia.org/wiki/PIC_microcontrollers
- [2] *Hướng dẫn sử dụng kit BKIT PIC: PIC18FManual*, BKIT Hardware Club.
- [3] *PIC18F4620 datasheet*, Microchip Technology. Truy cập ngày 14/7 tại <http://ww1.microchip.com/downloads/en/DeviceDoc/39626e.pdf>
- [4] *Mạch Hiển Thị Led Dèn Giao Thông Traffic Light*, HShop. Truy cập ngày 14/7 tại <https://hshop.vn/products/mach-hien-thi-led-den-giao-thong>