Vietnam National University, Ho Chi Minh City
University of Technology
Faculty of Computer Science and Engineering

Vi xử lý - Vi điều khiển (CO3009)

Hệ thống đèn giao thông hiện thực trên board Nucleo dựa trên vi điều khiển STM32F103RB

Advisor(s): PhD. Lê Trọng Nhân
TA. Cao Tiến Đạt

Student(s): Phạm Hồng My Sa - 2112173
Trịnh Thu Trang - 2213561

Ho Chi Minh City, Ngày 2 tháng 2 năm 2025

Member list

No.	Fullname	ID	Job	Percentage
1	Trịnh Thu Trang	2213561	Base-system & Add-edit-fix system	100%
2	Phạm Hồng My Sa	2112173	Add-edit-fix system & Write report	100%

Bảng 1: Member list & workload

Contents

1 Giới thiệu đề tài

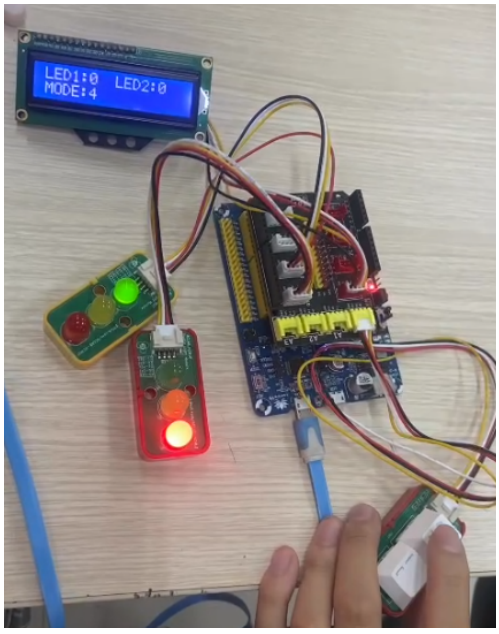
1.1 Mục tiêu

Hiện thực ứng dụng biểu diễn hệ thống đèn giao thông tại ngã tư trên board Nucleo-F103RB, dựa trên vi điều khiển STM32, sử dụng nút nhấn & màn hình hiển thị thông tin.

1.2 Yêu cầu

Hệ thống đảm bảo các yêu cầu cơ bản như sau, chi tiết tùy nhóm định nghĩa:

- Hệ thống được mô phỏng trên mạch thật là **board Nucleo-F103RB** dựa trên vi điều khiển STM32 (STM32F103RB).
- **Đèn giao thông**: Sử dụng 2 **đèn giao thông có 3 màu** RED (đỏ), YELLOW (vàng) và GREEN (xanh) để tượng trưng cho 4 đèn giao thông ở ngã tư.
- **Hiển thị**: Hiển thị thông tin (mode, status, time, led) trên **màn hình LCD 16x2** tượng trưng đèn đếm lùi thời gian trong hệ thống đèn giao thông ở ngã tư.
- **Chế độ**: Chế độ điều khiển đèn giao thông có ít nhất 2 chế độ, gồm chế độ tự động & chế độ thủ công (sử dụng **nút nhấn**).



Hình 1: *Các thành phần phần cứng mô phỏng hệ thống*

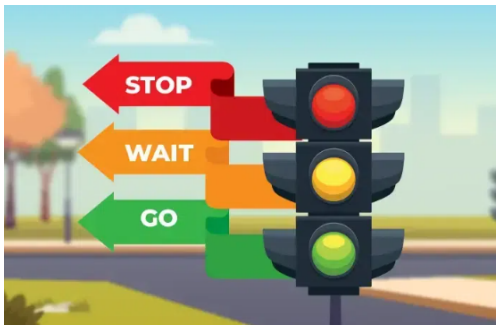
2 Kiến thức nền

Dưới đây là kiến thức cơ bản về hệ thống đèn giao thông ở ngã tư thực tế mà nhóm dựa theo để hiện thực trong bài này.

2.1 Ý nghĩa các thành phần

2.1.1 Đèn 3 màu

- **Đèn đỏ**: Khi gặp đèn đỏ, tất cả các phương tiện đang lưu thông bị cấm đi tiếp.
- **Đèn vàng**: Khi gặp đèn vàng, tất cả các phương tiện phải đi chậm lại.
- **Đèn xanh**: Khi gặp đèn xanh, tất cả các phương tiện được phép đi.



Hình 2: *Đèn 3 màu*

2.1.2 Màn hình hiển thị

Thời gian hiển thị trên màn hình là thời gian còn lại cho đèn đang sáng.

Hình 3: *Đèn đếm lùi (trong hệ thống là màn hình LCD tượng trưng)*

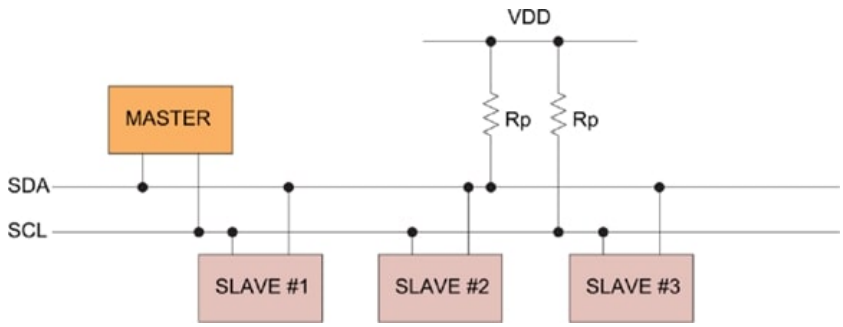
2.2 Quy định điều khiển đèn tín hiệu

- Đèn tín hiệu phải bật từng màu riêng biệt, đèn này tắt mới được bật đèn kia lên, không được bật nhiều màu cùng một lúc.
- Giữa 2 chiều đường, khi chiều A bật đèn đỏ thì lập tức chiều B phải bật ngay đèn xanh và ngược lại.
- Khi chuyển từ xanh-đỏ và đỏ-xanh bắt buộc phải bật qua màu vàng, vì màu vàng đệm giữa 2 màu xanh đỏ.

3 Cơ sở lý thuyết

3.1 Hiển thị LCD: Giao thức I2C

I2C (Inter-Integrated Circuit) là giao thức do Philips Semiconductors phát triển, truyền dữ liệu giữa các IC qua hai đường tín hiệu. Dữ liệu truyền từng bit theo tín hiệu đồng hồ và được dùng để giao tiếp với nhiều loại IC như vi điều khiển, cảm biến, EEPROM,...



Hình 4: *Hình minh họa cho giao thức I2C*

I2C sử dụng 2 đường truyền tín hiệu:

- SCL: Tạo xung nhịp đồng hồ do Master phát đi.
- SDA: Đường truyền nhận dữ liệu.

Giao tiếp I2C bao gồm quá trình truyền nhận dữ liệu giữa các Master và Slave.

- Thiết bị Master là một vi điều khiển, có nhiệm vụ điều khiển đường tín hiệu SCL và gửi/nhận dữ liệu hoặc lệnh thông qua đường SDA đến các thiết bị khác.
- Các thiết bị nhận tín hiệu điều khiển từ thiết bị Master được gọi là các thiết bị Slave.

3.2 Cooperative Scheduler

3.2.1 Khái niệm

Cooperative Scheduler:

- Là một interrupt service routine (ISR) được chia sẻ giữa nhiều task khác nhau.
- Cung cấp kiến trúc hệ thống single-tasking.
- Cho phép các task chạy vào những thời điểm cụ thể.
- Khi task được lên lịch chạy, nó được thêm vào danh sách chờ. Khi CPU rảnh, task chờ tiếp theo (nếu có) sẽ được thực thi. Task chạy đến khi hoàn thành, sau đó trả quyền điều khiển cho scheduler.

3.2.2 Thành phần

Các thành phần chính của Scheduler:

- **Cấu trúc dữ liệu**:
  - Là kiểu dữ liệu do người dùng định nghĩa, chứa thông tin về mỗi task.
  - Một task bao gồm các trường:
    * pTask: Con trỏ đến task (hàm).
    * Delay: Độ trễ (tính bằng ticks) trước khi task được chạy (lần tiếp theo).
    * Period: Khoảng thời gian giữa các lần chạy liên tiếp (tính bằng ticks).
    * RunMe: Cờ được tăng lên khi task đến thời điểm thực thi.
    * TaskID (optional): ID của task
  - Mảng task (SCH_tasks_G) dùng để lưu trữ các task được lên lịch: Kích thước mảng task phải đủ lớn để chứa tất cả các task trong ứng dụng.
- **Hàm khởi tạo (SCH_Init)**: Khởi tạo mảng task.
- **Hàm cập nhật (SCH_Update)**: Là một ISR được gọi khi timer tràn (overflow). Hàm này không trực tiếp thực thi task, mà kiểm tra xem task nào đến thời điểm chạy và tăng cờ RunMe của task đó. Ý nghĩa của hàm: Tính toán thời gian trễ còn lại của mỗi task.
- **Hàm thêm task (SCH_Add_Task)** dùng để thêm task vào mảng task.
- **Hàm thực thi (SCH_Dispatch_Tasks)**: Thực thi các task đã đến thời điểm chạy (có cờ RunMe > 0), chạy trong vòng lặp chính.
- **Hàm xóa task (SCH_Delete_Task)**: Xóa task khỏi mảng task.
- Một số hàm khác: Hàm báo cáo lỗi (SCH_Report_Status), Hàm đi vào chế độ sleep (SCH_Go_To_Sleep),...

3.2.3 Nguyên tắc thiết kế

Nguyên tắc thiết kế Scheduler:

- Không thể sử dụng các interrupt khác: chỉ interrupt timer được phép sử dụng.
- Tick interval: Nên chọn bằng ước số chung lớn nhất của tất cả các task interval.
- Độ dài task: Tất cả các task nên có độ dài nhỏ hơn tick interval.
- Timeout: Các task phải có cơ chế timeout để không block scheduler.
- Tổng thời gian task: Tổng thời gian thực thi của tất cả các task phải nhỏ hơn thời gian CPU có sẵn.
- Tránh task overlap: Cố gắng lên lịch các task sao cho không trùng nhau. Có thể tránh bằng cách điều chỉnh độ trễ ban đầu của task.

3.3 Xử lý input: Nút nhấn

3.3.1 Khái niệm

Button (nút nhấn) thường được sử dụng trong các hệ thống nhúng như một phần của giao diện người dùng. Chúng thường được kết nối với một MCU để tạo ra một mức logic nhất định khi được nhấn hoặc đóng (tức là "hoạt động") và mức logic ngược lại khi không được nhấn hoặc mở (tức là "không hoạt động"). Mức logic hoạt động có thể là '0' hoặc '1', nhưng mức '0' phổ biến hơn.

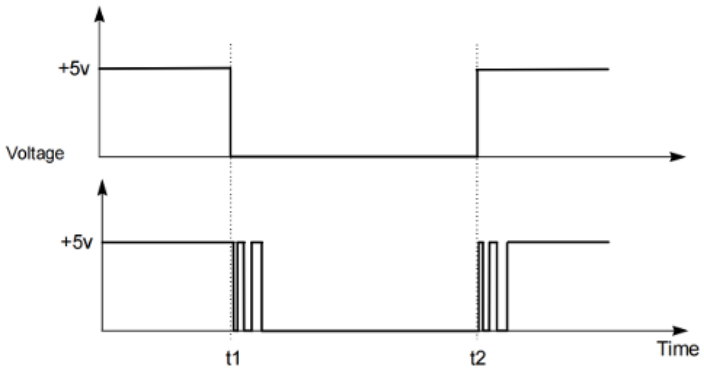
3.3.2 Nguyên tắc thiết kế

a. Điện trở kéo lên (pull-up)

Khi công tắc mở, chân cổng được kéo lên điện áp cung cấp của MCU (thường là 3.3V cho STM32F103) thông qua một điện trở bên trong. Khi công tắc đóng, điện áp chân sẽ là 0V.

Hình 5: *Nút nhấn sử dụng điện trở kéo lên*

b. Xử lý bouncing



Hình 6: *Tình trạng bounce của nút nhấn*

Các tiếp điểm cơ khí của công tắc có hiện tượng bounce - nảy (bật và tắt liên tục trong một khoảng thời gian ngắn) sau khi công tắc được đóng hoặc mở. Các hệ thống cần phải xử lý hiện tượng này để đảm bảo rằng vi điều khiển chỉ đọc một lần nhấn nút cơ khí (mechanical) thành một hành động (action) nhấn nút thành công chứ không phải nhiều lần.

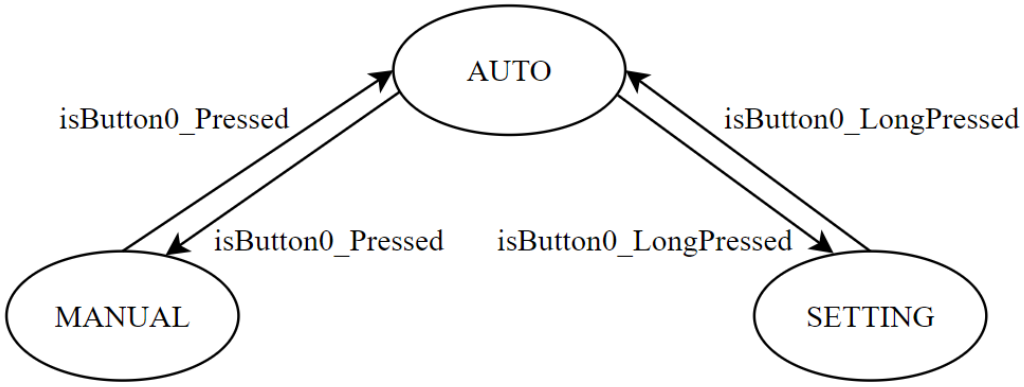
Sử dụng phần mềm để khử bounce: thường dùng vì tiết kiệm chi phí so với phương pháp điều chỉnh phần cứng để khử.

Nguyên tắc cơ bản khử nảy bằng phần mềm:

- Thiết lập tiêu chí tối thiểu cho một lần nhấn nút hợp lệ dựa trên sự khác biệt về thời gian.
- Bỏ qua các bounce trong vòng 10ms.
- Cung cấp phản hồi trong vòng 50ms sau khi phát hiện một lần nhấn nút.
- Có thể phát hiện một lần nhấn và nhả nút trong vòng 50ms.
- Quét nút: Kiểm tra trạng thái của nút sau mỗi N mili giây, với N $>$ 10ms và N $<=$ 50ms.
- Lọc nhiễu: So sánh trạng thái nút hiện tại với trạng thái nút cuối cùng và chỉ coi là hợp lệ nếu cả hai đều giống nhau (N=2). Có thể mở rộng kỹ thuật lọc bằng cách sử dụng N lớn hơn để tăng khả năng lọc.
- Điều chỉnh giá trị thời gian tùy yêu cầu.

4 Hiện thực hệ thống

4.1 Tổng quan hệ thống



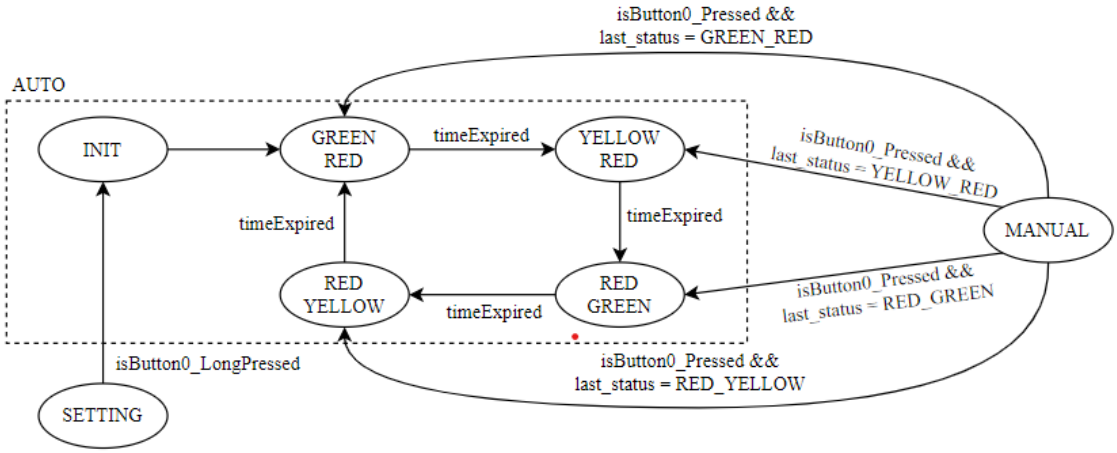
Hình 7: *FSM tổng quan hệ thống*

Mô tả tổng quan về các chế độ của hệ thống:

- **Chế độ tự động - Mode AUTO**: Mô phỏng trạng thái giao thông được điều khiển tự động theo đèn giao thông tại ngã tư trong thực tế. Chuyển trạng thái (chế độ): Tình trạng giao thông không thể luôn luôn ổn định mà có thể bị ảnh hưởng bởi nhiều yếu tố khác nhau (thời gian, thời tiết,...); khi tình trạng giao thông không còn ở mức độ ổn định (ùn tắc,...), tức là hệ thống đèn điều khiển tự động không còn hoạt động hiệu quả, ta cần thay đổi chế độ điều khiển giao thông.
- **Chế độ điều khiển bằng tay - Mode MANUAL**: Mô phỏng trạng thái giao thông tại ngã tư được điều khiển thủ công theo hiệu lệnh của cảnh sát giao thông (CSGT) trong thực tế. Tình huống: Khi xảy ra tình trạng kẹt xe, lưu thông khó khăn nghiêm trọng (giờ cao điểm); giao thông sẽ được điều khiển bởi CSGT.
- **Chế độ điều chỉnh thời gian của đèn - Mode SETTING**: Hiện thực chế độ điều chỉnh thời gian (duration) cho từng màu của đèn 3 màu. Tình huống: dựa vào tình trạng giao thông thực tế (giao thông tắc nghẽn,...) mà thời gian chờ (đèn ĐỎ) có thể cần dài hơn, thời gian lưu thông (đèn XANH) ngắn hơn để điều tiết giao thông ổn định.
- **Chuyển chế độ**: Chế độ AUTO chuyển sang chế độ MANUAL (và ngược lại) bằng việc nhấn thả nút nhấn 0, chế độ AUTO chuyển sang chế độ SETTING (và ngược lại) bằng việc nhấn đè nút nhấn 0.

4.2 Các chế độ cụ thể

4.2.1 Mode 1: AUTO

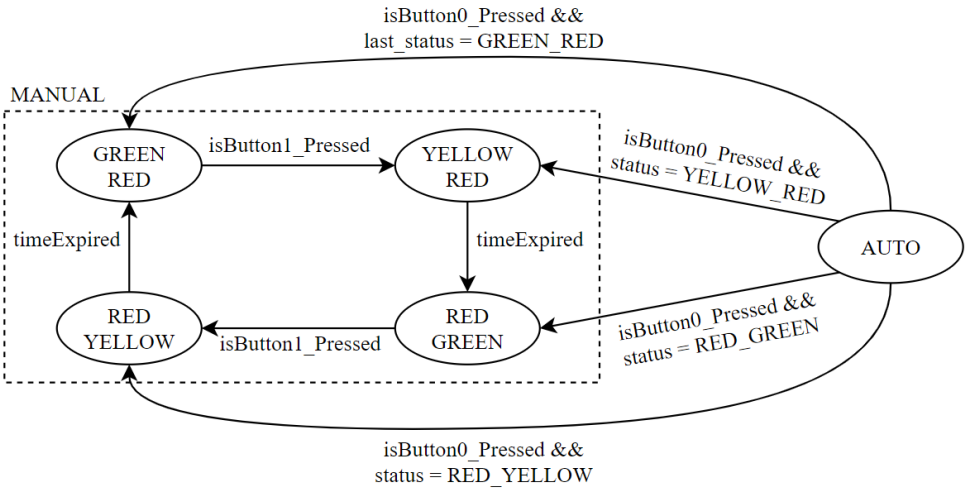


Hình 8: *FSM mode AUTO*

Quy tắc điều khiển giao thông tự động:

- Thời gian hiển thị: thời gian còn lại của đèn đang sáng.
- Phương tiện lưu thông theo quy tắc của 3 màu đèn trong thời gian (duration) của từng đèn. Sau khi hết thời gian thì chuyển trạng thái (đèn sáng-tắt), hoạt động tuần tự, lặp lại.
- Quy tắc về thời gian: để vận hành được hệ thống giao thông tại ngã tư thì ta tuân theo quy tắc thời gian đèn ĐỎ = thời gian đèn XANH + thời gian đèn VÀNG.

4.2.2 Mode 2: MANUAL

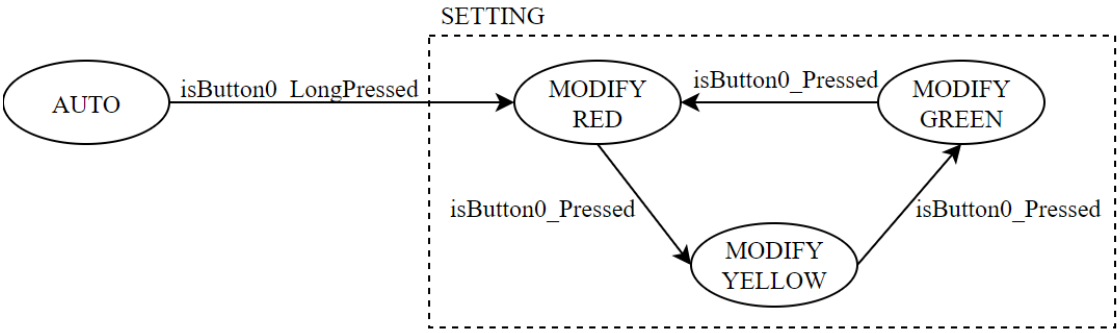


Hình 9: *FSM mode MANUAL*

Quy tắc điều khiển giao thông thủ công:

- Tín hiệu điều khiển của CSGT: đổi hướng lưu thông, mô phỏng bằng nút nhấn 1.
- Các hướng lưu thông đang di chuyển (đèn XANH) và dừng lại (đèn ĐỎ) thì giữ nguyên trạng thái cho đến khi có tín hiệu thay đổi theo sự điều khiển của CSGT.
- Các hướng lưu thông đang đi chậm để chuẩn bị dừng lại (đèn VÀNG) sau khi hết thời gian của đèn VÀNG thì dừng lại (đèn ĐỎ), hướng lưu thông đối diện đang chờ (đèn ĐỎ) có thể di chuyển (đèn XANH). Tiếp tục giữ nguyên trạng thái cho đến khi có tín hiệu thay đổi theo sự điều khiển của CSGT.
- Thời gian hiển thị: Không điều khiển tự động bằng đèn đếm lùi, mô phỏng bằng việc hiển thị giá trị 0 đối với trạng thái giữ nguyên và giá trị thời gian còn lại đối với trạng thái động đang chuẩn bị giữ nguyên (từ đèn VÀNG sang đèn ĐỎ).

4.2.3 Mode 1: SETTING



Hình 10: *FSM mode SETTING*

Quy tắc điều chỉnh thời gian (duration) cho từng màu đèn:

- Thay đổi đèn (chuyển sang các mode điều chỉnh giá trị của đèn khác): bằng việc nhấn thả nút nhấn 0.
- Điều chỉnh giá trị thời gian hoạt động của đèn (quy ước tăng): bằng việc nhấn thả nút nhấn 1, mỗi lần nhấn là tăng 1 đơn vị.
- Lưu giá trị vừa thay đổi (và chuyển về mode AUTO): bằng việc nhấn đè nút nhấn 0.
- Nếu như thay đổi giá trị (nhấn nút 1) nhưng không lưu giá trị (nhấn đè nút 0) mà đã chuyển sang điều chỉnh đèn khác (nhấn nút 0) thì giá trị thời gian hoạt động của đèn đó vẫn như cũ, không bị thay đổi.

Quy ước cho giá trị thời gian:

- Giá trị nằm trong khoảng [1, 99].
- Bất kể thay đổi giá trị ở đèn nào thì cũng phải duy trì quy tắc: thời gian đèn ĐỎ = thời gian đèn XANH + thời gian đèn VÀNG để có thể vận hành hệ thống đèn giao thông.

4.3 Chức năng các nút nhấn

Button No.	Mode AUTO	Mode MANUAL	Mode SETTING
0	Chuyển mode MANUAL	Chuyển mode AUTO	Chuyển đèn
1	X	Đổi hướng lưu thông	Tăng giá trị thời gian
Long_0	Chuyển mode SETTING	X	Lưu giá trị & chuyển mode AUTO
Long_1	X	X	X

Bảng 2: *Chức năng các nút nhấn*

5 Kết quả

Phần mềm chạy code: STM32CubeProgrammer (kết nối bằng ST-Link)
Kết quả hiện thực: **Source code** và **video demo**.

6 Kết luận

Sau khi đã hoàn thành xong project, nhóm em có những kết luận như sau:

- Sản phẩm đã hoạt động bình thường và đáp ứng đủ các yêu cầu theo đề tài.
- Nhờ đề tài mà các thành viên trong nhóm đã có thể tiếp cận được kiến thức về ứng dụng vi xử lý - vi điều khiển.
- Các thành viên trong nhóm vẫn hoàn thành đúng hạn các nhiệm vụ được giao và hợp tác tốt trong quá trình hoàn thành đồ án.
- Sản phẩm tuy đã hoàn thành các yêu cầu đề ra nhưng sẽ chưa tối ưu, còn những thiếu sót, hạn chế nhưng nhờ đó, các thành viên đã có thể có cho mình nhiều kinh nghiệm đối với làm việc với STM32 và IDE của nó.
- Nhóm xin chân thành cảm ơn sự hướng dẫn và hỗ trợ của các thầy đã giúp đỡ nhóm trong quá trình hoàn thành đồ án.

7 Tài liệu tham khảo

1. **Hướng dẫn nạp chương trình vào board**
2. **Lý thuyết scheduler**
3. **Xử lý nút nhấn**