ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

---ᴕᴕ---

BÁO CÁO BÀI TẬP LỚN

Đề tài: Flappy bird trên ESP32

Giảng viên hướng dẫn:	ThS. Nguyễn Đức Tiến	
Sinh viên - MSSV:	Đặng Tiến Đạt	20200133
	Nguyễn Minh Hiếu	20204830
Lớp:	CTTN - Khoa học máy tính	
Học phần:	Hệ nhúng	
Mã học phần:	IT4210	
Mã lớp:	143800	
Khóa:	K65	

Hà Nội, 01/2024

1. Giới thiệu

a. Đề tài

Bài tập lớn này triển khai trò chơi nổi tiếng Flappy Bird trên ESP32 sử dụng kiến trúc Arduino. Nhóm lựa chọn đề tài này do tựa game Flappy bird đã chứng tỏ được sự thành công của mình, đồng thời điều khiển, tương tác của trò chơi phù hợp với ESP32. Các tính năng chính bao gồm:

- Điều khiển: Người chơi sử dụng nút bấm để khiến nhân vật bay lên.
- Màn hình OLED: Trò chơi được thiết kế để hiển thị trên màn hình OLED 0.96 inches.
- Điểm cao: Trò chơi lưu high score của người chơi, và điểm cao có thể được reset.

b. Phân công nhiệm vụ thành viên

Thành viên	Nhiệm vụ
Nguyễn Minh Hiếu (nhóm trưởng)	Load ảnh .xbm trên SSD1306; cấu hình các chân trên ESP32; thiết kế, lắp, hàn mạch; viết báo cáo; quay video demo.
Đặng Tiến Đạt	Thiết kế các trạng thái của game; lập trình logic game (di chuyển, va chạm, tính điểm, lưu điểm).

2. Cấu hình mạch

a. Danh sách thiết bị

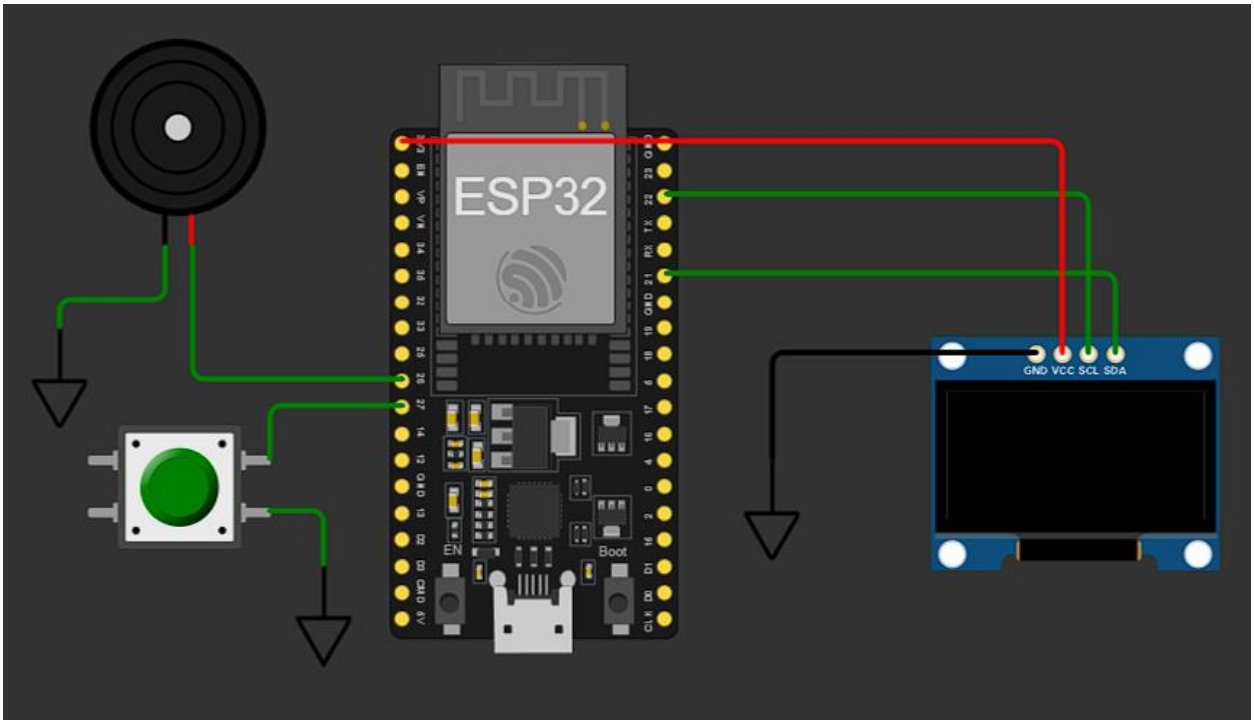
- Mạch tích hợp ESP32.
- Màn hình SSD1306 OLED.
- Nút bấm.
- Còi chíp (buzzer).
- Bảng mạch, dây nối.

b. Sơ đồ mạch

Màn hình SSD1306 OLED nối với ESP32 bằng giao thức I2C (SCL trên SSD1306 nối với chân 22 của ESP32, SDA trên SSD1306 nối với chân 21 của ESP32).

Nút bấm nối 1 chân với chân 23 của ESP32 (có thể thay đổi), chân còn lại nối đất, không cần mắc nối tiếp với điện trở do sử dụng **INPUT_PULLUP**.

Còi chíp nối 1 chân với chân 18 của ESP32 (có thể thay đổi), chân còn lại nối đất.



Hình 1. Minh họa sơ đồ mạch của bài tập lớn

3. Phần mềm Flappy bird trên ESP32

a. Thư viện

- ThingPulse OLED SSD1306: Thư viện đồ họa của màn hình SSD1306 Oled cho ESP8266 và ESP32.
- Preferences: Thư viện lưu trữ dữ liệu trên bộ nhớ flash của ESP32, được sử dụng để đọc ghi high score.

b. Load ảnh dưới định dạng XBM

- Bước 1: Tải icon trên Internet dưới định dạng .jpg hoặc .png (Ví dụ: https://icons8.com/).
- Bước 2: Chuyển định dạng .jpg (.png) thành định dạng .xbm bằng trang web: https://convertio.co/png-xbm/. Cấu trúc của file .xbm có định dạng như sau:

```
#define 7dc7446c3b2b424397e5fb44a1dcd0a5E4fhpr29aaOPTCpZ_width 12
#define 7dc7446c3b2b424397e5fb44a1dcd0a5E4fhpr29aaOPTCpZ_height 12
static char 7dc7446c3b2b424397e5fb44a1dcd0a5E4fhpr29aaOPTCpZ_bits[] = {
  0xC0, 0x00, 0x47, 0x01, 0xF9, 0x07, 0x61, 0x0F, 0x20, 0x0B, 0xE2, 0x0F,
  0x22, 0x0B, 0x4C, 0x0E, 0x24, 0x0E, 0x10, 0x05, 0x18, 0x05, 0xE0, 0x01,
  };
```

- Bước 3: Copy phần dữ liệu trên, tạo 1 file header chứa tất cả các ảnh .xbm.
- Bước 4: Sử dụng hàm drawXBM() trong thư viện ThingPulse OLED SSD1306, truyền vào vị trí render hình (x, y) tự chọn và width, height, char[] image trong dữ liệu .xbm ở trên.

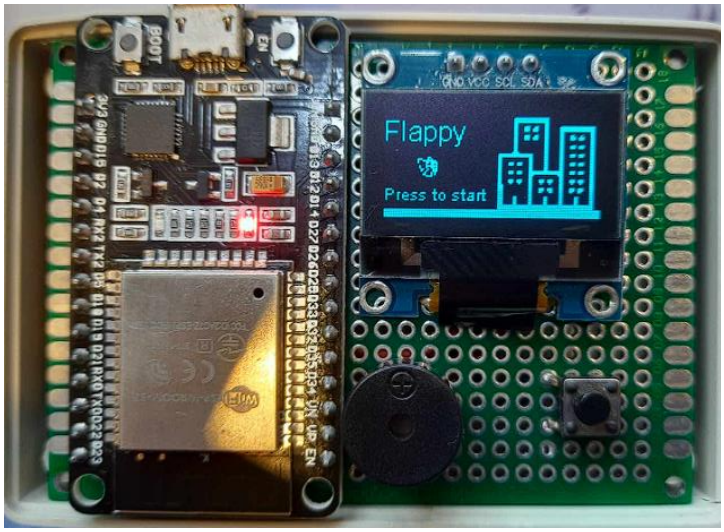
```
// Draw a XBM
void drawXbm(int16_t x, int16_t y, int16_t width, int16_t height, const uint8_t *xbm);
```

c. Cấu trúc chương trình, thuật toán

Trò chơi gồm 3 trạng thái: bắt đầu, đang chơi, và hiển thị điểm.

```
unsigned int gameState = 0;  // 0 - start, 1 - play, 2 - score
```

1. Trong trạng thái “bắt đầu”, màn hình OLED hiển thị các hình ảnh của game và text hướng dẫn, đồng thời khởi tạo các giá trị cho trạng thái đang chơi (vị trí cột, vị trí Flappy, đặt lại score = 0).

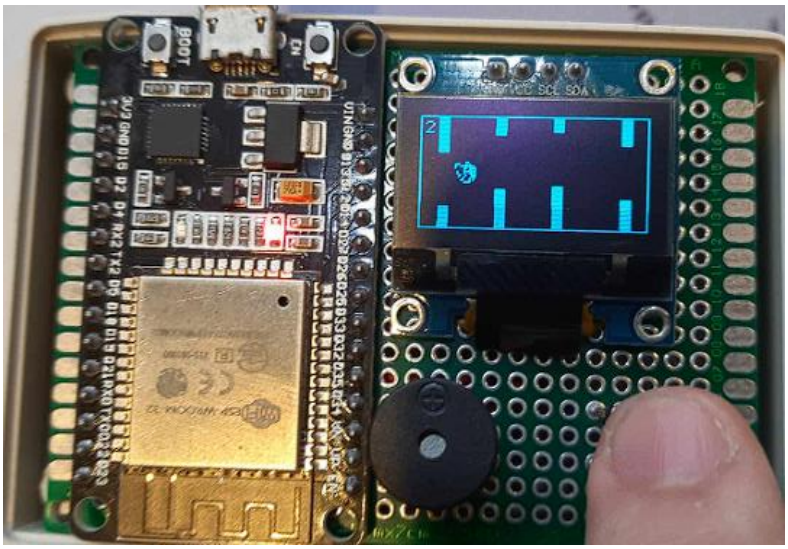


Hình 2. Trạng thái “bắt đầu”

```
if(gameState == 0) {
  // Reinitialize in-game status
  birdY = 28.00;
  score = 0;
  speed = 0.01;

  display.setFont(ArialMT_Plain_16);
  display.drawString(0, 4, "Flappy ");

  display.drawXbm(64, 0, Building_width, Building_height, Building);
  display.drawXbm(birdX, birdY, Flappy_width, Flappy_height, Flappy);
  display.setColor(WHITE);
  display.fillRect(0, SCREEN_HEIGHT - 5, SCREEN_WIDTH, 5);

  display.setFont(ArialMT_Plain_10);
  display.drawString(0, 44, "Press to start");

  // Reinitialize all tubes
  for(int i = 0; i < 4; i++) {
```

```
    tubeX[i] = 128 + ((i+1) * TUBE_DISTANCE);
    bottomTubeHeight[i] = random(8, 32);
    hasScored[i] = false;
  }

  if(digitalRead(BUTTON_PIN) == LOW) {
    gameState = 1;
    delay(50);
  }
}
```

2. Trong trạng thái “chơi”, trò chơi liên tục cho các cột di chuyển sang trái, khiến Flappy bay lên và còi kêu khi bấm nút, tính điểm, đồng thời kiểm tra va chạm.



Hình 3. Trạng thái “chơi”

- Di chuyển các cột sang trái và tính điểm:

```
for(int i = 0; i < 4; i++) {
  //Move all tube to the left
  tubeX[i] -= speed;

  // If a tube pass the bird, add a point
  if(tubeX[i] < birdX && !hasScored[i]){
    score++;
    hasScored[i] = true;

    // Increase speed every 10 tubes
    if(score % 10 == 0){
      speed += 0.01;
    }
  }
```

```
  // If a tube pass the screen, reinitialize that tube on the right of the screen
  if(tubeX[i] + TUBE_WIDTH < 0){
    bottomTubeHeight[i] = random(8, 32);
    tubeX[i] = 128;
    hasScored[i] = false;
  }
}
```

- Khiến Flappy bay lên và còi kêu khi bấm nút:

```
// Setup variables and flags if button is pressed
if(digitalRead(BUTTON_PIN) == LOW) {
keyPressTime = millis();
  isFlyingUp = true;
  isBuzzerOn = true;
}

// The bird will fly up for 80 milliseconds
if((keyPressTime + 80) < millis()) {
  isFlyingUp = false;
}

// The buzzer will make sound for 10 milliseconds
if((keyPressTime + 10) < millis()) {
  isBuzzerOn = false;
}
```

- Kiểm tra va chạm với 1 trong 2 điều kiện: Chạm biên trên/dưới màn hình

```
if(birdY > 63 || birdY < 0) { // Check if out of bound on vertical axis
```

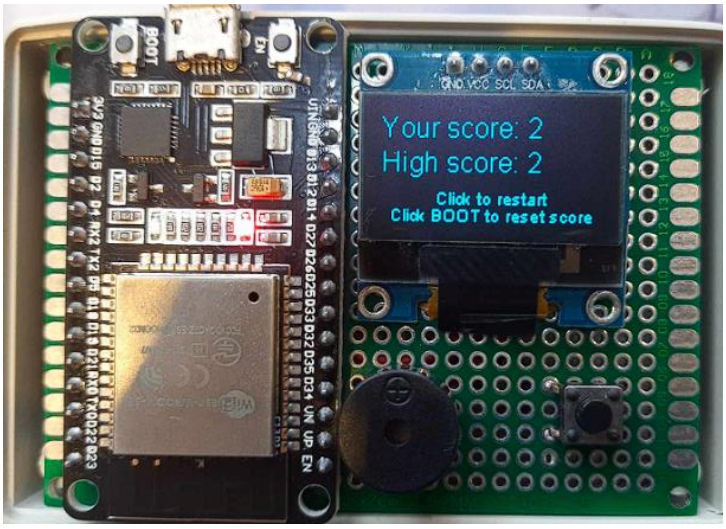
hoặc va chạm với 1 trong các cột

```
//Check for collision with tube
for(int i = 0; i < 4; i++){
  if(tubeX[i] <= birdX + 7 && birdX + 7 <= tubeX[i] + 6) {
    if(birdY < bottomTubeHeight[i] || birdY + 8 > bottomTubeHeight[i] +
PATH_WIDTH){
```

Khi đó, kiểm tra nếu score hiện tại lớn hơn high score, cập nhật high score mới:

```
if(score > highScore){
  highScore = score;

  // Write new high score to flash memory
  preferences.begin("Flappy", false);
  preferences.putUInt("highScore", highScore);
  preferences.end();
}
```

3. Trạng thái “hiển thị điểm”, màn hình OLED hiển thị score màn chơi vừa rồi và high score:



Hình 4. Trạng thái hiển thị điểm”

```
display.setFont(ArialMT_Plain_16);
display.drawString(0, 0, "Your score: " + String(score));
display.drawString(0, 20, "High score: " + String(highScore));

display.setFont(ArialMT_Plain_10);
display.drawString(32, 44, "Click to restart");

display.setFont(ArialMT_Plain_10);
display.drawString(5, 54, "Click BOOT to reset score");
```

Ngoài ra, trong trạng thái này, người chơi có thể bấm nút BOOT trên ESP32 để reset điểm về 0.

```
// If BOOT button is pressed, reset high score in game and in the flash memory
if(digitalRead(BOOT_BUTTON_PIN) == LOW){
  score = 0;
  highScore = 0;
  // Write high score (= 0) to flash memory
  preferences.begin("Flappy", false);
  preferences.putUInt("highScore", highScore);
  preferences.end();
}
```

4. Code, hướng dẫn cài đặt

- Code github cùng hướng dẫn cài đặt: https://github.com/nonameex1sts/FlappyBird_ESP32
- Video demo: https://youtu.be/28IEaFuHusE

5. Tài liệu tham khảo

- Slide, tài liệu môn học Hệ nhúng - IT4210.
- Arduino library documentation: https://www.arduino.cc/reference/en/libraries/
- Thư viện: ThingPulse OLED SSD1306 và Preferences.