

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA KỸ THUẬT ĐIỆN TỬ I

---o0o---



BÀI TẬP LỚN
HỆ THỐNG NHÚNG

ĐỀ TÀI:

THIẾT KẾ HỆ THỐNG MÁY NGHE NHẠC
SỬ DỤNG HỆ ĐIỀU HÀNH FREERTOS
TRÊN VI ĐIỀU KHIỂN STM32

Giảng viên hướng dẫn: ThS. Lương Công Dẫn

Lớp học phần: D22DTMT01

Nhóm sinh viên thực hiện: Lý Trọng Nghĩa – B22DCDT222

Nguyễn Đức Nam – B22DCDT206

Nguyễn Minh Đức – B22DCDT094

Dương Sơn Quang – B22DCDT238

Hà Nội – 11/2025

Mục lục

1	Mô tả tính năng, đặc tính hệ thống	2
1.1	Mô tả tính năng của hệ thống	2
1.1.1	Phát Nhạc Đa Chức Năng	2
1.1.2	Điều Khiển Linh Hoạt	2
1.1.3	Giao diện hệ thống trực quan	2
1.1.4	Tiết Kiệm Năng Lượng	2
1.2	Mô tả đặc tính kỹ thuật của hệ thống	3
1.2.1	Vì điều khiển	3
1.2.2	Cơ chế, kiến trúc phần mềm	3
1.2.3	Giao tiếp phần cứng	3
1.2.4	Mức tiêu thụ năng lượng ước tính	3
2	Chi tiết cấu hình các tác vụ trong FreeRTOS	4
2.1	Task 1: Quét Phím (vKeypadScanTask)	4
2.2	Task 2: Xử Lý Logic Chính (vPlayerLogicTask)	5
2.3	Task 3: Hiển Thị Màn Hình (vLcdDisplayTask)	5
2.4	Mối quan hệ giữa 3 Task	5
3	Đánh giá kết quả thực hiện	6
A	PHỤ LỤC - BẢNG PHÂN CÔNG CÔNG VIỆC	7

1 Mô tả tính năng, đặc tính hệ thống

1.1 Mô tả tính năng của hệ thống

Nhóm đã xây dựng hệ thống máy phát nhạc đơn giản dựa trên hệ vi xử lý STM32 với đầy đủ những chức năng cơ bản như sau:

1.1.1 Phát Nhạc Đa Chức Năng

- Hỗ trợ phát các định dạng nhạc phổ biến (MP3, WAV, WMA) thông qua module DFPlayer Mini.
- Có khả năng đọc thẻ nhớ microSD (hỗ trợ lên đến 32GB, định dạng FAT16/FAT32).
- Tích hợp loa công suất nhỏ (3W) trực tiếp hoặc xuất ra loa ngoài/tai nghe.

1.1.2 Điều Khiển Linh Hoạt

- Hỗ trợ chọn bài thông qua Keypad 4x4, chỉ cần nhập mã bài và nhấn phím “D” để chọn bài.
- Hỗ trợ các chức năng như chuyển bài kế tiếp (Next – Phím A), bài trước đó (Prev – Phím B), Pause/Resume (Phím C).
- Hỗ trợ tăng giảm âm lượng 30 mức thông qua phím “*” và “#”.

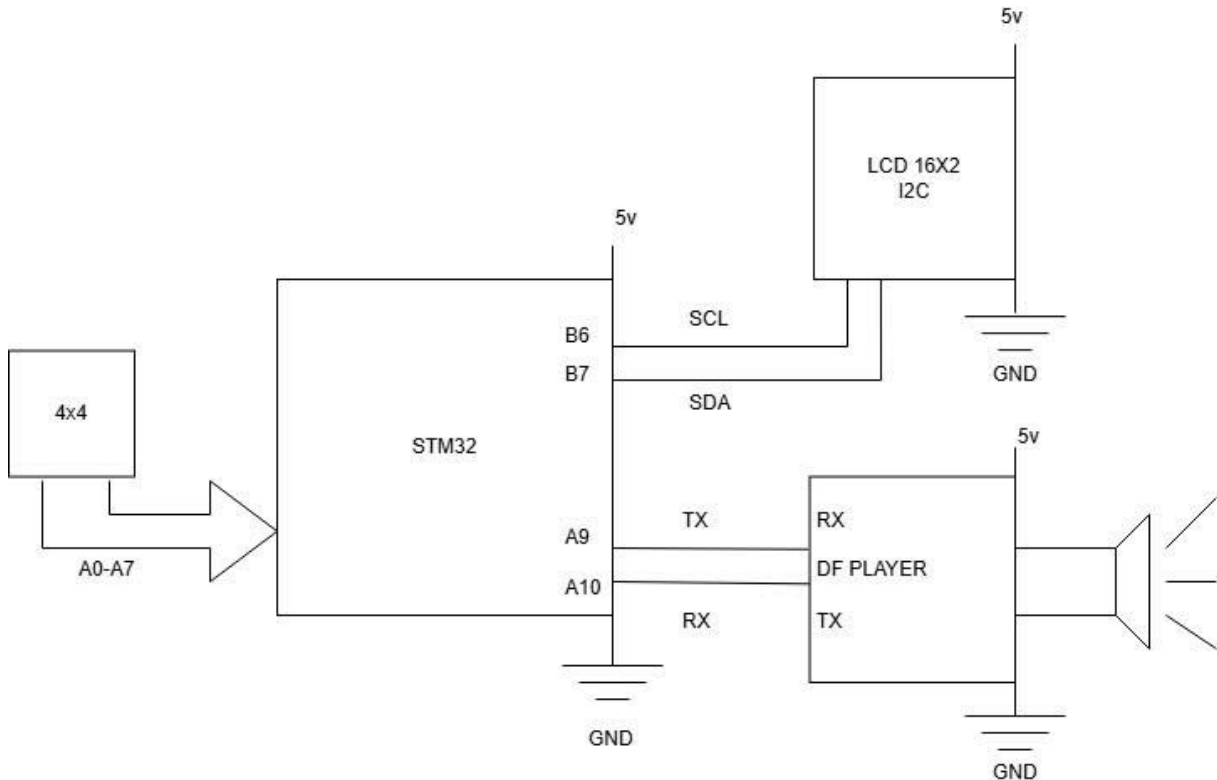
1.1.3 Giao diện hệ thống trực quan

- Hiển thị thông tin chi tiết trên màn hình LCD 16x2 (I2C).
- Thông tin hiển thị bao gồm: Số thứ tự bài hát (Track), Trạng thái (PLAY/PAUSE/STOP), Mức âm lượng, và các thông báo thao tác.
- Hiệu ứng giao diện: Có độ trễ hiển thị thông báo (0.5s) giúp người dùng dễ dàng quan sát thao tác.

1.1.4 Tiết Kiệm Năng Lượng

- Tự động tắt màn hình: Sau 30 giây không có tương tác, đèn nền LCD sẽ tự động tắt để tiết kiệm pin.
- Đánh thức tức thì: Hệ thống tự động bật lại đèn nền khi người dùng nhấn phím.
- Sleep Mode: STM32 tự động đi vào chế độ ngủ (Sleep Mode - __WFI).

1.2 Mô tả đặc tính kỹ thuật của hệ thống



Hình 1: Sơ đồ kết nối vật lý của hệ thống

1.2.1 Vi điều khiển

Sử dụng STM32F103C8T6 (Blue Pill), lõi ARM Cortex-M3, xung nhịp 72MHz.

1.2.2 Cơ chế, kiến trúc phần mềm

Phần mềm được phát triển dựa trên hệ điều hành thời gian thực FreeRTOS với cơ chế lập lịch ưu tiên. Phần mềm được chia thành 3 task rõ ràng với 3 mức độ ưu tiên khác nhau.

Bên cạnh đó, phần mềm được phát triển dựa trên kiến trúc hướng sự kiện với việc sử dụng ngắt ngoài (EXTI) cho keypad, Software Timer cho việc quản lý thời gian tắt backlight.

1.2.3 Giao tiếp phần cứng

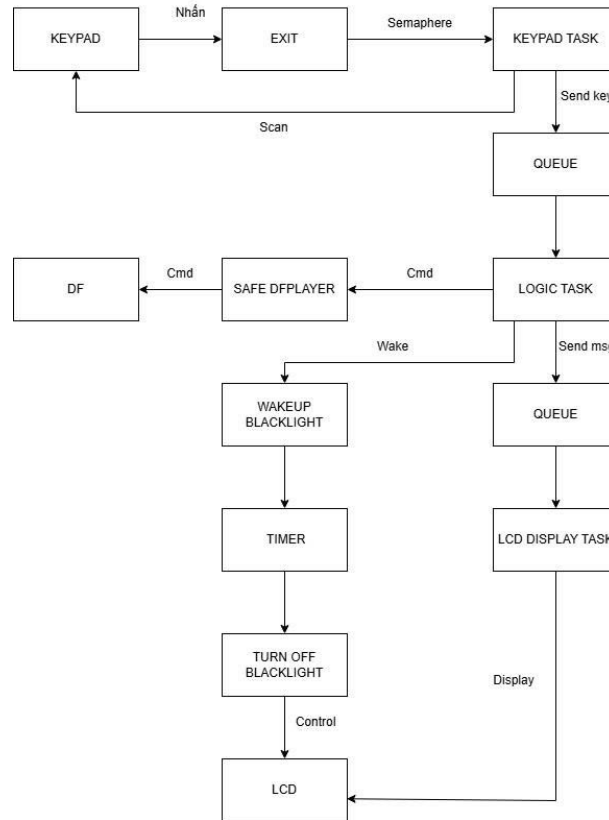
- UART (USART1): Giao tiếp với module DFPlayer Mini (9600 baud).
- I2C (I2C1): Giao tiếp với màn hình LCD 16x2.
- GPIO & EXTI: Quét bàn phím ma trận 4x4 sử dụng ngắt ngoài.

1.2.4 Mức tiêu thụ năng lượng ước tính

- Ở chế độ phát nhạc, backlight bật: $\sim 110\text{mA} - 170\text{mA}$.

- Ở chế độ phát nhạc, backlight tắt, hệ thống sleep: $\sim 100\text{mA} - 120\text{mA}$.
- Ở chế độ chờ (IDLE), backlight tắt, hệ thống sleep: $\sim 48\text{mA}$.

2 Chi tiết cấu hình các tác vụ trong FreeRTOS



Hình 2: Sơ đồ khối mô tả các task giao tiếp với nhau

2.1 Task 1: Quét Phím (vKeypadScanTask)

1. **Vai trò:** Đây là “đôi tai” của hệ thống, chịu trách nhiệm lắng nghe mọi thao tác của user từ bàn phím ma trận 4x4.
2. **Mức độ ưu tiên (Priority):** Cao nhất (3). Do phản hồi của user là quan trọng nhất (Real-time).
3. **Cơ chế hoạt động (Hybrid Interrupt + Polling):**
 - *Trạng thái chờ (Blocked):* Bình thường, task này ngủ hoàn toàn trên một Semaphore.
 - *Đánh thức (Wakeup):* Khi user nhấn bất kỳ phím nào, ngắt phần cứng xảy ra và “Give” Semaphore.
 - *Xử lý (Active):* Task tỉnh dậy, tắt ngắt, quét ma trận để xác định phím.
 - *Gửi dữ liệu:* Gửi mã phím vào g_keypadQueue.
 - *Ngủ lại:* Cấu hình lại ngắt và quay về trạng thái ngủ.

2.2 Task 2: Xử Lý Logic Chính (vPlayerLogicTask)

1. **Vai trò:** Đây là “bộ não” trung tâm, điều phối mọi hoạt động của hệ thống.
2. **Mức độ ưu tiên (Priority):** Trung bình (2).
3. **Cơ chế hoạt động (Event-Driven):** Task nằm trong vòng lặp vô hạn, bị Blocked tại hàm xQueueReceive chờ dữ liệu.
4. **Nhiệm vụ cụ thể:**
 - *Phân tích phím bấm:* Nhận mã phím và quyết định lệnh.
 - *Quản lý trạng thái (State Machine):* Cập nhật trạng thái Play/Pause/Stop.
 - *Điều khiển DFPlayer:* Gửi lệnh UART xuống module MP3 (bảo vệ bởi Mutex).
 - *Cập nhật màn hình:* Gửi gói tin LcdMessage_t vào Queue cho Task LCD.
 - *Quản lý năng lượng:* Reset bộ đếm thời gian tắt đèn nền.

2.3 Task 3: Hiển Thị Màn Hình (vLcdDisplayTask)

1. **Vai trò:** Đây là “gương mặt” của hệ thống, chịu trách nhiệm hiển thị thông tin lên màn hình LCD 16x2.
2. **Mức độ ưu tiên (Priority):** Thấp nhất (1). Do việc hiển thị có thể chậm trễ vài mili-giây mà không ảnh hưởng nghiêm trọng đến chức năng phát nhạc. Ta nhường CPU cho việc quét phím và xử lý logic quan trọng hơn.
3. **Cơ chế hoạt động:**
 - Task ngủ hoàn toàn (Blocked) chờ dữ liệu từ g_lcdQueue.
 - Khi nhận được bản tin LcdMessage_t, nó sẽ thức dậy.

Tối ưu hóa:

- So sánh nội dung mới với cũ, nếu giống thì bỏ qua.
- Sử dụng g_i2cMutex để chiếm quyền điều khiển Bus I2C.

2.4 Mối quan hệ giữa 3 Task

1. Người dùng bấm phím → Ngắt EXTI → Đánh thức Task 1.
2. Task 1 quét ra mã phím → Gửi vào Queue → Đánh thức Task 2.
3. Task 2 xử lý logic → Gửi lệnh UART + Gửi bản tin Queue → Đánh thức Task 3.
4. Task 3 nhận bản tin → Hiển thị lên màn hình LCD.

3 Đánh giá kết quả thực hiện

Nhóm đã xây dựng được một hệ thống phát nhạc hoạt động mượt mà, ổn định nhờ việc sử dụng cơ chế ngắt ngoài cùng với việc ứng dụng hệ điều hành thời gian thực FreeRTOS vào hệ thống. Hệ thống đáp ứng gần như ngay lập tức, hoạt động ổn định với từng bối cảnh sử dụng khác nhau của người dùng.

Nhờ cơ chế tối ưu hóa năng lượng thông qua cơ chế sleep (__WFI), tắt ngoại vi không cần thiết và tắt backlight, nhóm đã tối ưu hóa thành công một phần việc tiêu thụ năng lượng của hệ thống. Tuy nhiên, do giới hạn của phần cứng, việc tối ưu tiêu thụ năng lượng này là giới hạn cuối cùng mà nhóm có thể thực hiện.

A PHỤ LỤC - BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	Công việc đảm nhận
1	Lý Trọng Nghĩa	<ul style="list-style-type: none">- Thiết kế kiến trúc tổng thể hệ thống- Thiết kế Task vPlayerLogicTask- Liên kết các thành phần với nhau
2	Dương Sơn Quang	<ul style="list-style-type: none">- Thiết kế thư viện dfplayer- Thiết kế một phần vPlayerLogicTask (phần gọi lệnh DFPlayer)
3	Nguyễn Minh Đức	<ul style="list-style-type: none">- Thiết kế Task vKeypadScanTask- Thiết kế thư viện keypad_4x4
4	Nguyễn Đức Nam	<ul style="list-style-type: none">- Thiết kế task vLcdDisplayTask- Thiết kế thư viện lcd_i2c

Bảng 1: Bảng phân công công việc nhóm