

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KỸ THUẬT MÁY TÍNH**

**DƯƠNG ANH KHÔI - 22520696**  
**NGUYỄN ĐĂNG THANH TUỆ - 22521616**  
**TỔNG VIẾT TRƯỜNG - 22521583**

**BÁO CÁO ĐỒ ÁN**  
**MÔN HỌC: CE103.O22 VI XỬ LÝ-VI ĐIỀU KHIỂN**  
**GVHD: PHẠM MINH QUÂN**

**ĐỀ TÀI: FLAPPY BIRD SỬ DỤNG VI ĐIỀU KHIỂN**



**COMPUTER ENGINEERING**



**UIT**  
**TRƯỜNG ĐẠI HỌC**  
**CÔNG NGHỆ THÔNG TIN**

**TP. HỒ CHÍ MINH, 2024**

## Chương 1: Tổng quan đề tài đồ án

### 1.1. Lý do chọn đề tài:

Việc chọn đề tài đồ án làm game Flappy Bird sử dụng vi điều khiển không chỉ đòi hỏi sự sáng tạo trong thiết kế và lập trình mà còn cần sự hiểu biết sâu sắc về phần cứng. Đề tài này có tính ứng dụng cao từ giáo dục đến giải trí, và cung cấp một cơ hội tuyệt vời để phát triển kỹ năng kỹ thuật cũng như khả năng giải quyết vấn đề của sinh viên.

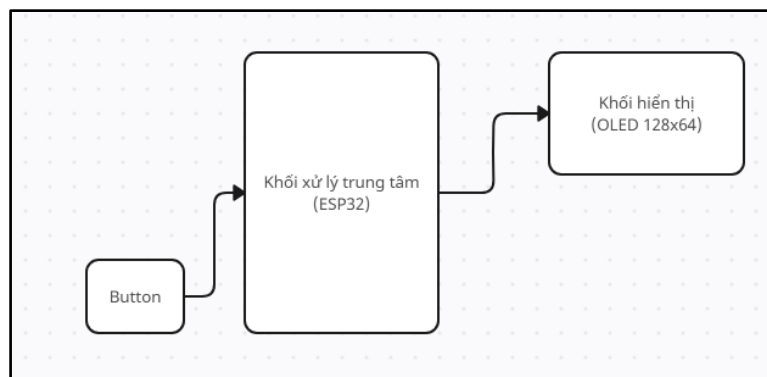
### 1.2. Mục tiêu:

Đây là cơ hội để bọn em khám phá khả năng của vi xử lý trong việc tạo ra trò chơi điện tử, từ việc lập trình giao diện người dùng đến xử lý đồ họa và logic trò chơi. Nghiên cứu này không chỉ giúp sinh viên phát triển kỹ năng kỹ thuật mà còn mở ra cánh cửa mới cho việc ứng dụng công nghệ vào giải trí và giáo dục.

Trong quá trình làm đồ án, không chỉ là việc tìm hiểu và phân tích để hoàn thành tốt sản phẩm đồ án, mà đây còn là cách để giúp nhóm có đạt được sự đoàn kết, gắn bó và cùng nhau hoàn thành tốt công việc. Để đạt được mục tiêu này, các thành viên trong nhóm cần phối hợp chặt chẽ và đóng góp ý kiến để hoàn thành đồ án một cách hiệu quả.

## Chương 2: Các phần tử trong mạch:

### 2.1. Sơ đồ khối của mạch:

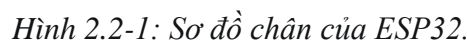


Hình 2.1: Sơ đồ khối.

- Khối xử lý trung tâm: Sử dụng vi điều khiển ESP32 để điều khiển hoạt động của cả mạch với các nhiệm vụ như: nhận tín hiệu từ button, xử lý tín hiệu, hiển thị lên khối hiển thị.
- Khối hiển thị: Sử dụng màn hình OLED 128x64 để hiển thị giao diện trò chơi lên màn hình.

ESP32 là một bộ vi điều khiển thuộc loại bộ vi điều khiển trên chip có công suất thấp và tiết kiệm chi phí. Hầu hết tất cả các biến thể của ESP32 đều có Bluetooth và Wi-Fi chế độ kép, khiến nó trở nên rất linh hoạt, mạnh mẽ và đáng tin cậy cho nhiều ứng dụng.

ESP32 cũng được thiết kế để tiêu thụ điện năng thấp, lý tưởng cho các ứng dụng chạy bằng pin. Nó có hệ thống quản lý năng lượng cho phép nó hoạt động ở chế độ ngủ và chỉ thức dậy khi cần, điều này có thể kéo dài đáng kể tuổi thọ pin.



### 2.3. Giới thiệu về màn hình hiển thị 128x64 SH1106:

#### *Thông số kỹ thuật:*

- Điện áp sử dụng: 3V3 đến 5V (DC).
- Công suất tiêu thụ: 0.04W.
- Độ phân giải: 128X64 pixel.
- Độ rộng màn hình: 0.96inch.
- Giao tiếp: I2C.
- Màu: Trắng và Đen.
- Driver: SSD1306.

#### *Sơ đồ nối chân:*

<b>VCC</b>	3V3 đến 5V
<b>GND</b>	GND
<b>SCL</b>	Xung clock - A5
<b>SDA</b>	Truyền dữ liệu - A4

Hình 2.3: Sơ đồ nối chân của màn hình.

### Chương 3: Thiết kế hệ thống:

#### 3.1. Ý tưởng:

Chương trình viết theo phong cách hướng đối tượng, chỉ có 2 loại đối tượng cần quan tâm:

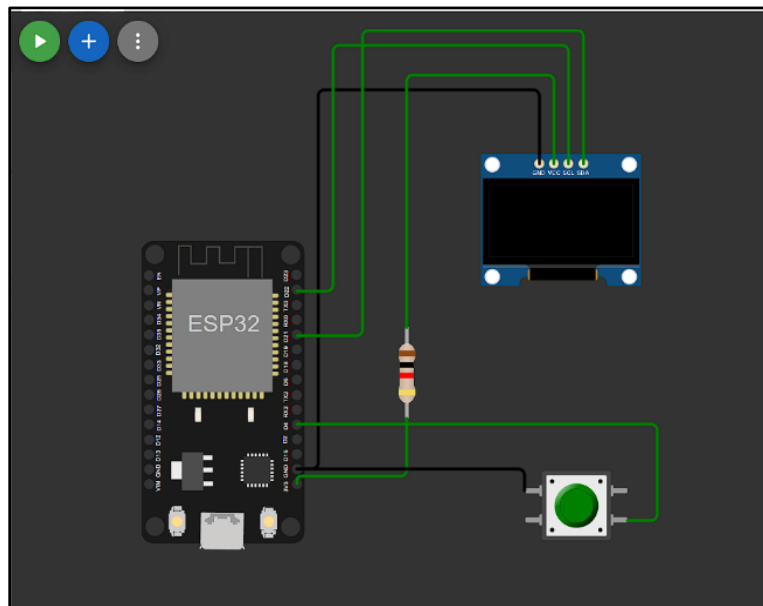
Đối tượng	Chức năng, nhiệm vụ	Thuộc tính hình học	Thuộc tính khác.
<b>Chú chim</b>	Người dùng nhấn chạm vào màn hình (với Project này là nhấn vào nút ấn), điều khiển chim bay lên, vượt qua các chướng ngại vật.	Tọa độ : X,Y	Sự tồn tại: Sống/Chết. Hướng: Bay lên/ bay xuống.
<b>Ống nước</b>	Là các chướng ngại vật, xuất hiện cả ở trên và dưới màn hình, chú chim bay và vượt qua các khe hở tạo bởi 2 ống nước.	Tọa độ: X,Y.	Sự tồn tại: Còn hoặc không còn ảnh hưởng tới chú chim. Hướng: di chuyển từ phải sang trái.

Hình dáng chi tiết của ống nước được phác họa bởi một hình chữ nhật đơn giản dạng ống.

Hệ quan sát của ta (Camera) gắn với hệ của chú chim, do đó , khi biểu diễn quá trình di chuyển của chim , đối tượng này luôn giữ một hoành độ cố định, trong khi đó những chiếc ống nước sẽ di chuyển từ phải qua trái, tạo ra chuyển động tương đối giữa chim và cột.  $\Rightarrow$  **Sẽ không phải là n chiếc ống di chuyển, mà sẽ là chú chim vượt qua chướng ngại vật n lần.**

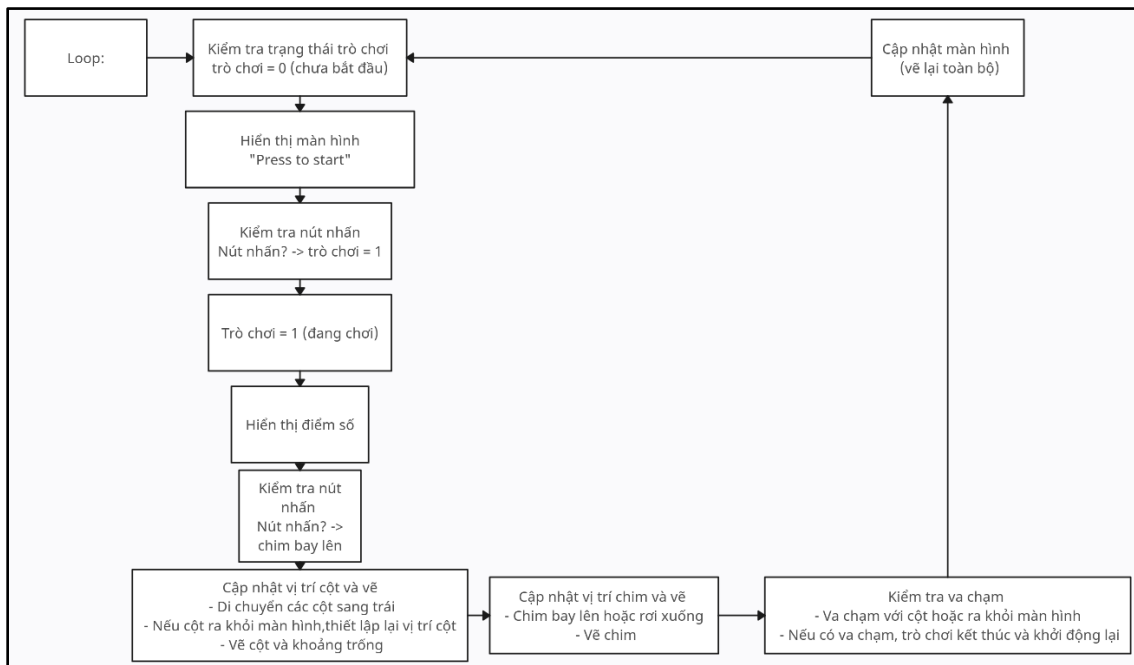
Khi lập trình, điều này có nghĩa là chỉ có một số ít ống nước cần quan tâm, cụ thể 6 ống nước di chuyển cách đều. Sau mỗi chu kì di chuyển (đi hết từ phải qua trái), chúng lại trở về vị trí xuất phát rồi lại di chuyển nhưng với độ cao thay đổi.

### 3.2. Sơ đồ kết nối các linh kiện:



Hình 3.2: Sơ đồ kết nối.

### 3.3. Lưu đồ thuật toán:



Hình 3.3: Lưu đồ thuật toán.

### 3.4. Hiện thực phần cứng:

#### 3.4.1. Thực hiện trên breadboard:

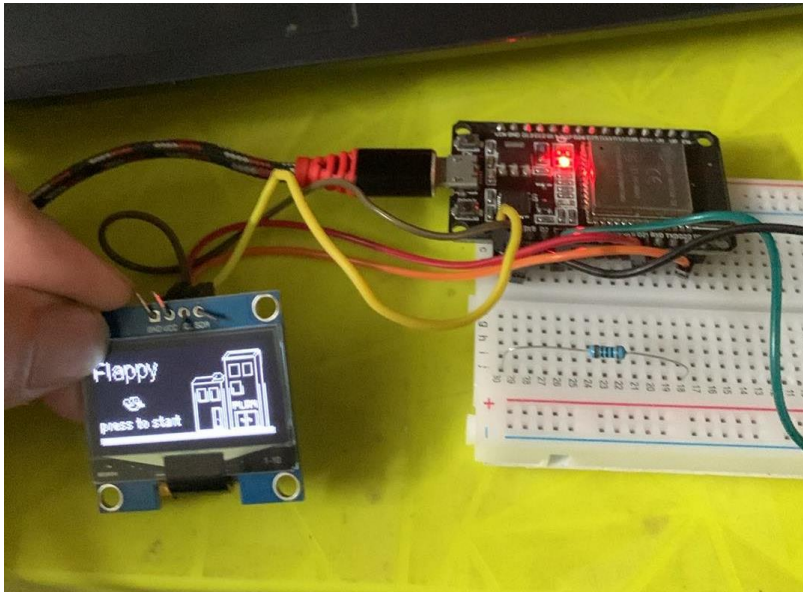
##### \*Vật liệu cần chuẩn bị:

- 1 đế breadboard
- 1 chip ESP32
- 1 con trở 10kΩ
- 1 nút bấm
- 6 cọc dây nối điện
- Nguồn từ laptop

**Cách đi dây:**

Trên màn hình có 4 cổng gồm VCC GND SCL và SDA, chúng ta nối các cổng đó theo thứ tự sau:

- VCC nối vào 3v3 (thông qua biến trở)
- GND nối vào GND
- SCL nối vào D22
- SDA nối vào D21
- 1 chân của button nối vào cổng GND của chip
- 1 chân còn lại nối vào cổng D4 của chip

**Hình minh họa:**

*Hình 3.4-1: Nạp code và lắp mạch trên breadboard.*

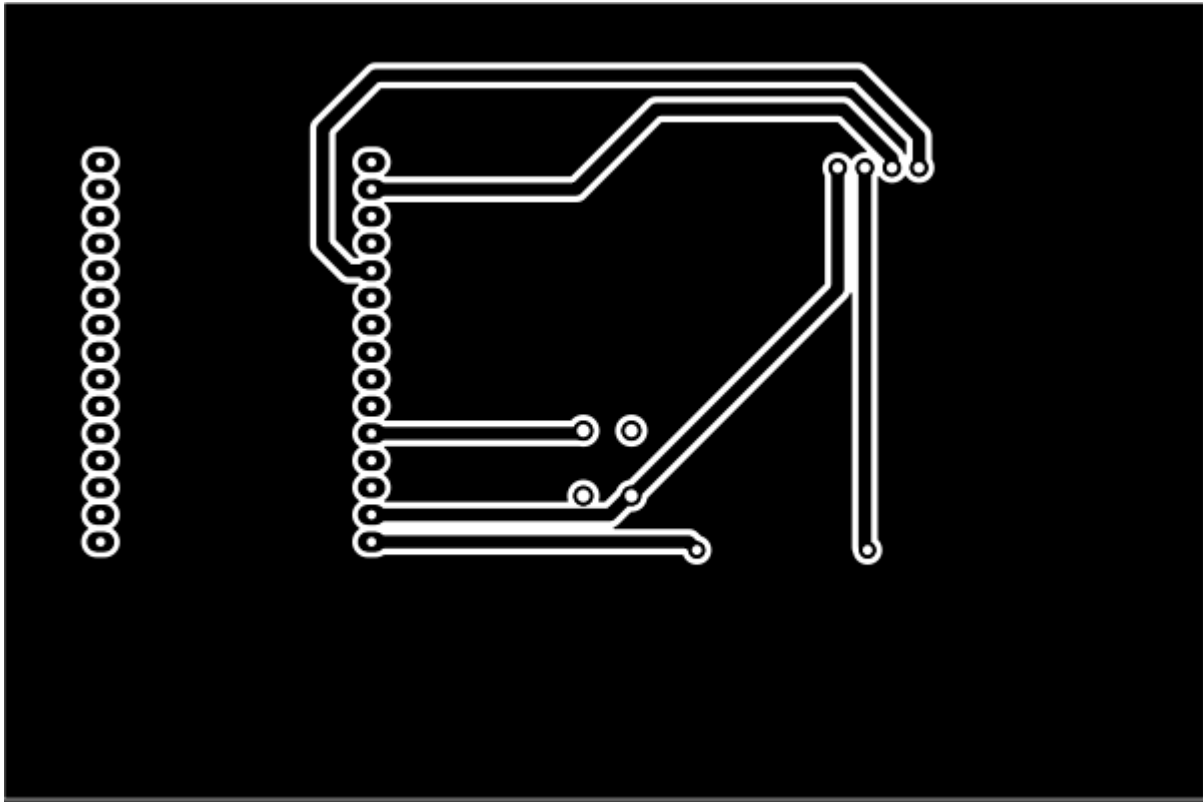
**3.4.2. Tiến hành làm PCB:****Vật liệu cần chuẩn bị:**

- 1 tấm đồng 1 lớp PCB
- Mạch PCB đã in (In decal hoặc giấy làm mạch)



- Bột sắt ( $\text{FeCl}_3$ )
- Miếng chùi nhôm
- 1 rào đực
- 1 rào cái

### Bản thiết kế PCB:



Hình 3.4-2: Bản thiết kế PCB.

### Các bước làm:

1. Cắt tấm đồng vừa đủ để in mạch.
2. Chà tấm đồng để lộ ra phần đồng chưa bị tác động của không khí.

3. Áp phần mạch PCB đã in vừa với miếng đồng vừa chà, sau đó ủ ở nhiệt độ vừa từ 20 - 30p (ủ cho đến khi nào chắc chắn mạch đã in đủ lên miếng đồng).
4. Sau khi đã ủ xong, ngâm liền với nước và bóc dần phần giấy ra.
5. Sau khi thấy mạch đã ổn, ngâm mạch với dung dịch bột sắt  $\text{FeCl}_3$  để mất các phần đồng cần thiết để lộ ra đường mạch.
6. Khoan các lỗ để cắm các linh kiện vào mạch.
7. Sử dụng rào được ghim vào mạch để gắn con chip và màn hình vào mạch.
8. Thực hiện hàn màn hình và các chân rào để dính vào mạch.
9. Thử chạy và hoàn tất!



## **Chương 4: Kết quả đạt được và đánh giá**

### **4.1 Kết quả đạt được:**

#### **4.1.1 Hiểu về game Flappy Bird**

- Nắm vững cách trò chơi Flappy Bird hoạt động, bao gồm cách chim di chuyển và cách tính điểm, qua đó có thể lập trình sao cho trò chơi hoạt động tốt nhất.
- Đã tìm hiểu về cách game loop và xử lý sự kiện được triển khai trong trò chơi.

#### **4.1.2 Lập trình ESP32:**

- Hiểu về cách lập trình cho mạch ESP32, bao gồm việc sử dụng ngôn ngữ C/C++ và các thư viện hỗ trợ.
- Tìm tòi, học hỏi cách hoạt động của mạch, thông qua đó đã tạo được ứng dụng chạy trên ESP32 để điều khiển game Flappy Bird.

#### **4.1.3 Giao tiếp với màn hình và cảm biến:**

- Kết nối thành công giữa ba thiết bị là màn hình, chip ESP32 và nút nhấn. Hiểu rõ việc đặt nút nhấn với cổng nào ở con chip để ứng dụng có thể hoạt động tốt (chẳng hạn cổng D4 của ESP32 cho phép khi chú chim thua thì sẽ quay về màn hình chính, khi ấn một lần nữa sẽ tiếp tục trò chơi).
- Hiểu rõ từng linh kiện để sử dụng chúng hiển thị trạng thái game và điều khiển chim trong trò chơi.
- Sử dụng một biến trở cắm vào cổng VCC của màn hình để đảm bảo điện áp ổn định tránh làm hỏng linh kiện.

#### **4.1.4 Tối ưu hóa và thử nghiệm:**

- Tối ưu hóa mã nguồn để đảm bảo trò chơi chạy mượt mà và không gặp lỗi.
- Thử nghiệm trò chơi trên ESP32 trên mô phỏng (Wokwi), breadboard và trên mạch để đảm bảo tính ổn định.

### **4.2 Đánh giá**

#### **4.2.1 Ưu điểm**

- Thành công trong triển khai trò chơi:
- Hoàn thành trong việc triển khai trò chơi Flappy Bird trên mạch ESP32, từ việc hiển thị chim và ống nước đến tính điểm.
- Giao diện người dùng cho phép người chơi thấy điểm số, trạng thái game và điều khiển chim.

#### **4.2.2 Nhược điểm:**

- Không lưu được điểm cho người dùng sau xem mà chỉ hiển thị điểm số lúc đó.
- Chưa có phần Menu mà chỉ có một nút để bấm bắt đầu game.

**Tự chấm: 9/10.**

## Mục lục

<b>Chương 1: Tổng quan đề tài đồ án.....</b>	<b>1</b>
1.1. Lý do chọn đề tài: .....	1
1.2. Mục tiêu:.....	1
<b>Chương 2: Các phần tử trong mạch: .....</b>	<b>1</b>
2.1. Sơ đồ khối của mạch:.....	1
2.2. Giới thiệu về vi điều khiển ESP32: .....	2
2.3. Giới thiệu về màn hình hiển thị 128x64 SH1106:.....	3
<b>Chương 3: Thiết kế hệ thống:.....</b>	<b>4</b>
3.1. Ý tưởng: .....	4
3.2. Sơ đồ kết nối các linh kiện:.....	6
3.3. Lưu đồ thuật toán: .....	7
3.4. Hiện thực phần cứng:.....	7
3.4.1. Thực hiện trên breadboard: .....	7
3.4.2. Tiến hành làm PCB:.....	8
<b>Chương 4: Kết quả đạt được và đánh giá: .....</b>	<b>11</b>
4.1. Kết quả đạt được .....	11
4.1.1. Hiểu về game Flappy Bird .....	11
4.1.2. Lập trình ESP32 .....	11
4.1.3. Giao tiếp với màn hình và cảm biến .....	11
4.1.4. Tối ưu hóa và thử nghiệm .....	12
4.2. Đánh giá .....	
4.2.1. Ưu điểm .....	12
4.2.2. Nhược điểm .....	12