**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

A red and yellow sign

Description automatically generated with low confidence

**BÀI TẬP LỚN**

**MÔN HỆ NHÚNG**

**Đề tài: XÂY DỰNG TRÒ CHƠI SPACE INVADERS**

**TRÊN KIT PHÁT TRIỂN STM32F429I-DISC1**

**Giảng viên:** TS. Ngô Lam Trung

**Lớp:** 714992

**Nhóm sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Lê Vũ Quang | 20194358 |
| 2 | Hoàng Kim Minh | 20194326 |
| 3 | Trung Lê Tuấn Anh | 20194227 |

Hà Nội, tháng 8 năm 2022

**Nội dung**

[I. Mô tả đề tài 2](#_Toc111456375)

[1. Yêu cầu 2](#_Toc111456376)

[2. Phân công công việc 2](#_Toc111456377)

[II. Hệ thống phần cứng 4](#_Toc111456378)

[1. Kiến trúc hệ thống 4](#_Toc111456379)

[2. II. Kit STM32F429I-DISC1 4](#_Toc111456380)

[3. Mạch nút bấm 6](#_Toc111456381)

[III. Phần mềm trò chơi 7](#_Toc111456382)

[1. Kiến trúc phần mềm 7](#_Toc111456383)

[2. Hoạt động của phần mềm 7](#_Toc111456384)

[IV. Kết quả thực hiện 12](#_Toc111456385)

# Mô tả đề tài

## Yêu cầu

Trò chơi Space Invaders là trò chời dạng shoot-em-up, trong đó người chơi điều khiển nhân vật di chuyển để tránh hoả lực của kẻ thù và tiêu diệt các mục tiêu.

Trò chơi sẽ được xây dựng trên kit STM32F429I-DISC1 kết hợp với thư viện đồ hoạ TouchGFX. Bộ 32F429IDISCOVERY tận dụng khả năng của các vi điều khiển hiệu suất cao STM32F429, cho phép người dùng dễ dàng phát triển các ứng dụng phong phú với giao diện người dùng đồ họa tiên tiến.

## Phân công công việc

|  |  |
| --- | --- |
| **Nhiệm vụ** | **Người thực hiện** |
| Hàn phím bấm  Xử lý tín hiệu nút bấm | Hoàng Kim Minh |
| Tìm hiểu thu thập hình ảnh đồ họa | Lê Vũ Quang |
| Thiết kế đồ họa bằng TouchGFX | Lê Vũ Quang |
| Tìm hiểu và lên ý tưởng trò chơi | Trung Lê Tuấn Anh  Lê Vũ Quang |
| Phát triển phần mềm điều khiển | Lê Vũ Quang  Hoàng Kim Minh  Trung Lê Tuấn Anh |
| Xử lý logic và hiển thị trò chơi | Lê Vũ Quang |
| Làm Báo cáo | Lê Vũ Quang  Hoàng Kim Minh  Trung Lê Tuấn Anh |

# Hệ thống phần cứng

## Kiến trúc hệ thống

Hệ thống gồm 4 thành phần chính:

* Kit STM32F429I-DISC1
* Mini USB-cable
* Mạch nút bấm
* Dây nối

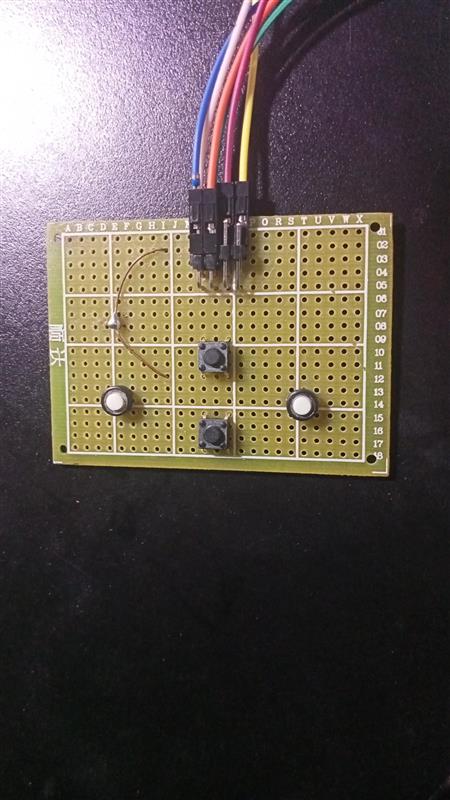
## Kit STM32F429I-DISC1

Dòng sản phẩm MCU hiệu suất cao STM32 F4 dựa trên ARM Cortex-M4 với các lệnh DSP và FPU là phần mở rộng của danh mục STM32 hàng đầu trong ngành hướng tới hiệu suất cao hơn nữa. Tần số hoạt động lên đến 180 MHz. ST Microelectronics STM32F429I-DISC1 là Discovery board dành cho các bộ vi điều khiển (MCU) STM32F429. Nó bao gồm mọi thứ cần thiết cho người mới bắt đầu và người dùng có kinh nghiệm để bắt đầu phát triển ứng dụng với giao diện người dùng đồ họa.

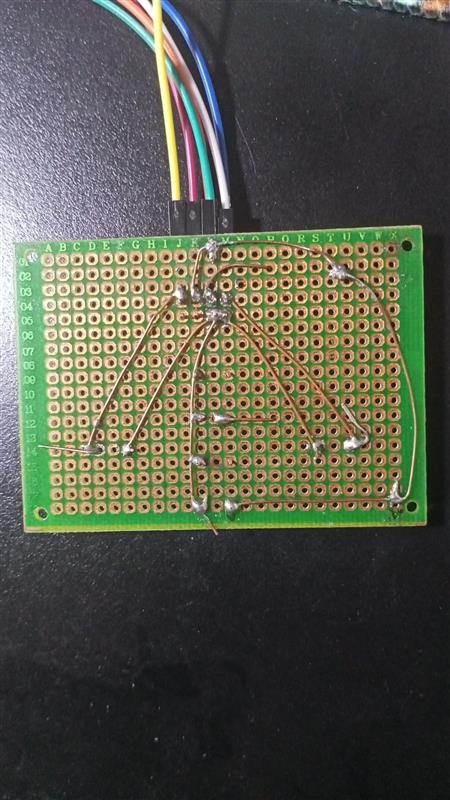


Hình III‑1 Kit phát triển STM32F429I-DISC1

## Mạch nút bấm



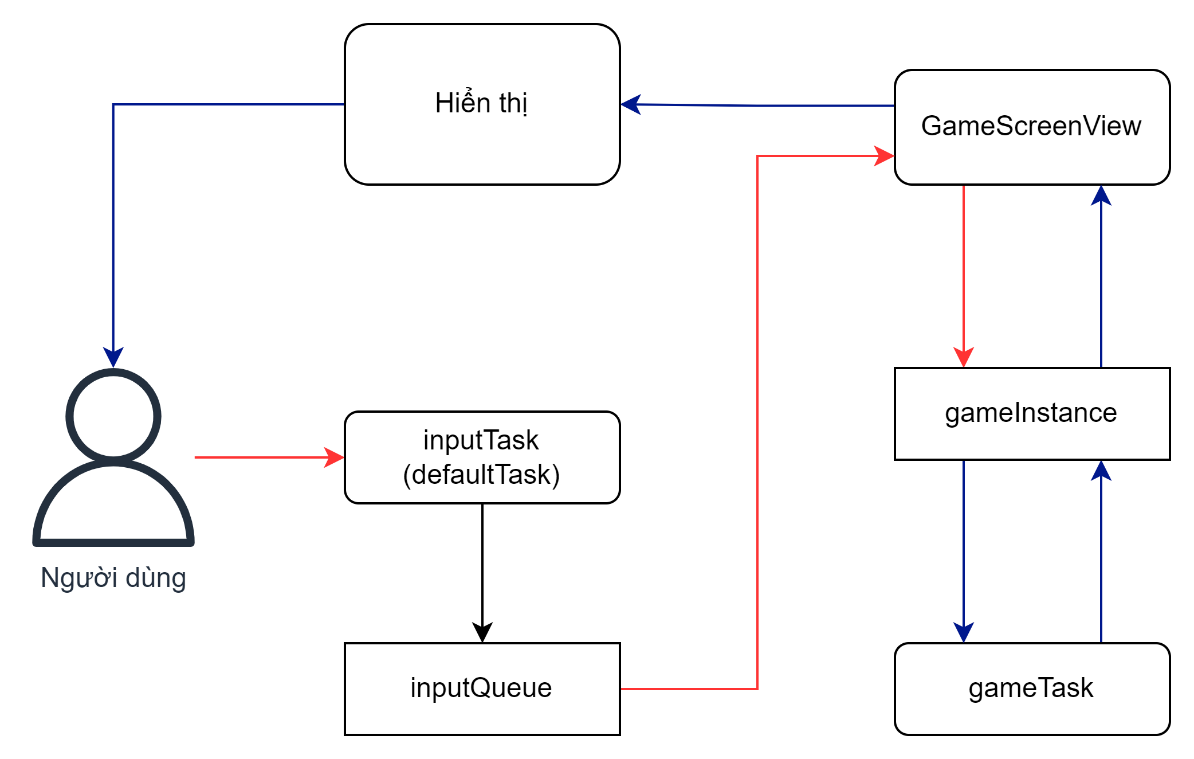
Hình III‑2 Mặt trước mạch nút bấm



Hình III‑3 Mặt sau mạch nút bấm

# Phần mềm trò chơi

## Kiến trúc phần mềm



Hình IV‑1 Kiến trúc phần mềm

Kiến trúc phần mềm gồm có 5 thành phần chính:

* GameScreenView: Tiếp nhận dữ liệu từ người dung và hiển thị hình ảnh lên màn hình
* inputTask: Nhận dữ liệu từ người dụng và gửi vào inputQueue để chờ xử lý
* gameTask: Cập nhật thông tin các đối tượng xử lý logic trò chơi
* inputQueue: Chứa dữ liệu được người dung gửi tới trò chơi thông qua các nút bấm
* gameInstance: Đối tượng trò chơi của chương trình, chứa thông tin về trò chơi

## Hoạt động của phần mềm

Phần mềm sẽ gồm 2 luồng hoạt động chính là *Nhận dữ liệu từ người dùng* và *Xử lý và hiển thị trò chơi.*

### Nhận dữ liệu từ người dung

Các nút bấm sẽ được cài đặt để gửi tín hiệu tới các cồng GPIO PE2, PE3, PE4, PE5. Chương trình sẽ duy trì 1 task (defaultTask) để liên tục đọc dữ liệu từ các cổng này và gửi lên hàng chờ.

Text

Description automatically generated

Hình IV‑2 Khởi tạo các chân vào ra để nhận dữ liệu



Hình IV‑3 Hàm nhận dữ liệu từ người dùng

Dữ liệu nằm trong hàng chờ sẽ được GameScreenView poll và lấy ra với mỗi lượt tick trong hàm.

Text

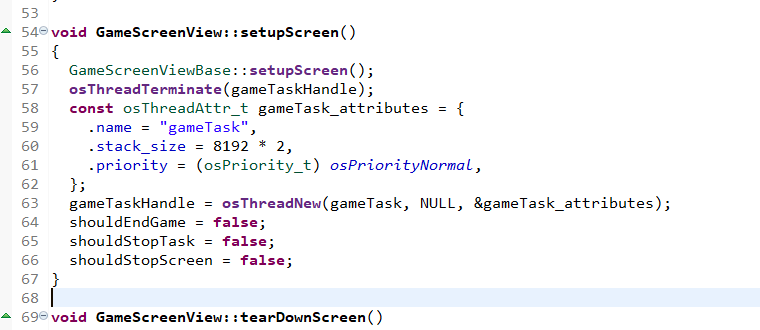
Description automatically generated

Hình IV‑4 Lấy dữ liệu từ trong inputQueue của GameScreenView

Sau đó GameScreenView sẽ xử lý dữ liệu và cập nhật vào đối tượng trong gameInstace (ở đây là hướng di chuyển của ship – đối tượng mà người chơi điều khiển).

### Xử lý và hiển thị trò chơi

Khi một GameScreenView được tạo, nó sẽ tạo them một task là gameTask.

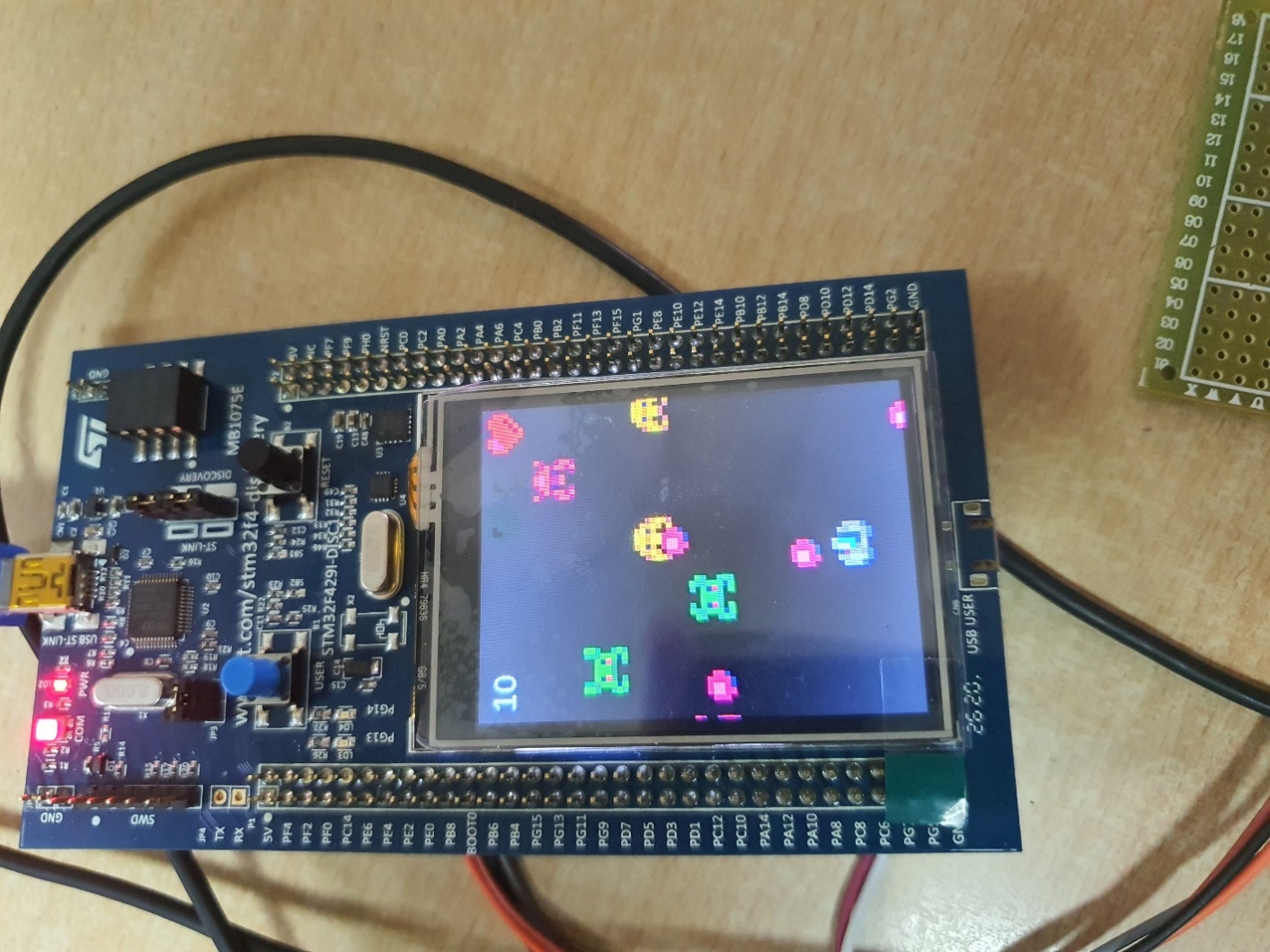


Hình IV‑ Hàm tạo gameTask

gameTask sẽ dựa trên các dữ liệu có được từ gameInstance để xử lý logic và cập nhật lại thông tin của gameInstace cho phù hợp. GameScreenView sẽ liên tục polling dữ liệu từ gameInstance để cập nhật lại hiển thị của các đối tượng trên màn hình và hiển thị cho người chơi

# Kết quả thực hiện

  
*Hình V-1:Giao diện menu trò chơi*

  
*Hình V-2:Giao diện trò chơi*

  
*Hình V-3: Giao diện khi kết thúc lượt chơi*

BỔ SUNG