

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

\*

# BÁO CÁO

## BÀI TẬP LỚN HỆ NHÚNG

### Tên đề tài: Game “Tetris”

#### Nhóm 1

**Sinh viên thực hiện:** Nguyễn Trung Sơn 20225225  
Nguyễn Quang Huy 20225336  
Ngân Văn Thiện 20215647

**Giảng viên hướng dẫn:** TS. Ngô Lam Trung

Hà Nội, tháng 6 năm 2025

## MỤC LỤC

<b>1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>3</b>
1.1. Mô tả về sản phẩm và mục tiêu của dự án.....	3
1.2. Yêu cầu đối với dự án.....	3
1.2.1. Yêu cầu chức năng.....	3
1.2.2. Yêu cầu phi chức năng.....	4
<b>2. THIẾT KẾ.....</b>	<b>4</b>
2.1. Phân chia chức năng.....	4
2.1.1. Phần cứng.....	4
2.1.2. Phần mềm.....	5
2.2. Thiết kế phần cứng.....	5
2.2.1. Linh kiện và mô-đun.....	5
2.2.2. Sơ đồ ghép nối.....	5
2.2.3. Tần số truyền dữ liệu.....	6
2.3. Thiết kế phần mềm.....	6
2.3.1. Module phần mềm.....	6
2.3.2. Sơ đồ khối hệ thống.....	7
2.3.3. Biểu đồ luồng.....	8
<b>3. XÂY DỰNG HỆ THỐNG.....</b>	<b>9</b>
3.1. Mô tả các chức năng phần mềm.....	9
3.2. Đánh giá và hướng phát triển.....	10
3.2.1. Đánh giá.....	10
3.2.2. Hướng phát triển.....	10

## PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	Mã sinh viên	Phân chia công việc
1	Nguyễn Trung Sơn	20225225	Xử lý logic game, quản lý khối rơi, di chuyển khối, xoay khối, phát hiện va chạm, tính điểm
2	Nguyễn Quang Huy	20225336	Thiết kế khối Tetris, bảng game, giao diện game, cấu hình chân GPIO, lưu điểm cao bằng bộ nhớ Flash
3	Ngân Văn Thiện	20215647	Xử lý logic tạo khối ngẫu nhiên sử dụng RNG, cấu hình và phát âm thanh sử dụng buzzer

# 1. GIỚI THIỆU ĐỀ TÀI

## 1.1. Mô tả về sản phẩm và mục tiêu của dự án

Dự án game Tetris nhằm triển khai trò chơi Tetris kinh điển trên vi điều khiển STM32F429ZIT6, sử dụng STM32CubeIDE để phát triển và TouchGFX để thiết kế giao diện người dùng đồ họa (GUI). Trò chơi chạy trên màn hình TFT LCD 2.4 inch với khả năng cảm ứng, được tích hợp trên bo STM32F429I-DISCOVERY. Người chơi điều khiển các khối tetromino rơi xuống bằng các nút bấm vật lý, với mục tiêu sắp xếp chúng thành các hàng ngang hoàn chỉnh để xóa và ghi điểm. Trò chơi bao gồm các tính năng như theo dõi điểm số, lưu điểm cao, hiệu ứng âm thanh qua loa buzzer thụ động.

Mục tiêu chính là tạo ra một trò chơi Tetris hấp dẫn, tương tác và đầy đủ chức năng, tận dụng các khả năng phần cứng của STM32F429ZIT6, bao gồm lõi ARM Cortex-M4, màn hình LCD và FreeRTOS để quản lý tác vụ thời gian thực. Dự án cũng nhằm thể hiện kỹ năng lập trình hệ thống nhúng, phát triển GUI, và tích hợp phần cứng - phần mềm.

## 1.2. Yêu cầu đối với dự án

### 1.2.1. Yêu cầu chức năng

**Cơ chế chơi game:**

- Triển khai lối chơi Tetris kinh điển với các khối rơi từ trên xuống lưới 10x20.
- Hỗ trợ bảy hình dạng tetromino tiêu chuẩn với sinh ngẫu nhiên theo hệ thống "bag". (Tetromino là một hình dạng hình học được tạo thành từ bốn ô vuông, được kết nối theo chiều ngang hoặc dọc. Có bảy tetromino khác nhau, bao gồm I, O, T, S, Z, L và J)
- Cho phép người chơi di chuyển khối sang trái (PG2), sang phải (PG3), xoay theo chiều kim đồng hồ (PC4), và làm rơi nhanh (PC5).
- Cung cấp chức năng bắt đầu/tạm dừng qua nút vật lý (PA0) hoặc nút cảm ứng trên màn hình.
- Khởi động lại trò chơi khi trạng thái game over được kích hoạt.

**Xử lý đầu vào:**

- Đọc đầu vào từ bốn nút bấm (PG2, PG3, PC4, PC5) để điều khiển trò chơi.
- Sử dụng nút có sẵn trên bo (PA0) để bắt đầu/tạm dừng.
- Hỗ trợ đầu vào cảm ứng trên màn hình LCD cho chức năng bắt đầu/tạm dừng.

**Hiển thị:**

- Hiển thị lưới trò chơi (10x20) trên màn hình LCD 240x320 pixel.
- Hiển thị khối tetromino hiện tại, khối tiếp theo, điểm số, điểm cao, và trạng thái trò chơi (tạm dừng/game over).
- Sử dụng màu sắc riêng biệt cho từng loại tetromino.
- Triển khai hiệu ứng hình ảnh cho việc xóa hàng và game over.

**Âm thanh:**

- Tạo hiệu ứng âm thanh cho di chuyển khối, xoay, rơi, xóa hàng, game over, lên cấp, và rơi nhanh.

- Phát nhạc nền (chủ đề Tetris) bằng loa buzzer thụ động kết nối với PA9 qua PWM (TIM1\_CH2).
- Cho phép bật/tắt nhạc nền.

#### **Sinh số ngẫu nhiên:**

- Sử dụng bộ tạo số ngẫu nhiên phần cứng (RNG) để xáo trộn các tetromino.

#### **Lưu trữ dữ liệu:**

- Lưu và truy xuất điểm cao trong bộ nhớ flash của vi điều khiển.

#### **Hoạt động thời gian thực:**

- Sử dụng FreeRTOS để quản lý các tác vụ cho logic trò chơi, xử lý đầu vào, cập nhật GUI, và tạo âm thanh.
- Đảm bảo lối chơi mượt mà với điều khiển phản hồi nhanh và cập nhật khung hình nhất quán.

### **1.2.2. Yêu cầu phi chức năng**

- **Độ tin cậy:**
  - Hệ thống phải hoạt động không bị treo hoặc lỗi trong thời gian chơi dài.
  - Phát hiện đầu vào phải được khử rung để tránh kích hoạt nhiều lần từ một lần nhấn.
- **Hiệu suất:**
  - Duy trì tốc độ trò chơi ổn định với tốc độ rơi có thể điều chỉnh cho chế độ bình thường và rơi nhanh.
  - Cập nhật GUI phải đạt ít nhất 30 FPS để đảm bảo hình ảnh mượt mà.
- **Độ bền phần cứng:**
  - Các nút bấm và buzzer phải được kết nối chắc chắn trên bo mạch thử nghiệm.
  - Kết nối phải chịu được sử dụng lặp lại mà không bị lỏng.
- **Khả năng sử dụng:**
  - Giao diện phải trực quan, với phản hồi hình ảnh và âm thanh rõ ràng cho các hành động của người dùng.
  - Hiệu ứng âm thanh phải nghe rõ nhưng không quá to (dải 1.5–2.5 kHz).
- **Tính di động:**
  - Hệ thống phải gọn nhẹ, nằm gọn trong bo STM32F429I-DISCOVERY và bo mạch thử nghiệm nhỏ.
- **Khả năng bảo trì:**
  - Mã nguồn phải được phân mô-đun, có tài liệu rõ ràng, và tuân theo quy ước của STM32CubeIDE và TouchGFX.

## **2. THIẾT KẾ**

### **2.1. Phân chia chức năng**

#### **2.1.1. Phần cứng**

**Đầu vào:** Bốn nút bấm (PG2, PG3, PC4, PC5) trên bo mạch thử nghiệm và một nút có sẵn trên bo (PA0) xử lý điều khiển trò chơi.

**Đầu ra:**

- Màn hình TFT LCD 2.4 inch (ILI9341) hiển thị lưới trò chơi, tetromino, và các yếu tố giao diện.
- Loa buzzer thụ động (KY-006) kết nối với PA9 tạo hiệu ứng âm thanh và nhạc.

**Xử lý:** Lõi Cortex-M4 của STM32F429ZIT6 xử lý logic trò chơi, đầu vào, và hiển thị GUI.

**Bộ nhớ:** Bộ nhớ flash (sector 11) lưu điểm cao.

**Sinh số ngẫu nhiên:** Ngoại vi RNG tạo số ngẫu nhiên để chọn tetromino.

### 2.1.2. Phần mềm

**Logic trò chơi:** Quản lý di chuyển tetromino, phát hiện va chạm, xóa hàng, và tính điểm.

**GUI:** TouchGFX hiển thị lưới trò chơi, tetromino, điểm số, và thông báo trạng thái.

**Xử lý đầu vào:** Đọc trạng thái nút bấm và đầu vào cảm ứng, khử rung tín hiệu, và gửi tín hiệu điều khiển qua hàng đợi FreeRTOS.

**Âm thanh:** Tạo tín hiệu PWM cho buzzer để phát âm thanh và nhạc.

**Quản lý tác vụ:** FreeRTOS lập lịch các tác vụ cho logic trò chơi, đầu vào, GUI, và âm thanh.

**Quản lý điểm cao:** Đọc/ghi điểm cao vào bộ nhớ flash.

## 2.2. Thiết kế phần cứng

### 2.2.1. Linh kiện và mô-đun

- **Bo STM32F429I-DISCOVERY:**
  - **Vi điều khiển:** STM32F429ZIT6 (ARM Cortex-M4, 180 MHz, 2 MB Flash, 256 KB SRAM).
  - **Màn hình:** TFT LCD 2.4 inch (240x320 pixel, bộ điều khiển ILI9341) với bảng cảm ứng.
  - **Nút có sẵn:** PA0 (nút người dùng).
  - **SDRAM:** IS42S16400J (64 Mbit, 16-bit data width), có sẵn trên bo, kết nối qua FMC.
  - **Ngoại vi:** LTDC, SPI5, I2C3, FMC, DMA2D, RNG, TIM1, GPIO.
- **Bo mạch thử nghiệm:**
  - **Nút bấm:** Bốn công tắc xúc giác (kết nối với PG2, PG3, PC4, PC5).
  - **Loa buzzer thụ động (KY-006):**
    - Tần số dao động: 1.5–2.5 kHz.
    - Điện áp hoạt động: 3–5V.
    - Chân tín hiệu: PA9 (đầu ra PWM từ TIM1\_CH2).
    - Kích thước: 1.85 x 1.5 cm.
    - Khoảng cách chân cắm: 2.54 mm.

### 2.2.2. Sơ đồ ghép nối

#### Nút bấm:

- PG2 (Trái): Kết nối với đầu vào GPIO có điện trở kéo lên.
- PG3 (Phải): Kết nối với đầu vào GPIO có điện trở kéo lên.
- PC4 (Xoay): Kết nối với đầu vào GPIO có điện trở kéo lên.
- PC5 (Rơi nhanh): Kết nối với đầu vào GPIO có điện trở kéo lên.
- PA0 (Bắt đầu/Tạm dừng): Nút có sẵn trên bo, đầu vào GPIO không có điện trở kéo lên.
- Tất cả nút bấm chia sẻ một đất chung với bo STM32F429I-DISCOVERY.

#### Loa buzzer:

- Tín hiệu: PA9 (đầu ra PWM TIM1\_CH2).
- VCC: 3.3V từ STM32F429I-DISCOVERY.
- GND: Đất chung.

#### Màn hình LCD:

- Kết nối qua SPI5 (chế độ master, dữ liệu 8-bit, bộ chia tần số 16).
- Chip Select: PC2 (đầu ra GPIO).
- Data/Command: PD13 (đầu ra GPIO).
- Bảng cảm ứng: I2C3 (100 kHz, định địa chỉ 7-bit).

#### SDRAM:

- Có sẵn trên bo STM32F429I-DISCOVERY, kết nối qua FMC Bank 2 (8-bit cột, 12-bit hàng, độ rộng dữ liệu 16-bit, độ trễ CAS 3).

### 2.2.3. Tần số truyền dữ liệu

- SPI5 (LCD): ~5.625 MHz (180 MHz / bộ chia tần số 16).
- I2C3 (Bảng cảm ứng): 100 kHz.
- FMC (SDRAM): Hoạt động ở HCLK/2 (90 MHz).
- TIM1 (PWM Buzzer): Xung nhịp 1 MHz (84 MHz / bộ chia 84), tạo âm thanh 200–2000 Hz.
- RNG: Tạo số ngẫu nhiên 32-bit ở tốc độ xung ngoại vi (90 MHz).

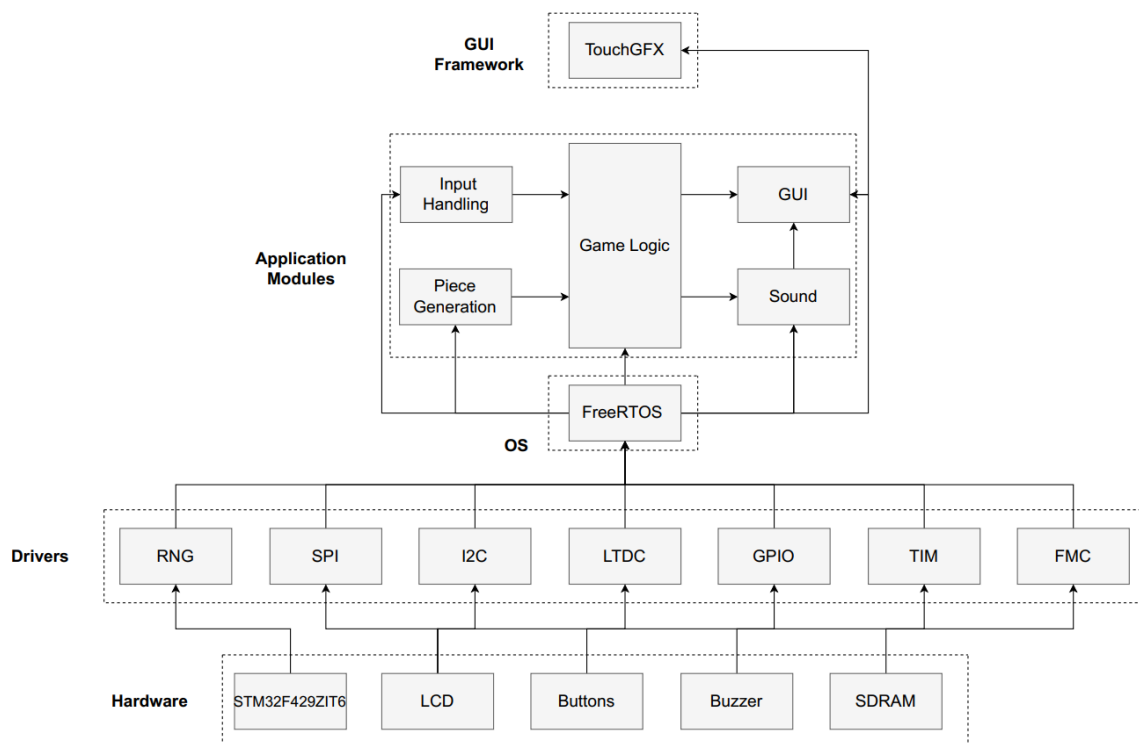
## 2.3. Thiết kế phần mềm

### 2.3.1. Module phần mềm

- **Logic trò chơi (Screen1View.cpp):**
  - Quản lý di chuyển tetromino, xoay, phát hiện va chạm, xóa hàng, và tính điểm.
  - Sử dụng mảng lưới 20x10 và cấu trúc tetromino.
  - Xử lý các trạng thái trò chơi (đang chơi, tạm dừng, game over).
- **GUI (Screen1ViewBase.cpp, Screen1View.cpp):**
  - Hiển thị lưới trò chơi, tetromino, khối tiếp theo, điểm số, điểm cao, và văn bản trạng thái.
  - Triển khai hiệu ứng hình ảnh (xóa hàng, game over, làm nổi bật điểm số).
  - Sử dụng các container và box của TouchGFX để hiển thị tetromino.

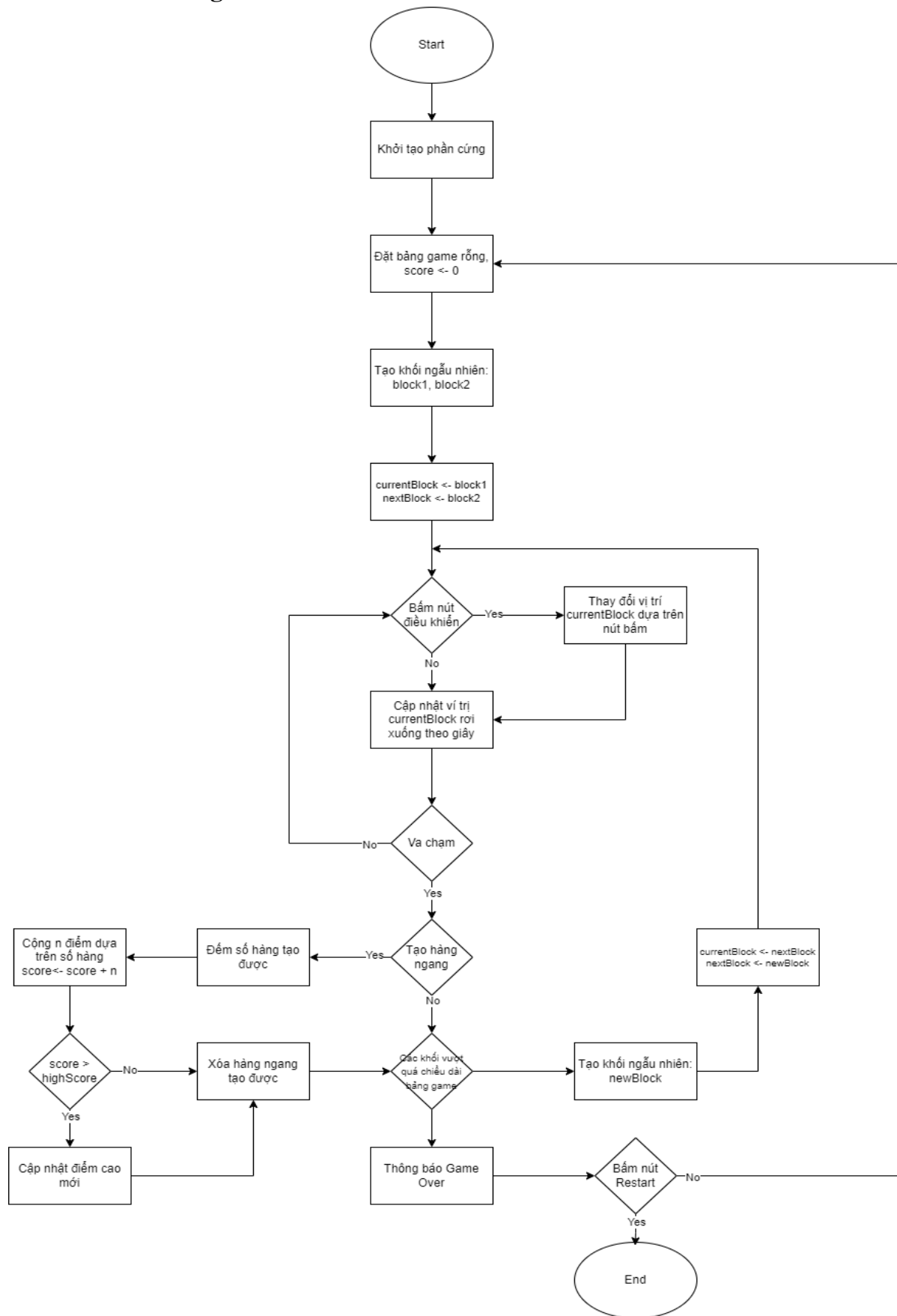
- **Xử lý đầu vào (main.c: StartInputTask):**
  - Đọc trạng thái nút bấm (PG2, PG3, PC4, PC5, PA0) với khử rung.
  - Gửi tín hiệu điều khiển (1: trái, 2: phải, 3: xoay, 4: bắt đầu rơi nhanh, 5: kết thúc rơi nhanh, 6: bắt đầu/tạm dừng) đến controlQueue.
  - Xử lý đầu vào cảm ứng cho nút bắt đầu/tạm dừng.
- **Âm thanh (main.c: StartSoundTask, Screen1View.cpp):**
  - Tạo tín hiệu PWM trên TIM1\_CH2 cho hiệu ứng âm thanh và nhạc.
  - Hỗ trợ tần số từ 200–2000 Hz, ánh xạ từ 0–255.
  - Phát chủ đề Tetris làm nhạc nền với chức năng bật/tắt.
- **Tạo khối (main.c: StartGameTask):**
  - Sử dụng RNG để xáo trộn một "bag" gồm bảy tetromino.
  - Gửi chỉ số khối đến pieceQueue.
- **Quản lý tác vụ (main.c):**
  - Các tác vụ FreeRTOS: defaultTask, GUI\_Task (TouchGFX), gameTask, inputTask, soundTask.
  - Hàng đợi: pieceQueue (chỉ số tetromino), controlQueue (tín hiệu điều khiển), soundQueue (dữ liệu âm thanh).
- **Quản lý điểm cao (Screen1View.cpp):**
  - Đọc/ghi điểm cao vào sector 11 của flash (0x080E0000).

### 2.3.2. Sơ đồ khối hệ thống





### 2.3.3. Biểu đồ luồng



### 3. XÂY DỰNG HỆ THỐNG

#### 3.1. Mô tả các chức năng phần mềm

Logic trò chơi (Screen1View.cpp):

- **Chức năng:** Quản lý di chuyển tetromino (**handleTickEvent**), phát hiện va chạm (**checkCollision**), xóa hàng (**checkAndClearLines**), và tính điểm (**updateScoreDisplay**).
- **Triển khai:** Sử dụng mảng lưới 20x10 và cấu trúc tetromino với tọa độ ô và màu sắc. Hỗ trợ xoay với xử lý va chạm tường và hệ thống bag cho sinh ngẫu nhiên.
- **Tính năng chính:**
  - Hiệu ứng hình ảnh: Xóa hàng (nhấp nháy trắng), game over (mờ đỏ), làm nổi bật điểm số (nhấp nháy vàng).
  - Tính điểm: 100, 300, 600, 1000 điểm cho 1–4 hàng xóa.
- **GUI (Screen1ViewBase.cpp, Screen1View.cpp):**
  - **Chức năng:** Hiển thị lưới trò chơi (60,40,120,240 pixel), tetromino hiện tại, khối tiếp theo (180,80,48,48 pixel), điểm số, và văn bản trạng thái.
  - **Triển khai:** Sử dụng container TouchGFX (**TetrominoContainer**, **NextTetrominoContainer**) và box cho các ô tetromino. Mỗi loại tetromino có container riêng với màu sắc độc đáo.
  - **Tính năng chính:**
    - Nút cảm ứng bắt đầu/tạm dừng.
    - Văn bản động cho điểm số và thông báo game over/điểm cao.
- **Xử lý đầu vào (main.c: StartInputTask):**
  - **Chức năng:** Thăm dò nút bấm mỗi 5 ms, khử rung bằng cách theo dõi trạng thái, và gửi tín hiệu điều khiển đến **controlQueue**.
  - **Triển khai:** Đọc chân GPIO (**PG2, PG3, PC4, PC5, PA0**) với điện trở kéo lên (trừ PA0). Tín hiệu: 1 (trái), 2 (phải), 3 (xoay), 4 (bắt đầu rơi nhanh), 5 (kết thúc rơi nhanh), 6 (bắt đầu/tạm dừng).
  - **Tính năng chính:** Khử rung đảm bảo kích hoạt đơn lẻ mỗi lần nhấn.
- **Âm thanh (main.c: StartSoundTask, Screen1View.cpp):**
  - **Chức năng:** Tạo hiệu ứng âm thanh và nhạc qua **PWM** trên **TIM1\_CH2**.
  - **Triển khai:** Ánh xạ giá trị tần số 0–255 thành 200–2000 Hz. Dữ liệu âm thanh (tần số << 8 | thời lượng) được gửi đến soundQueue. Phát chủ đề Tetris với 32 nốt.
  - **Tính năng chính:**
    - Hiệu ứng: Di chuyển (C4, 50 ms), xoay (E4, 80 ms), rơi (G4, 60 ms), xóa hàng (chuỗi C5–G5), game over (chuỗi C6–C5).
    - Bật/tắt nhạc nền với stopSound.
- **Tạo khối (main.c: StartGameTask):**
  - **Chức năng:** Xáo trộn bag bảy tetromino bằng RNG và gửi chỉ số đến pieceQueue mỗi 10 ms.
  - **Triển khai:** Sử dụng **HAL\_RNG\_GenerateRandomNumber** để xáo trộn. Xáo lại khi bag rỗng.
- **Quản lý điểm cao (Screen1View.cpp):**
  - **Chức năng:** Đọc/ghi điểm cao vào sector 11 của flash.
  - **Triển khai:** Sử dụng **HAL\_FLASH\_Program** và **HAL\_FLASHEx\_Erase**. Khởi tạo điểm cao là 0 nếu flash bị xóa (0xFFFFFFFF).

## 3.2. Đánh giá và hướng phát triển

### 3.2.1. Đánh giá

- **Chức năng:**
  - Tất cả cơ chế game (di chuyển, xoay, xóa hàng, tính điểm) được triển khai đầy đủ.
  - Xử lý đầu vào phản hồi nhanh với khur rung.
  - GUI hiển thị tất cả yếu tố cần thiết với hoạt hình mượt mà.
  - Hiệu ứng âm thanh và nhạc rõ ràng, có thể bật/tắt.
  - RNG đảm bảo phân phối tetromino công bằng.
  - Điểm cao được lưu trữ bền vững qua các chu kỳ nguồn.
- **Phi chức năng:**
  - **Độ tin cậy:** Hệ thống chạy ổn định không bị treo trong thời gian chơi dài.
  - **Hiệu suất:** Đạt ~30 FPS cập nhật GUI và tốc độ rơi nhất quán (30 tick bình thường, 3 tick rơi nhanh).
  - **Độ bền phần cứng:** Nút bấm và buzzer được hàn chắc chắn trên bo mạch thử nghiệm.
  - **Khả năng sử dụng:** Điều khiển trực quan và phản hồi hình ảnh/âm thanh rõ ràng.
  - **Tính di động:** Thiết kế gọn nhẹ trong bo discovery và bo thử nghiệm.
  - **Khả năng bảo trì:** Mã nguồn phân mô-đun với phân tách tác vụ rõ ràng và các tệp do TouchGFX tạo.

### 3.2.2. Hướng phát triển

- Sử dụng buzzer chủ động hoặc DAC để có âm thanh phong phú hơn.
- Triển khai điều khiển dựa trên cảm ứng cho di chuyển/xoay.
- Thêm chế độ tiết kiệm năng lượng (ví dụ, làm mờ LCD khi tạm dừng).
- Thiết kế PCB tùy chỉnh để tích hợp nút và buzzer tốt hơn.

## Đóng góp các thành viên

Link github nhóm: <https://github.com/son26704/TetrisGame>

