

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

\_\_\_\_\_ \* \_\_\_\_\_



# BÁO CÁO ĐỒ ÁN MÔN HỌC

## Môn Hệ Nhúng

**Phát nhạc với ESP32 dùng DAC**

Giảng viên hướng dẫn: TS.Ngô Lam Trung

**Nhóm 6**  
Sinh viên  
Đào Xuân Đạt

Mã lớp học: **139318**  
Mã số sinh viên  
20205065

*Hà Nội, tháng 7 năm 2023*

# Mục lục

<b>I. Mục tiêu.....</b>	<b>3</b>
<b>II. Nội dung thực hiện .....</b>	<b>4</b>
<b>1. Mô hình hệ thống.....</b>	<b>4</b>
<b>2. Sơ đồ đầu nối .....</b>	<b>7</b>
<b>3. Lập trình .....</b>	<b>9</b>
<b>III. Kết quả .....</b>	<b>15</b>
<b>Note: Những công việc muốn phát triển thêm. ....</b>	<b>17</b>
<b>IV. Phân chia công việc .....</b>	<b>18</b>

## **I. Mục tiêu**

Khi cuộc sống ngày càng trở lên hiện đại, việc mà người dùng muốn trải nghiệm cuộc sống hiện đại, những công nghệ mới hay những thứ giúp họ có thể giải trí mỗi ngày càng cao. Chính vì những nhu cầu đó của người dùng mà em đã hướng tới một giải pháp đó chính là hệ thống phát nhạc sử dụng ESP32 với đầu vào là thẻ SD và đầu ra DAC.

Project của em hướng tới việc người dùng có thể trải nghiệm những thức nhạc giúp họ giải tỏa đi những áp lực trong cuộc sống hiện tại.

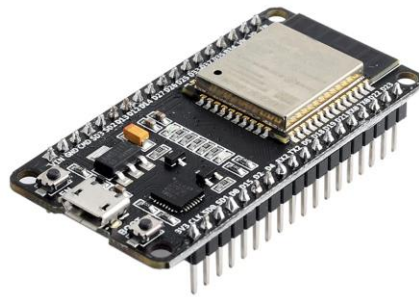
Hệ thống hoạt động theo nguyên lý tự động phát nhạc khi được cấp điện, phát nhạc được đọc từ thẻ SD và có thể tùy trình được bài hát mong muốn hay tạm dừng phát khi người dùng bận làm một việc gì đó khác và sau đó sẽ quay trở lại. Hệ thống có đi kèm với 1 màn hình cho phép người dùng có thể biết được tên bài nhạc được phát hiện tại.

Đồ án này sẽ không được hoàn thành nếu không có lời khuyên bổ ích và sự hỗ trợ nhiệt tình mà em nhận được từ thầy - TS.Ngô Lam Trung. Mặc dù rất bận rộn trong công việc giảng dạy, nghiên cứu nhưng thầy vẫn dành thời gian quý báu của mình để hướng dẫn em hoàn thiện đồ án môn học này.

## II. Nội dung thực hiện

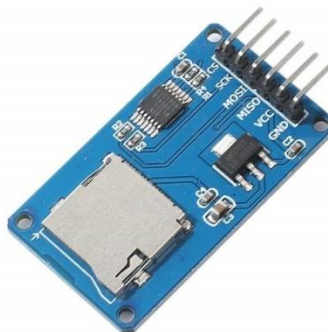
### 1. Mô hình hệ thống

- Hệ thống gồm có:
  - Bộ điều khiển chính ESP32 Devkit v1: thực hiện chức năng chính trong việc đọc các dữ liệu đầu vào ra phát ra thông qua DAC GPIO25 và GPIO26



Hình 1: ESP32 Devkit v1

- Module MicroSD Card Adapter: để đọc được dữ liệu được lưu trữ trong thẻ nhớ SD



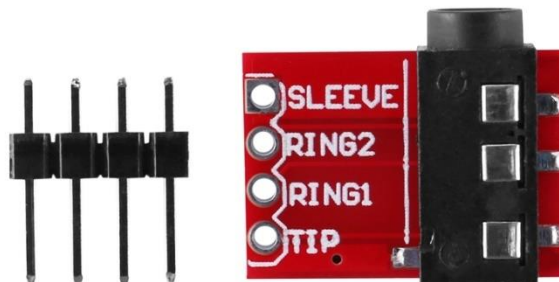
Hình 2: Module MicroSD Card Adapter

- Module khuếch đại âm thanh PAM8403: có chức năng nhận tín hiệu đầu ra từ ESP32 qua DAC và khuếch đại tín hiệu đó ra với công suất 3W



Hình 3: Module khuếch đại PAM8403

- Module chuyển đổi âm thanh sang tai nghe TRRS: nhận tín hiệu từ module khuếch đại và phát ra thông qua tai nghe được cắm vào.



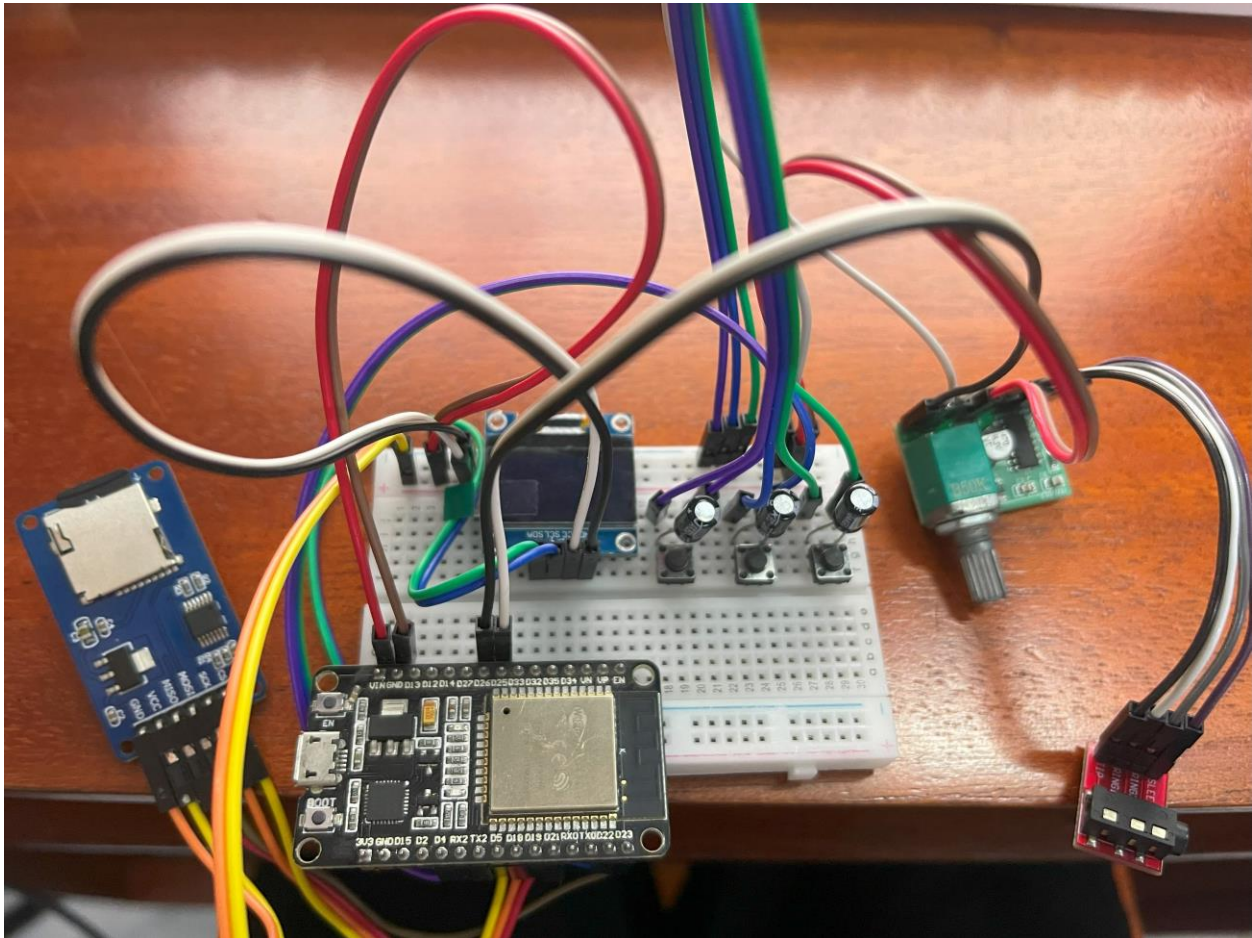
Hình 4: Module chuyển đổi âm thanh TRRS

- 3 nút bấm: tương đương với 3 chức năng Back, Next và Play Pause bài.
- 1 Màn hình Oled SSD1306: được sử dụng hiển thị các thông tin như tên bài hát đang được phát hay chức năng được thực hiện như Back bài hay Next bài,....



Hình 5: Màn hình Oled SSD1306

## 2. Sơ đồ đấu nối

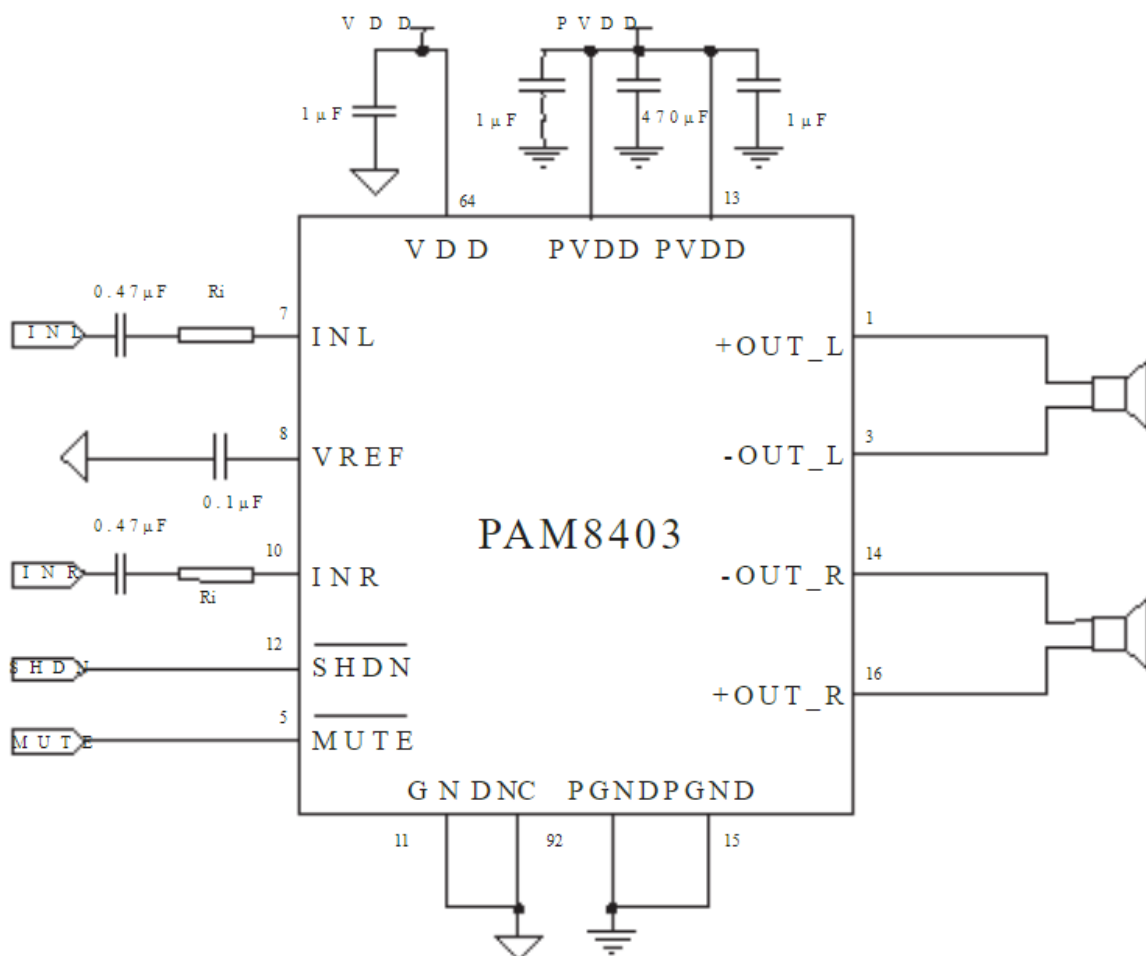


Hình 6: Tổng quan về sơ đồ của hệ thống

### Giải thích về sơ đồ đấu nối:

1. Bộ điều khiển chính kết nối với module đọc dữ liệu MicroSD Card qua chuẩn giao thực SPI gồm có các chân đầu vào GPIO23, 19, 18, 5 tương đương với MOSI, MISO, SCK và CS.
2. Bộ điều khiển chính kết nối với module khuếch đại âm thanh qua đầu ra DAC của mình từ 2 chân GPIO25 và GPIO26.
3. Bộ khuếch đại âm thanh được nối với đầu vào của module TRRS với 2 đầu vào TIP và RING1 là 2 đầu vào trái phải của tai nghe khi được phát ra. RING2 là đầu ra của Mic nhưng trong bài toán hiện tại chưa thực hiện điều đó nên được đấu nối với Ground.
4. Bộ điều khiển chính kết nối với màn hình oled SSD1306 hiển thị các thông tin cần thiết.

5. Các nút bấm được thiết kế nối với đầu vào là GPIO4, 16, 17 tương ứng với các chức năng như Back, Play Pause, Next nhạc. Button được kéo xuống đất đồng thời với 1 tụ giá trị 1uF để tránh trường hợp nút bấm gửi các giá trị sai lệnh (debounce). Link tài liệu: [Debounce cho nút nhấn bằng tụ điện | Cộng đồng Arduino Việt Nam](#)
6. Ngoài ra việc phát nhạc qua đầu ra DAC có hiện tượng nhiễu, tuy nhiên trên module khuếch đại PAM8403 đã có sẵn bộ lọc để làm chúng bằng cách đấu nối đầu vào module bằng 1 bộ lọc RC với giá trị của tụ điện là 0.47uF











### 3. Lập trình

Tổng quan về chương trình:

- Chương trình được chia thành 2 phần gồm có hàm chính và 1 module được import vào được sử dụng để lấy ra được thông tin tên bài hát được gọi đến.

 Final_Project.ino		8/8/2023 11:09 PM	INO File	6 KB
 GetFileNameUtils.cpp		8/8/2023 8:15 PM	CPP File	3 KB
 GetFileNameUtils.h		8/8/2023 8:15 PM	H File	1 KB

- Trong hàm chính đã include lại module được phân tách ra ngoài nhằm dễ dàng thực thi debug hay tái sử dụng chúng trong các bài toán tương tự.

```
7  #include "GetFileNameUtils.h"
```

- Phân tích về module chức năng phụ gồm có 2 function để lấy ra tên bài tương ứng với các chức năng back bài và next bài

```
1  #ifndef GET_FILE_NAME_UTILS_H
2  #define GET_FILE_NAME_UTILS_H
3
4  #include "FS.h"
5
6  String getBackFileName(fs::FS &SD, String fileName);
7  String getNextFileName(fs::FS &SD, String fileName);
8
9  #endif // GET_FILE_NAME_UTILS_H
```

- Với chức năng lấy ra tên của file nhạc trước đó. Chương trình kiểm tra với tên file nhạc đang phát hiện tại có phải là file đầu tiên trong thẻ:
  - Chuyển tới cuối thẻ và lấy ra file cuối cùng nếu file nhạc phát trước đó là file đầu tiên.

```

15 // If fileName is first of SD Card
16 if (String(dirFirst.openNextFile().name()) == fileName) {
17     // LastFile is last of SD Card
18     File lastFile;
19     while (true) {
20         File entry = dirFirst.openNextFile();
21         if (!entry) {
22             break;
23         }
24         // Search and find is lastFile of SD Card
25         if (!entry.isDirectory()) {
26             lastFile = entry;
27         }
28     }
29     if (!lastFile) {
30         return "File can't found 1";
31     }
32     return String(lastFile.name());
33 }

```

- Chuyển tới file trước đó nếu file nhạc phát trước đó không là file đầu tiên trong thẻ.

```

35 File prevFile;
36 // Get PrevFileName and lastFileName
37 while (true) {
38     File currentFile = dir.openNextFile();
39     if (!currentFile) {
40         break;
41     }
42     // Search and file is lastFile
43     if (!currentFile.isDirectory() && String(currentFile.name()) == fileName) {
44         break;
45     }
46     // If currentFile is not fileName and currentFile go to fileName => currentFile is prev of fileName in SD
47     if (String(currentFile.name()) != fileName) {
48         prevFile = currentFile;
49     }
50 }
51
52 if (!prevFile) {
53     return "File can't found 2";
54 }
55
56 return String(prevFile.name());
57 prevFile.close();

```

- Tương tự với chức năng lấy ra tên file nhạc sau cũng sẽ kiểm tra file nhạc đang file có đang ở cuối thẻ không và làm các công việc tương tự

```

60 String getNextFileName(fs::FS &SD, String fileName) {
61     // Search and find fileName
62     File dir = SD.open("/");
63
64     // Check open folder
65     if (!dir) {
66         return "Unable to open folder";
67     }
68
69     // Get lastFileName
70     while (true) {
71         File entry = dir.openNextFile();
72         if (!entry) {
73             break;
74         }
75         // Search and file is lastFile
76         if (!entry.isDirectory() && String(entry.name()) == fileName) {
77             break;
78         }
79         entry.close();
80     }
81
82     // Get nextFileName of fileName
83     File nextFile = dir.openNextFile();
84
85     // if nextFile is not found (dir pointer is null "lastFile of SD")
86     if (!nextFile) {
87         File dirNew = SD.open("/");
88         File firstFile = dirNew.openNextFile();
89         firstFile = dirNew.openNextFile();
90         if (!firstFile) {
91             return "Unable to open folder";
92         } else {
93             return String(firstFile.name());
94         }
95     } else {
96         return String(nextFile.name());
97     }
98     nextFile.close();
99     return "NULL";
100 }

```

- Trong chương trình chính thực hiện các khởi tạo cho các module được khởi chạy như SSD1306, SPI cho MicroSD Card hay thư viện phát nhạc ESP32AudioI2S  
Link thư viện: [schreibfaul1/ESP32-audioI2S: Play mp3 files from SD via I2S \(github.com\)](https://github.com/schreibfaul1/ESP32-audioI2S)
- Setup các thông số cho các đầu vào và ra qua các chân GPIO. Khởi tạo đầu ra cho audio nhạc với 2 chân DAC GPIO25 và 26, đồng thời set mức âm lượng của nó về 12/21 (Mức âm lượng được thư viện hỗ trợ sẵn)

```

63 void setup() {
64     // Setup SPI mode using read data from SD card
65     pinMode(SD_CS, OUTPUT);
66     digitalWrite(SD_CS, HIGH);
67     SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI);
68     SPI.setFrequency(1000000);
69
70     // Setup serial baud rate
71     Serial.begin(115200);
72     SD.begin(SD_CS);
73
74     // Setup button in. Use INPUT_PULLUP to avoid a hovering button situation where there is no defined signal when the button is not pressed
75     pinMode(Back, INPUT_PULLUP);
76     pinMode(Play_Pause, INPUT_PULLUP);
77     pinMode(Next, INPUT_PULLUP);
78
79     //Init display start.
80     if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
81         Serial.println(F("SSD1306 allocation failed"));
82         for (;;)
83             ; // Don't proceed, loop forever
84     }
85
86     // Setup first play audio
87     fileName = "Hero.m4a";
88     printStateAndFileName("First Audio.....");
89     audio.setPinout(0, I2S_DOUT, I2S_LRC);
90     audio.setVolume(12); // 0...21
91
92     audio.connecttoFS(SD, fileName.c_str());
93 }

```

- Thực hiện phát nhạc và nhận các tín hiệu từ nút bấm hay serial để chuyển nhạc hay dừng nhạc

```

95 void loop() {
96     audio.loop();
97     checkBackButton();
98     checkPlayPauseButton();
99     checkNextButton();
100     if (Serial.available()) { // put streamURL in serial monitor
101         audio.stopSong();
102         Serial.println(fileName);
103         fileName = Serial.readString();
104         fileName.trim();
105         fileName = fileName + ".m4a";
106         Serial.println(fileName);
107         printStateAndFileName("Go to.....");
108         if (fileName.length() > 1) {
109             audio.connecttoFS(SD, fileName.c_str());
110         }
111         log_i("free heap=%i", ESP.getFreeHeap());
112     }
113 }

```

- Thực hiện kiểm tra giá trị đầu vào từ button và thực hiện lấy ra tên file trước đó khi nút bấm được nhấn

```

115 void checkBackButton() {
116     backCurrentState = digitalRead(Back);
117     if (backCurrentState != backLastState && millis() - backLastTime > 500 && backCurrentState == LOW) {
118         backLastTime = millis();
119         fileName = getBackFileName(SD, fileName);
120         printStateAndFileName("Back Audio.....");
121         audio.connecttoFS(SD, fileName.c_str());
122     }
123     // save the the last state
124     backLastState = backCurrentState;
125 }

```

- Tương tự với hàm thực hiện kiểm tra nút nhấn và lấy ra file sau đó.

```

149 void checkNextButton() {
150     nextCurrentState = digitalRead(Nxt);
151     if (nextCurrentState != nextLastState && millis() - nextLastTime > 500 && nextCurrentState == LOW) {
152         nextLastTime = millis();
153         fileName = getNextFileName(SD, fileName);
154         printStateAndFileName("Next Audio.....");
155         audio.connecttoFS(SD, fileName.c_str());
156     }
157     // save the the last state
158     nextLastState = nextCurrentState;
159 }

```

- Thực hiện kiểm tra giá trị của đầu vào nút bấm và dừng hay tiếp tục phát nhạc khi nút bấm được nhấn thông qua 1 cờ được xác định là trạng thái của nhạc đang được phát hay không để chuyển trạng thái. Khi dừng nhạc hay phát đều sử dụng chung 1 hàm **audio.pauseResume()** được thư viện ESP32AudioI2S hỗ trợ thực hiện dừng hay tiếp tục phát nhạc.

```

127 void checkPlayPauseButton() {
128     playPauseCurrentState = digitalRead(Play_Pause);
129     if (playPauseCurrentState != playPauseLastState && millis() - playPauseLastTime > 500) {
130         // Ghi nhận thời gian nhấn nút cuối cùng
131         playPauseLastTime = millis();
132
133         // Cập nhật trạng thái nút trước đó
134         playPauseLastState = playPauseCurrentState;
135         if (playPauseCurrentState == LOW) {
136             if (playOrPause == 0) {
137                 printStateAndFileName("Play Audio.....");
138                 audio.pauseResume();
139                 playOrPause = 1;
140             } else if (playOrPause == 1) {
141                 printStateAndFileName("Pause Audio.....");
142                 audio.pauseResume();
143                 playOrPause = 0;
144             }
145         }
146     }
147 }

```

- Ngoài ra mỗi khi thực hiện các thay đổi đều được hiển thị lên màn hình.

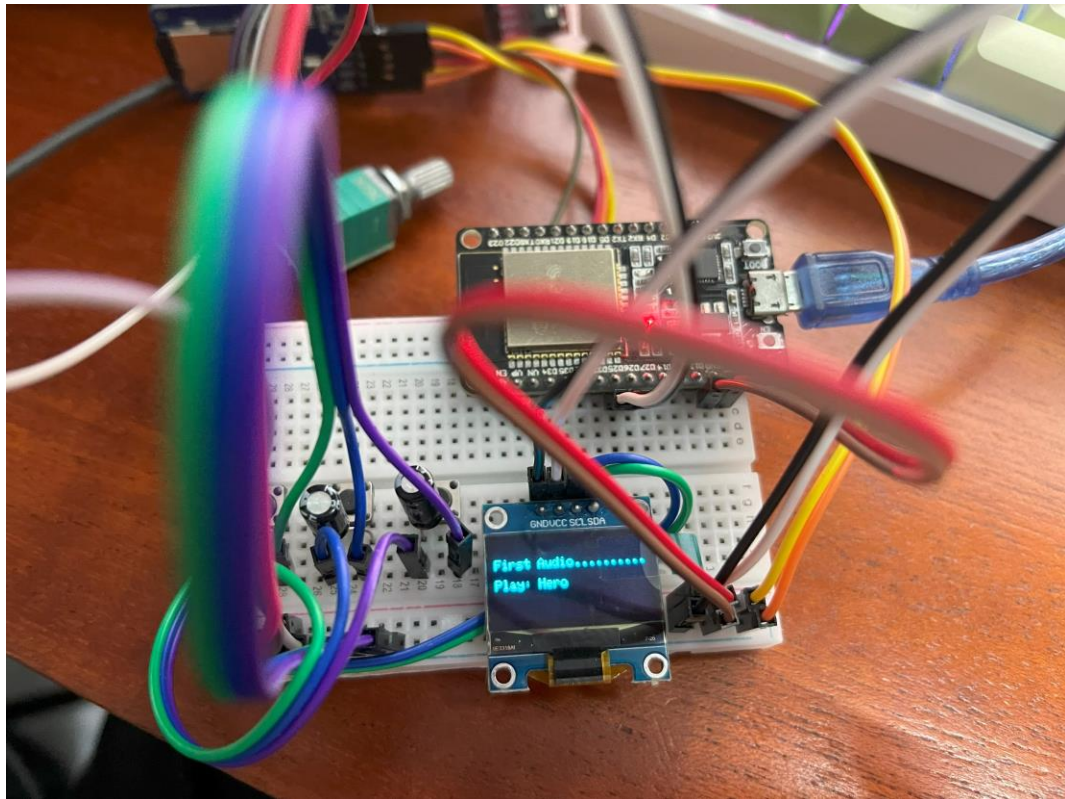
```
161 void printStateAndFileName(String nameButton) {  
162  
163     int dotIndex = fileName.lastIndexOf(".");  
164     String audioName = "Play: " + fileName.substring(0, dotIndex);  
165  
166     display.clearDisplay();  
167     display.setTextSize(1);  
168     display.setTextColor(WHITE);  
169     display.setCursor(0, display.height() / 4);  
170     display.println(nameButton);  
171     display.setCursor(0, display.height() / 2);  
172     display.println(audioName);  
173     display.display();  
174 }
```

Toàn bộ mã nguồn chương trình đã được đẩy lên github:

Link: [xd3011/ESP32\\_Audio\\_DAC \(github.com\)](https://github.com/xd3011/ESP32_Audio_DAC)

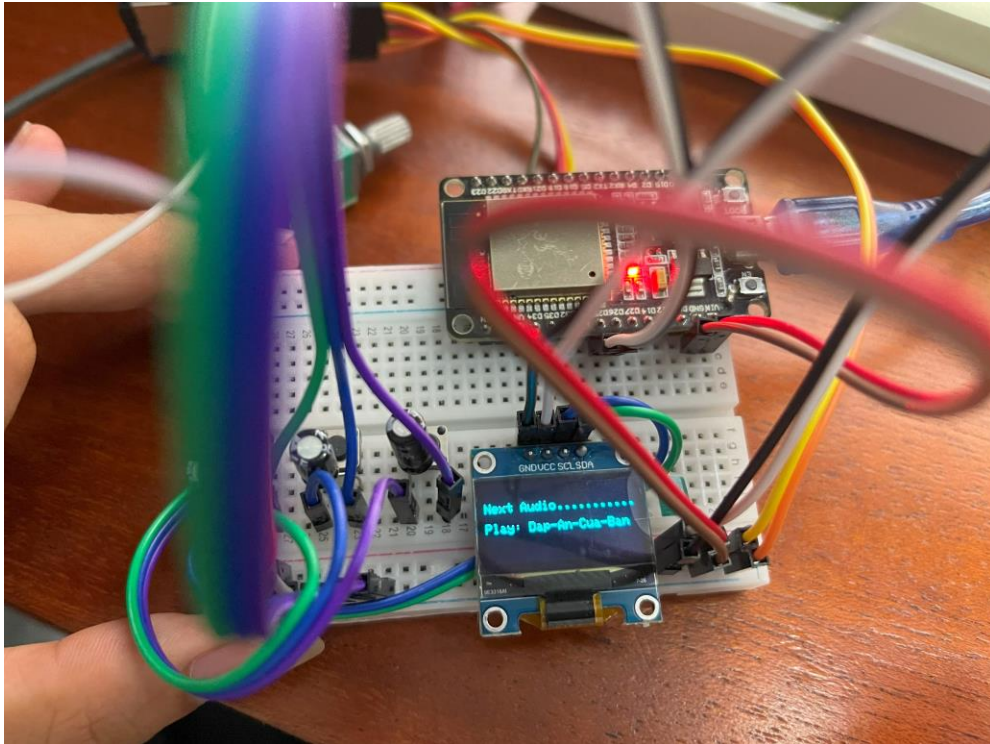
### III. Kết quả

Em hoàn thiện thành công hệ thống phát nhạc từ ESP32 qua DAC với đầu vào đọc từ thẻ nhớ SD và đầu ra là tai nghe. Dù đã có kết quả như mong đợi tuy nhiên vẫn còn khuyết điểm như nhiễu từ đầu ra của DAC hay giá trị đầu ra từ module khuếch đại chỉ nhận được 1 đầu vào (do module PAM8403 hiện tại). Tuy nhiên dựa trên những gì đã làm được thì đây là một nghiên cứu có thể được phát triển lên những mức độ cao hơn.

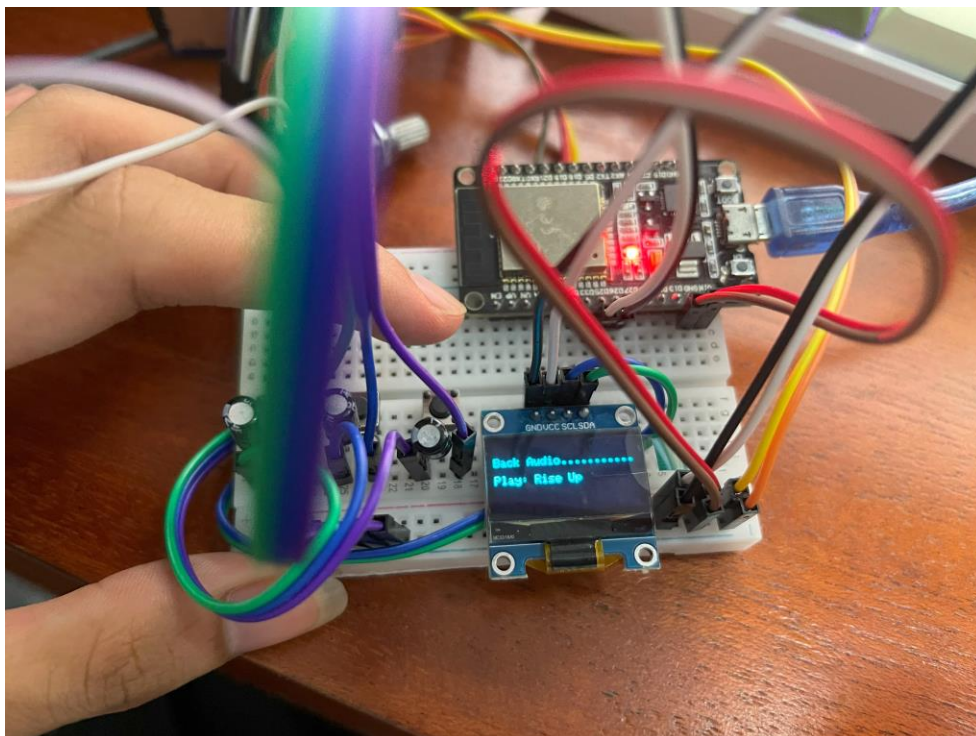


Hình 11: Phát nhạc với bài đầu tiên



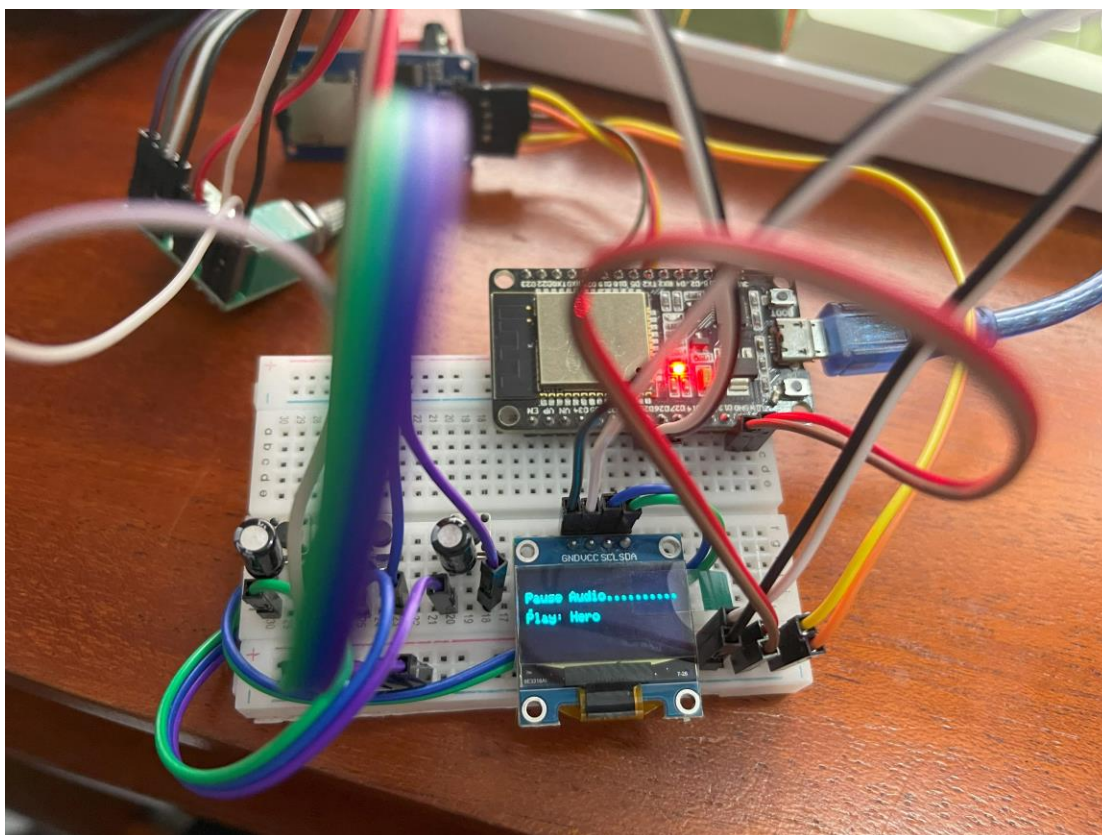


Hình 12: Thực hiện Next bài



Hình 13: Thực hiện Back bài





Hình 14: Thực hiện Pause nhạc

Ngoài những hình ảnh trên, việc demo chương trình với phát nhạc sẽ được tạo video gửi kèm khi báo cáo với âm thanh được phát khi chạy chương trình thực.

**Note: Những công việc muốn phát triển thêm.**

Ngoài những công việc đã làm được ở trên, với bài toán đề ra em mong muốn rằng mình sẽ:

Thêm chức năng tăng giảm âm lượng trực tiếp bằng nút bấm khi trong trường hợp module PAM8403 được thay thế bằng các module khác tương tự không có kèm biến trở tăng giảm được âm lượng (Do không đủ linh kiện)

Thêm chức năng sử dụng nguồn nhạc online thông qua ESP32. ESP32 có khả năng sử dụng wifi độc lập và có thể lấy được dữ liệu nhạc trên môi trường mạng.

Đóng gói toàn bộ module thành 1 sản phẩm có thể sử dụng thực sự trên thực tế (Điều này có thể nhưng với các sản phẩm trên thị trường hiện nay thì với các chức năng hay những ưu điểm của mô hình trên là chưa đủ để đáp ứng được thị trường)

## IV. Phân chia công việc

Công việc	Xuân Đạt
Lên ý tưởng và xây dựng mô hình sản phẩm	100%
Thiết kế mạch	100%
Lập trình	100%

### Tổng hợp các tài liệu đã sử dụng:

1. Thư viện ESP32AudioI2S: [schreibfaul1/ESP32-audioI2S: Play mp3 files from SD via I2S \(github.com\)](https://github.com/schreibfaul1/ESP32-audioI2S)
2. Chỉnh sửa giá trị của nút bấm bị debounce thông qua việc ghép nối thêm tụ điện: [Debounce cho nút nhấn bằng tụ điện | Cộng đồng Arduino Việt Nam](#)
3. Repository code: [xd3011/ESP32\\_Audio\\_DAC \(github.com\)](https://github.com/xd3011/ESP32_Audio_DAC)