BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO THỰC TẬP CƠ SỞ**

**TÌM HIỂU CẤU TRÚC FILE ÂM THANH (WAV), THỂ HIỆN ĐỒ HỌA DẠNG SÓNG VÀ PHỔ VÀ CÁC THÔNG TIN HEADER**

**Giảng viên hướng dẫn: Nguyễn Thị Hương Lý**

**Sinh viên thực hiện: Mạnh Văn Hiệp**

**Mã số sinh viên: 61133631**

Khánh Hòa – 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO THỰC TẬP CƠ SỞ**

**TÌM HIỂU CẤU TRÚC FILE ÂM THANH (WAV), THỂ HIỆN ĐỒ HỌA DẠNG SÓNG VÀ PHỔ VÀ CÁC THÔNG TIN HEADER**

GVHD: Nguyễn Thị Hương Lý

SVTH: Mạnh Văn Hiệp

MSSV: 61133631

Khánh Hòa – Tháng 01/2022

# LỜI CẢM ƠN

Để hoàn thành bài báo cáo thực tập cơ sở này, trước hết em xin gửi đến quý thầy, cô Khoa Công nghệ Thông tin - Trường Đại học Nha Trang lời cảm ơn chân thành.

Em xin gửi đến cô Nguyễn Thị Hương Lý, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành bài báo cáo này lời cảm ơn sâu sắc nhất.

Trong quá trình nghiên cứu thực hiện đề tài, cũng như là trong quá trình làm bài báo cáo thực tập cơ sở, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do kiến thức cũng như kinh nghiệm thực tiễn của bản thân còn hạn chế nên đề tài, bài báo cáo cơ sở này khó thể không tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp từ quý thầy, cô để em học hỏi thêm được nhiều kinh nghiệm, cũng như kỹ năng cần thiết.

Em xin chân thành cảm ơn!

# MỤC LỤC

[LỜI CẢM ƠN i](#_Toc92485108)

[MỤC LỤC ii](#_Toc92485109)

[PHẦN MỞ ĐẦU 1](#_Toc92485110)

[1. Lý do chọn đề tài: 1](#_Toc92485111)

[2. Mục tiêu của đề tài. 1](#_Toc92485112)

[3. Phạm vi nghiên cứu. 1](#_Toc92485113)

[4. Phương pháp nghiên cứu. 1](#_Toc92485114)

[Chương 1. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU 2](#_Toc92485115)

[1.1 CƠ SỞ LÝ THUYẾT 2](#_Toc92485116)

[1.1.1 Tổng quan về file âm thanh (wav) 2](#_Toc92485117)

[1.1.1.1 Mô tả 3](#_Toc92485118)

[1.1.1.2 Chi tiết kỹ thuật 4](#_Toc92485119)

[1.1.1.3 Cách mở file WAV, WAVE 5](#_Toc92485120)

[1.1.1.4 WAV – Waveform Audio File Format 6](#_Toc92485121)

[1.1.2 Biến đổi Fourier 9](#_Toc92485178)

[1.1.2.1 Ứng dụng 9](#_Toc92485179)

[1.1.2.2 Các dạng của biến đổi Fourier 10](#_Toc92485180)

[1.2 CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH ĐƯỢC SỬ DỤNG 12](#_Toc92485181)

[1.2.1 Tổng quan về VISUAL STUDIO 2019 12](#_Toc92485182)

[1.2.2 Ngôn ngữ C++ 13](#_Toc92485183)

[1.2.3 Ngôn ngữ C# 14](#_Toc92485184)

[Chương 2. Đặc tả bài toán 15](#_Toc92485185)

[2.1 BÀI TOÁN 15](#_Toc92485186)

[2.2 Hướng xử lý và các thuật toán để thực hiện các yêu cầu: 15](#_Toc92485187)

[Mức 1: Đọc và thể hiện các thông tin header của file wav, hiển thị dạng sóng 15](#_Toc92485188)

[Đọc thông tin header file wav 15](#_Toc92485189)

[Hiển thị dạng sóng 15](#_Toc92485190)

[Chương 3. CÀI ĐẶT CHƯƠNG TRÌNH 16](#_Toc92485191)

[3.1 Code chương trình 16](#_Toc92485192)

[3.1.1 Mức 1 16](#_Toc92485193)

[3.1.1.1 Đọc thông tin header file wav 16](#_Toc92485194)

[3.1.1.2 Hiển thị dạng sóng 18](#_Toc92485195)

[3.1.2 Mức 2 21](#_Toc92485196)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22](#_Toc92485197)

[4.1 KẾT QUẢ ĐẠT ĐƯỢC 22](#_Toc92485198)

[4.2 ƯU ĐIỂM 22](#_Toc92485199)

[4.3 HẠN CHẾ 22](#_Toc92485200)

[4.4 HƯỚNG PHÁT TRIỂN 22](#_Toc92485201)

[4.5 KẾT LUẬN 22](#_Toc92485202)

[TÀI LIỆU THAM KHẢO 23](#_Toc92485203)

# PHẦN MỞ ĐẦU

1. **Lý do chọn đề tài:**

**Công nghệ thông tin** là một trong những ngành phát triển vượt bậc trong những năm gần đây. Ngày nay với sự phát triển nhanh chóng của xã hội thì công nghệ thông tin được ứng dụng rộng rãi ở hầu hết tất cả các lĩnh vực và ngày càng đóng vai trò quan trọng, trở thành một phần thiết yếu trong đời sống hằng ngày. Công nghệ thông tin là một ngành đòi hỏi ở người học một nền tảng kiến thức vững chắc, sự tư duy logic cao, hiểu biết sâu rộng trên nhiều lĩnh vực. Với chúng em hiện đang là những sinh viên công nghệ thông tin cần phải có sự đầu tư, không ngừng học hỏi để nâng cao kiến thức. Do đó để củng cố lại kiến thức đã học, đề tài mà em chọn để thực hiện là: **Tìm hiểu cấu trúc file âm thanh (wav), thể hiện đồ họa dạng sóng và phổ và các thông tin header.**

1. **Mục tiêu của đề tài.**
   * Củng cố lại kiến thức đã học về các cấu trúc điều khiển, ôn tập lại kiến thức về phần mảng, chuỗi cũng như kiểu dữ liệu cấu trúc...
   * Rèn luyện kỹ năng lập trình trên ngôn ngữ C++, C#.
   * Ứng dụng lý thuyết đã học giải quyết bài toán ứng dụng cụ thể.
2. **Phạm vi nghiên cứu.**
   * Đọc thông tin header file wav, hiện thị dạng song.
   * Lý thuyết về biến đổi Fourier, sử dụng Fourier để phân tích phổ
3. **Phương pháp nghiên cứu.**
   * Tìm kiếm và nghiên cứu trên mạng Internet.

# TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

## CƠ SỞ LÝ THUYẾT

### Tổng quan về file âm thanh (wav)

WAV, được biết đến với WAVE (Waveform Audio File Format), là một tập hợp con của Microsoft’s Resource Interchange File Format (RIFF) để lưu trữ các tệp âm thanh kỹ thuật số. Định dạng này không áp dụng bất kỳ quá trình nén nào cho dòng bit và lưu trữ các bản ghi âm với tốc độ lấy mẫu và tốc độ bit khác nhau. Nó đã và đang là một trong những định dạng tiêu chuẩn cho đĩa CD âm thanh. Các tệp sóng có kích thước lớn hơn so với các định dạng tệp âm thanh mới như MP3 , sử dụng tính năng nén mất dữ liệu để giảm kích thước tệp trong khi vẫn giữ nguyên chất lượng âm thanh. Tuy nhiên, các tệp WAV có thể được nén bằng  Audio Compression Manager (ACM) codecs . Có một số API và ứng dụng có sẵn có thể chuyển đổi tệp WAV sang các định dạng tệp âm thanh phổ biến khác.

File đuôi có dạng .WAV hoặc .WAVE là các file Waveform Audio được phát triển bởi sự hợp tác của Microsoft và IBM vào những năm 1991.

Mặc dù các file WAV thường không được nén nhưng chúng ta vẫn có thể nén chúng nếu muốn, phù hợp để sử dụng trên các hệ thống Windows, đối với MacOS vẫn có thể sử dụngnhưng không phổ biến bằng.

WAV thường được dùng cho công việc chỉnh sửa âm nhạc, trò chơi điện tử, không được dùng trong chia sẻ vì kích thước tệp khá lớn. Về bản chất, WAV khá giống với các tệp AIFF và 8SVX, cả hai đều được sử dụng phổ biến.



WAV là định dạng file chứa âm thanh của Windows

#### ****Mô tả****

WAV chứa các tác phẩm âm nhạc, bản ghi âm giọng nói và các hiệu ứng âm thanh. Có thể phát được trong QuickTime, Windows Media Player và một số phần mềm hỗ trợ khác.

WAV có kích thước lớn hơn MP3, bạn sẽ rất tốn nhiều thời gian cho việc chia sẻ file qua Internet.

Một trong những ưu điểm chính của WAV được liên kết với việc sử dụng Điều chế mã xung tuyến tính (LPCM) để lưu trữ luồng âm thanh.



File WAV thường dùng trong phần mềm chỉnh sửa âm thanh

#### ****Chi tiết kỹ thuật****

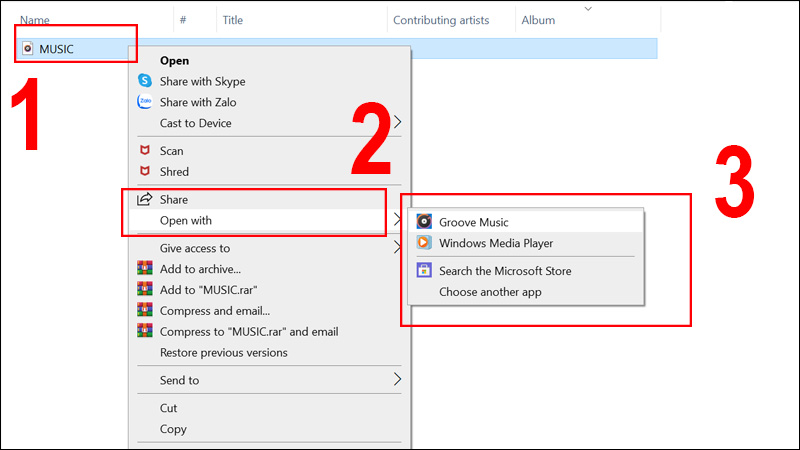
* WAV được ghi thành 2 loại 8 bit và 16 bit cho mỗi mẫu.
* Một tùy chọn tiêu chuẩn cho CD Audio là một dòng âm thanh 16 bit cho mỗi mẫu và tần số lấy mẫu là 44,1 KHz.
* Một giây của âm thanh tương ứng với 88 Kb bộ nhớ trong.
* Các tệp WAV có thể lưu trữ siêu dữ liệu trong đoạn INFO và chúng cũng bao gồm các danh sách IFF được tích hợp.
* Trong một số trường hợp, định dạng chuẩn có thể được sử dụng để phát sóng.



WAV được phát triển bởi Microsoft và IBM

#### ****Cách mở file WAV, WAVE****

Nhấn chuột phải vào file cần mở > **Open with**>Chọn phần mềm bạn muốn dùng là hoàn tất.



Chọn file muốn mở với Open with

#### WAV – Waveform Audio File Format

**Kích thước bit là gì?**  
 Kích thước bit xác định lượng thông tin có thể được lưu trữ trong một tệp. Đối với hầu hết các mục đích ngày nay, kích thước bit phải là 16 bit. Các tệp 8 bit nhỏ hơn (1/2 kích thước), nhưng có độ phân giải thấp hơn.

Kích thước bit liên quan đến biên độ. Trong các bản ghi 8 bit, có sẵn tổng cộng 256 (0 đến 255) mức biên độ. Trong 16 bit, có sẵn tổng cộng 65,536 (-32768 đến 32767) mức biên độ. Độ phân giải của tệp càng lớn thì phạm vi động thực tế của tệp càng lớn. CD-Audio sử dụng các mẫu 16 bit.

**Sample Rate là gì?**  
 Sample rate là số lượng mẫu trong một giây. CD-Audio có tỷ lệ mẫu là 44.100. Điều này có nghĩa là 1 giây âm thanh có 44.100 mẫu. Băng DAT có tỷ lệ mẫu là 48.000.

Khi nhìn vào đáp ứng tần số, tần số cao nhất có thể được coi là 1/2 sample rate.

**Channels là gì?**  
 Channels là số phần tử ghi riêng biệt trong dữ liệu. Đối với một ví dụ nhanh thực tế, channel là đơn âm và hai channels là âm thanh nổi. Trong tài liệu này, cả bản ghi kênh đơn và kênh đôi sẽ được thảo luận.

**Dữ liệu là gì?**  
 Dữ liệu là các mẫu riêng lẻ. Một mẫu riêng lẻ là kích thước bit nhân với số kênh. Ví dụ, một kênh đơn (kênh đơn), ghi tám bit có kích thước mẫu riêng là 8 bit. Bản ghi mười sáu bit đơn nguyên có kích thước mẫu riêng là 16 bit. Bản ghi âm thanh nổi mười sáu bit có kích thước mẫu riêng lẻ là 32 bit.

Các mẫu được đặt từ đầu đến cuối để tạo thành dữ liệu. Vì vậy, ví dụ: nếu bạn có bốn mẫu (s1, s2, s3, s4) thì dữ liệu sẽ giống như sau: s1s2s3s4.

**Tiêu đề là gì?**  
 Tiêu đề là phần đầu của tệp WAV (RIFF). Tiêu đề được sử dụng để cung cấp thông số kỹ thuật về loại tệp, tỷ lệ mẫu, kích thước mẫu và kích thước bit của tệp, cũng như độ dài tổng thể của nó.

* Tiêu đề của tệp WAV (RIFF) dài 44 byte và có định dạng sau:

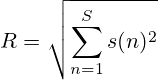
|  |  |  |
| --- | --- | --- |
| Vị trí | Giá trị mẫu | Sự miêu tả |
| 1 - 4 | "RIFF" | Đánh dấu tệp là tệp riff. Mỗi ký tự dài 1 byte. |
| 5 - 8 | File size (integer) | Kích thước của tệp tổng thể - 8 byte, tính bằng byte (số nguyên 32 bit). Thông thường, bạn sẽ điền vào thông tin này sau khi tạo. |
| 9 - 12 | “WAVE” | Tiêu đề Loại Tệp. Đối với mục đích của chúng tôi, nó luôn luôn bằng "WAVE". |
| 13 - 16 | “fmt" | Định dạng điểm đánh dấu chunk. Bao gồm null ở cuối. |
| 17 - 20 | 16 | Độ dài của dữ liệu định dạng như được liệt kê ở trên. |
| 21 - 22 | 1 | Loại định dạng (1 is PCM) - 2 byte integer. |
| 23 - 24 | 2 | Số kênh - 2 byte integer. |
| 25 – 28 | 44100 | Tỷ lệ mẫu - 32 byte integer. Common values là 44100 (CD), 48000 (DAT). Tốc độ lấy mẫu = Số lượng mẫu mỗi giây hoặc Hertz. |
| 29 - 32 | 176400 | (Sample Rate \* BitsPerSample \* Channels) / 8. |
| 33 – 34 | 4 | (BitsPerSample \* Channels) / 8.1 - 8 bit mono2 - 8 bit stereo/16 bit mono4 - 16 bit stereo |
| 35 – 36 | 16 | Số bit trên mỗi mẫu |
| 37 – 40 | “data” | Tiêu đề phân đoạn "data". Đánh dấu phần bắt đầu của phần dữ liệu. |
| 41 – 44 | File size (data) | Kích thước của phần dữ liệu. |
| Các giá trị mẫu được đưa ra ở trên đối với nguồn âm thanh nổi 16 bit. | | |

Có thể tìm thấy ví dụ về nội dung của tiêu đề cho tệp wave

|  |  |  |  |
| --- | --- | --- | --- |
| ****Name**** | ****offset**** | ****Size**** | ****Value**** |
| ChunkID | 0 | 4 | “RIFF” |
| ChunkSize | 4 | 4 | 405040 |
| Format | 8 | 4 | “WAVE” |
| Subchunk1 ID | 12 | 4 | “fmt” |
| Subchunk1 Size | 16 | 4 | 16 |
| Audio Format | 20 | 2 | 1 |
| Num Channels | 22 | 2 | 2 |
| Sample Rate | 24 | 4 | 22050 |
| Byte Rate | 28 | 4 | 88200 |
| Block Align | 32 | 2 | 4 |
| Bits per Sample | 34 | 2 | 16 |
| Subchunk2 ID | 36 | 4 | “data” |
| Subchunk 2 Size | 40 | 4 | 405004 |

#### Waveform draw algorithm

Dữ liệu âm thanh PCM được lưu trữ dưới dạng một chuỗi các mẫu biên độ tín hiệu được ghi lại trong các khoảng thời gian đều đặn. Một giây của âm thanh 8kHz chất lượng thấp bao gồm 8000 mẫu biên độ. Để hiển thị mọi điểm của phân mảnh này, bạn cần ít nhất một màn hình rộng 8000 pixel. Tôi tin rằng độ phân giải màn hình như vậy vẫn còn hiếm vào năm 2021. Do đó, chúng tôi cần một thuật toán để giảm sự biểu diễn trực quan của dạng sóng.  
Máy tính sử dụng thuật toán bình phương trung bình căn (RMS) để biểu diễn tập mẫu dưới dạng một đường đơn trên biểu đồ dạng sóng. Mô tả ngắn gọn về thuật toán vẽ:

* Xác định số điểm chiều rộng đồ thị P (thay đổi bằng công tắc cài đặt trong máy tính)
* Xác định số lượng mẫu mỗi điểm S = T / P, trong đó T - tổng số mẫu
* Đối với mọi điểm, hãy tính RMS:  
    
  trong đó s (n) - là mẫu thứ n của một điểm nhất định
* Đối với mọi điểm, vẽ một đường thẳng đứng từ -R đến R

**Sắc thái định dạng tệp PCM**

Các mẫu biên độ âm thanh được lưu trữ dưới dạng giá trị thực hoặc số nguyên ở định dạng PCM. Máy tính chuyển đổi các biên độ số nguyên thành các biên độ số nguyên trong phạm vi (-1 ... 1) để biểu diễn dạng sóng tín hiệu trên biểu đồ một cách thống nhất. Định dạng PCM có thể lưu trữ hai loại dữ liệu số nguyên. Nếu kích thước mẫu số nguyên nhỏ hơn hoặc bằng 8 bit (một byte), nó được lưu trữ dưới dạng giá trị không dấu. Nếu không (hơn 8 bit), đó là phần bổ sung của hai ký tự. Máy tính sẽ biến đổi dữ liệu số nguyên 8 bit thành float theo cách này: (s (n) -128) / 128. Dữ liệu số nguyên lớn hơn (dài 16, 24 hoặc 32 bit) được chuyển đổi thành float dưới dạng s (n) / | int\_min |. | int\_min | bằng 32768; 8388608 hoặc 2147483648 cho số nguyên 16, 24 hoặc 32 bit tương ứng.

**Chart, histogram

Description automatically generated**

**Chart

Description automatically generated with medium confidence**

### Biến đổi Fourier

**Biến đổi Fourier** hay **chuyển hóa Fourier**, được đặt tên theo nhà toán học người Pháp [Joseph Fourier](https://vi.wikipedia.org/wiki/Joseph_Fourier), là phép [biến đổi](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_t%C3%ADch_ph%C3%A2n) một [hàm số](https://vi.wikipedia.org/wiki/H%C3%A0m_s%E1%BB%91) hoặc một tín hiệu theo miền thời gian sang miền tần số. Chẳng hạn như một bản nhạc có thể được phân tích dựa trên tần số của nó.

#### Ứng dụng

Biến đổi Fourier có rất nhiều ứng dụng [khoa học](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc), ví dụ như trong [vật lý](https://vi.wikipedia.org/wiki/V%E1%BA%ADt_l%C3%BD_h%E1%BB%8Dc), [số học](https://vi.wikipedia.org/wiki/S%E1%BB%91_h%E1%BB%8Dc), [xử lý tín hiệu](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%BD_t%C3%ADn_hi%E1%BB%87u), [xác suất](https://vi.wikipedia.org/wiki/X%C3%A1c_su%E1%BA%A5t), [thống kê](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_Th%E1%BB%91ng_k%C3%AA), [mật mã](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc), [âm học](https://vi.wikipedia.org/wiki/%C3%82m_h%E1%BB%8Dc), [hải dương học](https://vi.wikipedia.org/wiki/H%E1%BA%A3i_d%C6%B0%C6%A1ng_h%E1%BB%8Dc), [quang học](https://vi.wikipedia.org/wiki/Quang_h%E1%BB%8Dc), [hình học](https://vi.wikipedia.org/wiki/H%C3%ACnh_h%E1%BB%8Dc) và rất nhiều lĩnh vực khác. Trong xử lý tín hiệu và các ngành liên quan, biến đổi Fourier thường được nghĩ đến như sự chuyển đổi [tín hiệu](https://vi.wikipedia.org/wiki/T%C3%ADn_hi%E1%BB%87u) thành các thành phần [biên độ](https://vi.wikipedia.org/wiki/Bi%C3%AAn_%C4%91%E1%BB%99) và [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91). Sự ứng dụng rộng rãi của biến đổi Fourier bắt nguồn từ những [tính chất](https://vi.wikipedia.org/w/index.php?title=T%C3%ADnh_ch%E1%BA%A5t&action=edit&redlink=1) hữu dụng của biến đổi này:

* Tính [tuyến tính](https://vi.wikipedia.org/wiki/Tuy%E1%BA%BFn_t%C3%ADnh): {\displaystyle {\mathfrak {F}}[a.f+b.g]=a.{\mathfrak {F}}[f]+b.{\mathfrak {F}}[g]}
* Tồn tại biến đổi nghịch đảo, và thực tế là biến đổi Fourier nghịch đảo gần như có cùng dạng với biến đổi thuận.
* Những hàm số sin cơ sở là các [hàm riêng](https://vi.wikipedia.org/w/index.php?title=H%C3%A0m_ri%C3%AAng&action=edit&redlink=1) của phép [vi phân](https://vi.wikipedia.org/wiki/%C4%90%E1%BA%A1o_h%C3%A0m_v%C3%A0_vi_ph%C3%A2n_c%E1%BB%A7a_h%C3%A0m_s%E1%BB%91), có nghĩa là khai triển này biến những [phương trình vi phân](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_tr%C3%ACnh_vi_ph%C3%A2n) tuyến tính với các hệ số không đổi thành các [phương trình đại số](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_tr%C3%ACnh_%C4%91%E1%BA%A1i_s%E1%BB%91) cơ bản. Ví dụ, trong một hệ vật lý tuyến tính không phụ thuộc thời gian, [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91) là một đại lượng không đổi, do đó những thành phần tần số khác nhau có thể được tính toán một cách độc lập.
* Theo [định lý tích tổng chập](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%8Bnh_l%C3%BD_t%C3%ADch_t%E1%BB%95ng_ch%E1%BA%ADp&action=edit&redlink=1), biến đổi Fourier chuyển một [tích tổng chập](https://vi.wikipedia.org/w/index.php?title=T%C3%ADch_t%E1%BB%95ng_ch%E1%BA%ADp&action=edit&redlink=1) phức tạp thành một tích đại số đơn giản.
* [Biến đổi Fourier rời rạc](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_r%E1%BB%9Di_r%E1%BA%A1c) có thể được tính toán một cách nhanh chóng bằng máy tính nhờ thuật toán [FFT](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_nhanh) (*fast Fourier transform*).
* Theo [định lý Parseval-Plancherel](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%8Bnh_l%C3%BD_Parseval-Plancherel&action=edit&redlink=1), [năng lượng](https://vi.wikipedia.org/wiki/N%C4%83ng_l%C6%B0%E1%BB%A3ng) của [tín hiệu](https://vi.wikipedia.org/wiki/T%C3%ADn_hi%E1%BB%87u) (tích phân của bình phương giá trị tuyệt đối của hàm) không đổi sau biến đổi Fourier.

#### Các dạng của biến đổi Fourier

##### Biến đổi Fourier liên tục

Thông thường, tên gọi biến đổi Fourier[[1]](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier#cite_note-1) được gắn cho [biến đổi Fourier liên tục](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_li%C3%AAn_t%E1%BB%A5c), biến đổi này biểu diễn một hàm bình phương khả tích *f(t)* bất kì theo tổng của các hàm e lũy thừa phức với tần số góc ω và biên độ phức *F*(ω):

{\displaystyle f(t)={\mathcal {F}}^{-1}(F)(t)={\frac {1}{2\pi }}\int \limits \_{-\infty }^{\infty }F(\omega )e^{i\omega t}\,d\omega .} Diagram, schematic

Description automatically generated

Đây là biến đổi nghịch đảo của biến đổi Fourier liên tục, trong khi biến đổi Fourier biểu diễn hàm *F*(ω) theo *f(t)*.

##### Chuỗi Fourier

Biến đổi Fourier liên tục là dạng tổng quát của một khái niệm có từ trước, đó là [chuỗi Fourier](https://vi.wikipedia.org/wiki/Chu%E1%BB%97i_Fourier). Chuỗi Fourier khai triển các [hàm tuần hoàn](https://vi.wikipedia.org/wiki/H%C3%A0m_tu%E1%BA%A7n_ho%C3%A0n) *f(x)* với chu kì 2π (hoặc các hàm có tập xác định bị chặn) theo chuỗi của các hàm sin:

A picture containing schematic

Description automatically generated

trong đó {\displaystyle F\_{n}}*F*n là biên độ phức. Cho các hàm thực, chuỗi Fourier có thể được viết dưới dạng:

Text, letter

Description automatically generated

trong đó *an* và *bn* là các hằng số Fourier (giá trị thực).

##### Biến đổi Fourier rời rạc

Trong [toán học](https://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc), phép **biến đổi Fourier rời rạc (DFT)**, đôi khi còn được gọi là [biến đổi Fourier hữu hạn](https://vi.wikipedia.org/w/index.php?title=Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_h%E1%BB%AFu_h%E1%BA%A1n&action=edit&redlink=1), là một biến đổi trong [giải tích Fourier](https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_t%C3%ADch_Fourier) cho các tín hiệu thời gian rời rạc. Đầu vào của biến đổi này là một chuỗi hữu hạn các [số thực](https://vi.wikipedia.org/wiki/S%E1%BB%91_th%E1%BB%B1c) hoặc [số phức](https://vi.wikipedia.org/wiki/S%E1%BB%91_ph%E1%BB%A9c), làm biến đổi này là một công cụ lý tưởng để xử lý thông tin trên các [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh). Đặc biệt, biến đổi này được sử dụng rộng rãi trong [xử lý tín hiệu](https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%BD_t%C3%ADn_hi%E1%BB%87u) và các ngành liên quan đến phân tích tần số chứa trong một tín hiệu, để giải [phương trình đạo hàm riêng](https://vi.wikipedia.org/wiki/Ph%C6%B0%C6%A1ng_tr%C3%ACnh_vi_ph%C3%A2n_ri%C3%AAng_ph%E1%BA%A7n), và để làm các phép như [tích chập](https://vi.wikipedia.org/wiki/T%C3%ADch_ch%E1%BA%ADp). Biến đổi này có thể được tính nhanh bởi thuật toán [biến đổi Fourier nhanh](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_nhanh) (FFT).

###### Định nghĩa

Dãy của *N* [số phức](https://vi.wikipedia.org/wiki/S%E1%BB%91_ph%E1%BB%A9c):{\displaystyle x\_{0},...,x\_{N-1}} x0, …, xN-1 được biến đổi thành chuỗi của *N* số phức *X*0,..., *XN*−1 bởi công thức sau đây:

Text, letter

Description automatically generated

với *e* là [cơ số của lôgarit tự nhiên](https://vi.wikipedia.org/wiki/S%E1%BB%91_e), {\displaystyle i\,}i là [đơn vị ảo](https://vi.wikipedia.org/wiki/%C4%90%C6%A1n_v%E1%BB%8B_%E1%BA%A3o) ({\displaystyle i^{2}=-1}i2 = -1), và π là [pi](https://vi.wikipedia.org/wiki/Pi). Phép biến đổi đôi khi được ký hiệu bởi {\displaystyle {\mathcal {F}}}

Phép **biến đổi Fourier rời rạc ngược (IDFT)** được cho bởi công thức sau

Text, letter

Description automatically generated

Những phương trình này có thể được mô tả đơn giản như sau: các số phức *Xk* đại diện cho biên độ và pha ở các bước sóng khác nhau của "tín hiệu vào" *xn*. Phép biến đổi DFT tính các giá trị *Xk* từ các giá trị *xn*, trong khi IDFT tính *xn* bằng tổng của các sóng thành phần {\displaystyle {\frac {1}{N}}X\_{k}e^{{\frac {2\pi i}{N}}kn}} Text

Description automatically generated với [tần số](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91) *k / N*. Khi viết các phương trình dưới dạng như trên, ta đã sử dụng [công thức Euler](https://vi.wikipedia.org/wiki/C%C3%B4ng_th%E1%BB%A9c_Euler) để biểu diễn các hàm lượng giác dưới dạng lũy thừa số phức để biến đổi được dễ dàng. Khi viết *Xk* dưới dạng [tọa độ cực](https://vi.wikipedia.org/wiki/H%E1%BB%87_t%E1%BB%8Da_%C4%91%E1%BB%99_c%E1%BB%B1c), ta thu được biên độ *Ak / N* và pha *φk* từ modulus và argument của *Xk*:

Text, letter

Description automatically generated

trong đó [atan2](https://vi.wikipedia.org/w/index.php?title=Atan2&action=edit&redlink=1) là dạng hai đối số của hàm [arctan](https://vi.wikipedia.org/w/index.php?title=Arctan&action=edit&redlink=1). Cần ghi chú rằng các thừa số chuẩn hóa của DFT và IDFT (ở đây là 1 và 1/*N*) và dấu của các số mũ chỉ là quy ước, và có thể khác nhau trong các tài liệu khác nhau. Điều kiện duy nhất cho các quy ước này là DFT và IDFT có dấu ngược nhau ở các số mũ và tích của hai thừa số chuẩn hóa phải là 1/*N*.

###### Ứng dụng

DFT có nhiều ứng dụng rộng rãi trong nhiều ngành khác nhau. Tất cả các ứng dụng của DFT đều dựa trên một tính chất quan trọng là DFT và IDFT đều có thể được tính nhanh chóng bằng thuật toán [biến đổi Fourier nhanh](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_nhanh).

**Phân tích phổ**

Khi sử dụng DFT để [phân tích phổ](https://vi.wikipedia.org/w/index.php?title=Ph%E1%BB%95_t%E1%BA%A7n_s%E1%BB%91&action=edit&redlink=1), dãy {*x\_n*} thường đại diện cho một dãy hữu hạn các mẫu tại các thời điểm cách đều nhau của một tín hiệu *x(t)*, trong đó *t* để chỉ thời gian. Việc chuyển từ thời gian liên tục sang mẫu (thời gian rời rạc) chuyển [biến đổi Fourier liên tục](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_li%C3%AAn_t%E1%BB%A5c) của *x(t)* thành [biến đổi Fourier thời gian rời rạc](https://vi.wikipedia.org/w/index.php?title=Bi%E1%BA%BFn_%C4%91%E1%BB%95i_Fourier_th%E1%BB%9Di_gian_r%E1%BB%9Di_r%E1%BA%A1c&action=edit&redlink=1) (DTFT), và thường gây ra [hiệu ứng răng cưa](https://vi.wikipedia.org/w/index.php?title=Hi%E1%BB%87u_%E1%BB%A9ng_r%C4%83ng_c%C6%B0a&action=edit&redlink=1). Việc chọn lựa tần số lấy mẫu thích hợp (xem [tần số Nyquist](https://vi.wikipedia.org/wiki/T%E1%BA%A7n_s%E1%BB%91_Nyquist)) là vô cùng quan trọng cho việc giảm thiểu hiệu ứng này.

## CÔNG CỤ VÀ NGÔN NGỮ LẬP TRÌNH ĐƯỢC SỬ DỤNG

### Tổng quan về VISUAL STUDIO 2019

* **Microsoft Visual Studio** là một [môi trường phát triển tích hợp](https://vi.wikipedia.org/wiki/M%C3%B4i_tr%C6%B0%E1%BB%9Dng_ph%C3%A1t_tri%E1%BB%83n_t%C3%ADch_h%E1%BB%A3p) (IDE) từ [Microsoft](https://vi.wikipedia.org/wiki/Microsoft). Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới ", được dùng để lập trình [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [C#](https://vi.wikipedia.org/wiki/C) là chính. Nó được sử dụng để phát triển [chương trình máy tính](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh) cho [Microsoft Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), cũng như các trang web, các ứng dụng web và các [dịch vụ web](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_web). Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như [Windows API](https://vi.wikipedia.org/wiki/Windows_API), [Windows Forms](https://vi.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://vi.wikipedia.org/wiki/Windows_Store) và [Microsoft Silverlight](https://vi.wikipedia.org/wiki/Microsoft_Silverlight). Nó có thể sản xuất cả hai [ngôn ngữ máy](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%A1y) và [mã số quản lý](https://vi.wikipedia.org/w/index.php?title=M%C3%A3_s%E1%BB%91_qu%E1%BA%A3n_l%C3%BD&action=edit&redlink=1).
* **Ưu điểm:**
* Đa nền tảng.
* Đa ngôn ngữ lập trình
* Hỗ trợ website.
* Kho tiện ích mở rộng phong phú.
* Lưu trữ phân cấp.
* Kho lưu trữ an toàn.
* …
* **Hạn chế:**
* Tuy có giao diện đẹp, nhưng khá khó sử dụng cho người mới học lập trình. Vì khá nhiều bước thiết lập và cài đặt.
* Vì Visual Studio là một IDE nặng nên cần sử dụng nhiều tài nguyên để khởi động và vận hành.
* …

### Ngôn ngữ C++

C++ là một loại ngôn ngữ lập trình bậc trung (middle-level). Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại ưa thích và phổ biến của lập trình viên.

* **Ưu điểm:**
* Là ngôn ngữ lập trình hướng đối tượng.
* Cho phép thoải mái quản lý vùng nhớ
* Là một ngôn ngữ low-level dễ dàng giao tiếp với phần cứng.
* …
* **Hạn chế:**
* Tràn vùng nhớ.
* OOP trong C++ khá phức tạp và khó hiểu.
* Con trỏ là một cái gì đó rất … khó.
* …

### Ngôn ngữ C#

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

**C#** (**C Sharp**, đọc là *"xi-sáp"*) là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) đa năng, mạnh mẽ được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), C# là phần khởi đầu cho kế hoạch [.NET](https://vi.wikipedia.org/wiki/.NET_Framework) của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo [ECMA](https://vi.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1) là **C#**, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

* **Ưu điểm:**
* C Sharp là ngôn ngữ đơn giản.
* Ngôn ngữ đa nền tảng, an toàn và hiệu quả.
* Là ngôn ngữ hiện đại thông dụng, ít từ khóa dễ hiểu.
* Là ngôn ngữ lập trình thuần hướng đối tượng
* …
* **Hạn chế:**
* Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows.
* …

# Đặc tả bài toán

## BÀI TOÁN

Vận dụng kiến thức lập trình C, C#. Biết làm việc với file nhị phân. Tìm hiểu biến đổi Fourier để phân tích phổ

* 1. **Mức 1:** Đọc và thể hiện các thông tin header của file wav, hiển thị dạng sóng
  2. **Mức 2:** Phân tích phổ và hiển thị

## Hướng xử lý và các thuật toán để thực hiện các yêu cầu:

### Mức 1: Đọc và thể hiện các thông tin header của file wav, hiển thị dạng sóng

#### Đọc thông tin header file wav

Ý tưởng:

* B1: Mở file wav cần đọc thông tin.
* B2: Hiển thị các thông tin header của file wav để đọc.
* File is

cout << "File is:" << filelength << " bytes." << endl;

* RIFF header

cout << "RIFF header:" << wavHeader.RIFF[0] << wavHeader.RIFF[1] << wavHeader.RIFF[2] << wavHeader.RIFF[3] << endl;

* WAVE header

cout << "WAVE header:" << wavHeader.WAVE[0] << wavHeader.WAVE[1] << wavHeader.WAVE[2] << wavHeader.WAVE[3] << endl;

* FMT

cout << "FMT:" << wavHeader.fmt[0] << wavHeader.fmt[1] << wavHeader.fmt[2] << wavHeader.fmt[3] << endl;

* Data size

cout << "Data size:" << wavHeader.ChunkSize << endl;

// Display the sampling Rate from the header

* Sampling Rate

cout << "Sampling Rate:" << wavHeader.SamplesPerSec << endl;

* Number of bits used

cout << "Number of bits used:" << wavHeader.bitsPerSample << endl;

* Number of channels

cout << "Number of channels:" << wavHeader.NumOfChan << endl;

* Number of bytes per second

cout << "Number of bytes per second:" << wavHeader.bytesPerSec << endl;

* Data length

cout << "Data length:" << wavHeader.Subchunk2Size << endl;

* Audio Format

cout << "Audio Format:" << wavHeader.AudioFormat << endl;

// Audio format 1=PCM,6=mulaw,7=alaw, 257=IBM Mu-Law, 258=IBM A-Law, 259=ADPCM

* Block align

cout << "Block align:" << wavHeader.blockAlign << endl;

* Data string

cout << "Data string:" << wavHeader.Subchunk2ID[0] << wavHeader.Subchunk2ID[1] << wavHeader.Subchunk2ID[2] << wavHeader.Subchunk2ID[3] << endl;

#### Hiển thị dạng sóng

# CÀI ĐẶT CHƯƠNG TRÌNH

## Code chương trình

### Mức 1

#### Đọc thông tin header file wav

##### **Code**

|  |
| --- |
| *#define \_CRT\_SECURE\_NO\_DEPRECATE*  *#include <iostream>*  *#include <string>*  *#include <fstream>*  *#include <cstdint>*  *using std::cin;*  *using std::cout;*  *using std::endl;*  *using std::fstream;*  *using std::string;*  *typedef struct WAV\_HEADER*  *{*  */\* RIFF Chunk Descriptor \*/*  *uint8\_t RIFF[4]; // RIFF Header Magic header*  *uint32\_t ChunkSize; // RIFF Chunk Size*  *uint8\_t WAVE[4]; // WAVE Header*  */\* "fmt" sub-chunk \*/*  *uint8\_t fmt[4]; // FMT header*  *uint32\_t Subchunk1Size; // Size of the fmt chunk*  *uint16\_t AudioFormat; // Audio format 1=PCM,6=mulaw,7=alaw, 257=IBM Mu-Law, 258=IBM A-Law, 259=ADPCM*  *uint16\_t NumOfChan; // Number of channels 1=Mono 2=Sterio*  *uint32\_t SamplesPerSec; // Sampling Frequency in Hz*  *uint32\_t bytesPerSec; // bytes per second*  *uint16\_t blockAlign; // 2=16-bit mono, 4=16-bit stereo*  *uint16\_t bitsPerSample; // Number of bits per sample*  */\* "data" sub-chunk \*/*  *uint8\_t Subchunk2ID[4]; // "data" string*  *uint32\_t Subchunk2Size; // Sampled data length*  *} wav\_hdr;*  *// Function prototypes*  *int getFileSize(FILE\* inFile);*  *int main(int argc, char\* argv[])*  *{*  *wav\_hdr wavHeader;*  *int headerSize = sizeof(wav\_hdr), filelength = 0;*  *const char\* filePath;*  *string input;*  *if (argc <= 1)*  *{*  *cout << "Input wave file name: ";*  *cin >> input;*  *cin.get();*  *filePath = input.c\_str();*  *}*  *else*  *{*  *filePath = argv[1];*  *cout << "Input wave file name: " << filePath << endl;*  *}*  *FILE\* wavFile = fopen(filePath, "r");*  *if (wavFile == nullptr)*  *{*  *fprintf(stderr, "Unable to open wave file: %s\n", filePath);*  *return 1;*  *}*  *//Read the header*  *size\_t bytesRead = fread(&wavHeader, 1, headerSize, wavFile);*  *cout << "Header Read " << bytesRead << " bytes." << endl;*  *if (bytesRead > 0)*  *{*  *//Read the data*  *uint16\_t bytesPerSample = wavHeader.bitsPerSample / 8; //Number of bytes per sample*  *//uint64\_t numSamples = wavHeader.ChunkSize / bytesPerSample; //How many samples are in the wav file?*  *static const uint16\_t BUFFER\_SIZE = 4096;*  *int8\_t\* buffer = new int8\_t[BUFFER\_SIZE];*  *while ((bytesRead = fread(buffer, sizeof buffer[0], BUFFER\_SIZE / (sizeof buffer[0]), wavFile)) > 0)*  *{*  */\*\* DO SOMETHING WITH THE WAVE DATA HERE \*\*/*  *cout << "Read " << bytesRead << " bytes." << endl;*  *}*  *delete[] buffer;*  *buffer = nullptr;*  *filelength = getFileSize(wavFile);*  *cout << "File is :" << filelength << " bytes." << endl;*  *cout << "RIFF header :" << wavHeader.RIFF[0] << wavHeader.RIFF[1] << wavHeader.RIFF[2] << wavHeader.RIFF[3] << endl;*  *cout << "WAVE header :" << wavHeader.WAVE[0] << wavHeader.WAVE[1] << wavHeader.WAVE[2] << wavHeader.WAVE[3] << endl;*  *cout << "FMT :" << wavHeader.fmt[0] << wavHeader.fmt[1] << wavHeader.fmt[2] << wavHeader.fmt[3] << endl;*  *cout << "Data size :" << wavHeader.ChunkSize << endl;*  *// Display the sampling Rate from the header*  *cout << "Sampling Rate :" << wavHeader.SamplesPerSec << endl;*  *cout << "Number of bits used :" << wavHeader.bitsPerSample << endl;*  *cout << "Number of channels :" << wavHeader.NumOfChan << endl;*  *cout << "Number of bytes per second :" << wavHeader.bytesPerSec << endl;*  *cout << "Data length :" << wavHeader.Subchunk2Size << endl;*  *cout << "Audio Format :" << wavHeader.AudioFormat << endl;*  *// Audio format 1=PCM,6=mulaw,7=alaw, 257=IBM Mu-Law, 258=IBM A-Law, 259=ADPCM*  *cout << "Block align :" << wavHeader.blockAlign << endl;*  *cout << "Data string :" << wavHeader.Subchunk2ID[0] << wavHeader.Subchunk2ID[1] << wavHeader.Subchunk2ID[2] << wavHeader.Subchunk2ID[3] << endl;*  *}*  *fclose(wavFile);*  *return 0;*  *}*  *// find the file size*  *int getFileSize(FILE\* inFile)*  *{*  *int fileSize = 0;*  *fseek(inFile, 0, SEEK\_END);*  *fileSize = ftell(inFile);*  *fseek(inFile, 0, SEEK\_SET);*  *return fileSize;*  *}* |

##### **Kết quả chạy thử chương trình**

**Text

Description automatically generated**

#### Hiển thị dạng sóng

##### **Code**

|  |
| --- |
| *using System;*  *using System.Collections.Generic;*  *using System.ComponentModel;*  *using System.Data;*  *using System.Drawing;*  *using System.Linq;*  *using System.Text;*  *using System.Threading.Tasks;*  *using System.Windows.Forms;*  *namespace Plotting\_Audio\_Waveforms*  *{*  *public partial class Form1 : Form*  *{*  *public Form1()*  *{*  *InitializeComponent();*  *}*  *private void openWaveToolStripMenuItem\_Click(object sender, EventArgs e)*  *{*  *OpenFileDialog open = new OpenFileDialog();*  *open.Filter = "Wave File (\*.wav)|\*.wav;";*  *if (open.ShowDialog() != DialogResult.OK) return;*  *chart1.Series.Add("wave");*  *chart1.Series["wave"].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.FastLine;*  *chart1.Series["wave"].ChartArea = "ChartArea1";*  *NAudio.Wave.WaveChannel32 wave = new NAudio.Wave.WaveChannel32(new NAudio.Wave.WaveFileReader(open.FileName));*  *byte[] buffer = new byte[16384];*  *int read = 0;*  *while (wave.Position < wave.Length)*  *{*  *read = wave.Read(buffer, 0, 16384);*  *for (int i = 0; i < read / 4; i++)*  *{*  *chart1.Series["wave"].Points.Add(BitConverter.ToSingle(buffer, i \* 4));*  *}*  *}*  *}*  *}*  *}* |

##### **Kết quả chạy thử chương trình**

**Chart, histogram

Description automatically generated**

##### **Kết quả chạy file wav trên phần mềm Audacity**

**Chart

Description automatically generated with medium confidence**

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## KẾT QUẢ ĐẠT ĐƯỢC

* + Đọc được thông tin header của file wav, hiển thị dạng sóng.
  + Hiển thị phổ âm thanh.

## ƯU ĐIỂM

* + Chương trình dễ sử dụng, giao diện thân thiện.
  + Bài toán hiển thị phổ âm thanh có thể làm tiền đề để giải quyết các bài toán khó hơn, phức tạp hơn

## HẠN CHẾ

* Giao diện hệ thống chưa được tối ưu cho trải nghiệm người dùng.
* Code chương trình còn dài, còn nhiều chỗ chưa được tối ưu.

## HƯỚNG PHÁT TRIỂN

* Hoàn thiện thêm những chức năng hệ thống hiện có, cải thiện về giao diện, trải nghiệm người.

## KẾT LUẬN

Ngày nay, cuộc cách mạng công nghiệp 4.0 và vai trò của Internet ngày càng được khẳng định và không thể thiếu trong quản lý và chia sẻ thông tin của một quốc gia hay địa phương. Cùng với sự phát triển mạnh mẽ của hệ thống mạng toàn cầu – Internet và nhu cầu chia sẻ, tra cứu thông tin trên Internet.

Định dạng tệp WAV không nén, nên nó hoạt động tốt hơn về chất lượng âm thanh và tốc độ bit, không giới hạn về độ sâu bit.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Wikipedia tiếng việt, "C Sharp (ngôn ngữ lập trình)," 07 01 2021. [Online]. Available: https://vi.wikipedia.org/wiki/C\_Sharp\_(ng%C3%B4n\_ng%E1%BB%AF\_l%E1%BA%ADp\_  tr%C3%ACnh). [Accessed 07 01 2021]. |
| [2] | Wikipedia tiếng việt, “Biến đổi Fourier,” 7 1 2022. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn\_%C4%91%E1%BB%95i\_Fourier. [Đã truy  cập 7 1 2022]. |
| [3] | Wikipedia tiếng việt, "Biến đổi Fourier rời rạc," 7 1 2022. [Online]. Available: https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn\_%C4%91%E1%BB%95i\_Fourier\_r%E1%  BB%9Di\_r%E1%BA%A1c. [Accessed 7 1 2022]. |
| [4] | Wikipedia, “Microsoft Visual Studio,” 7 1 2022. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Microsoft\_Visual\_Studio. [Đã truy cập 7 1 2022]. |
| [5] | Wikipedia tiếng việt, "C++," 7 1 2022. [Online]. Available: https://vi.wikipedia.org/wiki/C%2B%2B. [Accessed 7 1 2022]. |
| [6] | GiawaVideos, "C# Audio Tutorial 10 - Plotting Audio Waveforms," [Online]. Available: https://www.youtube.com/watch?v=ZnFoVuOVrUQ. |
| [7] | S. Harden, “Realtime Microphone Audio FFT Graph with C#,” [Trực tuyến]. Available: https://www.youtube.com/watch?v=q9cRZuosrOs. |
| [8] | FILEFORMAT, "WAV - Waveform Audio File Format," [Online]. Available: https://docs.fileformat.com/audio/wav/. |
| [9] | stackoverflow, "C++ Reading the Data part of a WAV file," [Online]. Available: https://stackoverflow.com/questions/13660777/c-reading-the-data-part-of-a-wav-file. |
| [10] | O. S. Code, "Reading the header of a wav file - One Step! Code," [Online]. Available: https://onestepcode.com/read-wav-header/. |
| [11] | PLANETCALC, "Audio file waveform - PLANETCALC Online calculators," [Online]. Available: https://planetcalc.com/8627/. |