

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG



HCMUTE

ĐỒ ÁN 1

ĐỀ TÀI: THIẾT KẾ HỆ THỐNG TODO – LIST

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

SINH VIÊN

Nguyễn Võ Phương Thảo

20119029

Lê Phú Cường

20119203

TP. HỒ CHÍ MINH 6/2023

TRANG THÔNG TIN BÁO CÁO

1. Thông tin sinh viên

Họ và tên sinh viên 1: NGUYỄN VÕ PHƯƠNG THẢO

MSSV: 20119029

Email: 20119029@student.hcmute.edu.vn

Họ và tên sinh viên 2: LÊ PHÚ CƯỜNG

MSSV: 20119203

Email: 20119203@student.hcmute.edu.vn

2. Thông tin đề tài

Tên đề tài: THIẾT KẾ HỆ THỐNG TODO – LIST.

Đơn vị quản lý: Bộ môn Kỹ thuật máy tính – Viễn thông, Khoa Đào tạo Chất lượng cao

Tử: Trường Đại học Sư phạm Kỹ Thuật Tp. Hồ Chí Minh.

Thời gian thực hiện: Từ ngày 6/3/2023 đến ngày 6/6/2023.

3. Lời cam đoan

Nhóm em: Nguyễn Võ Phương Thảo và Lê Phú Cường cam đoan đề tài này là do chính chúng em nghiên cứu và thực hiện. Chúng em không sao chép bất kì nguồn tài liệu được công khai mà không trích dẫn nguồn gốc. Nếu có vi phạm, chúng em xin hoàn toàn chịu trách nhiệm.

Thành phố Hồ Chí Minh, ngày 6 tháng 6 năm 2023

NHIỆM VỤ ĐỒ ÁN

Họ và tên sinh viên 1:	Nguyễn Võ Phương Thảo	MSSV: 20119029
Chuyên ngành:	Kỹ thuật máy tính	Mã ngành: 119
Hệ đào tạo	Đại học chính quy	
Khoá	K20	Lớp: 20119CL2A
Họ và tên sinh viên 1:	Lê Phú Cường	MSSV: 20119203
Chuyên ngành:	Kỹ thuật máy tính	Mã ngành: 119
Hệ đào tạo	Đại học chính quy	
Khoá	K20	Lớp: 20119CL2A

TÊN ĐỀ TÀI: THIẾT KẾ HỆ THỐNG TODO – LIST

1. NHIỆM VỤ:

- Các số liệu ban đầu:

Sử dụng module INMP441 để thực hiện chức năng ghi âm.

Sử dụng module MAX98357 để mã hoá và khuếch đại tín hiệu âm thanh.

Module ILI9341 có chức năng TOUCH dùng để thao tác bằng cảm ứng trên màn hình.

Sử dụng module Encoder để thực hiện chức năng tăng – giảm âm lượng phát ra từ loa.

Module MicroSD Adapter dùng để đọc dữ liệu trong thẻ nhớ, kết hợp với module INMP441 để đưa file ghi âm vào trong thẻ nhớ và phát lại thông qua module MAX98357.

- Nội dung thực hiện:

Nghiên cứu về ESP32, các module INMP441, MAX98357, ILI9341, Encoder, MicroSD Adapter.

Tìm hiểu và cài đặt phần mềm lập trình cho board vi xử lý là Arduino IDE.

Tạo một 1 trang web để có thể nhập thông báo và hiển thị lên màn hình ILI9341.

Test các module riêng lẻ để đảm bảo nó thực hiện đúng chức năng.

Kết nối các module tạo thành 1 hệ thống hoàn chỉnh.

Thực hiện thiết kế và tiến hành thi công mạch in, sau đó lắp ráp các linh kiện phân cứng.

Cân chỉnh hệ thống.

Viết báo cáo.

2. NGÀY GIAO NHIỆM VỤ:

3. NGÀY HOÀN THÀNH NHIỆM VỤ:

4. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. Huỳnh Hoàng Hà

Cán bộ hướng dẫn

BM. Kỹ thuật máy tính – Viễn thông

ThS. Huỳnh Hoàng Hà

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN

TÊN ĐỀ TÀI: THIẾT KẾ HỆ THỐNG TODO – LIST

Tuần – ngày	Nội dung	Xác nhận GVHD
Tuần 1 (13/3 – 19/3)	<ul style="list-style-type: none"> Gặp GVHD để nghe phổ biến, yêu cầu làm đồ án Chọn đề tài 	
Tuần 2 (20/3 – 26/3)	<ul style="list-style-type: none"> Tìm hiểu board ESP32, các module INMP441, MAX98357, ILI9341, Encoder, MicroSD Adapter. Chọn và mua linh kiện 	
Tuần 3 (27/3 – 2/4)	Tìm hiểu code và test từng module lẻ	
Tuần 4 (3/4 – 9/4)	Thiết kế mô hình mạch tổng quát	
Tuần 5 (10/4 – 16/4)	Nghiên cứu và xây dựng code cho toàn hệ thống.	
Tuần 6 (17/4 – 23/4)	Thiết kế giao diện web	
Tuần 7 (24/4 – 30/4)	Tiến hành chạy mạch thực tế và khắc phục lỗi	
Tuần 8 (1/5 – 7/5)	<ul style="list-style-type: none"> Làm file Word và nộp báo cáo Làm mạch in. 	
Tuần 9 (8/5 – 14/5)		
Tuần 10 (15/5 – 21/5)	Vẽ hộp 3D.	
Tuần 11 (22/5 – 28/5)	Tuần dự trữ.	

LỜI CẢM ƠN

Đầu tiên, nhóm em xin gửi lời cảm ơn sâu sắc đến Thầy Huỳnh Hoàng Hà – Giảng viên bộ môn Kỹ thuật máy tính đã trực tiếp hướng dẫn và tận tình giúp đỡ, tạo điều kiện để chúng em hoàn thành tốt đề tài của đồ án 1.

Vì thời gian và lượng kiến thức có hạn nên việc thực hiện đồ án không thể tránh khỏi việc có một số sai sót, mong thầy cô thông cảm.

Nhóm chúng em xin chân thành cảm ơn!

Người thực hiện đề tài

Nguyễn Võ Phương Thảo Lê Phú Cường

TÓM TẮT ĐỀ TÀI

Đề tài “Thiết kế hệ thống TODO - LIST” có các thành phần chính như sau:

- Board ESP32: Đây là bo mạch phát triển dựa trên chip ESP32, có khả năng kết nối Wi-Fi và Bluetooth. Nó được sử dụng làm nền tảng chính cho hệ thống.
- Màn hình ILI9341: Một màn hình TFT màu với giao diện SPI, được sử dụng để hiển thị thông tin và tương tác với người dùng. Màn hình này có tính năng cảm ứng, cho phép sử dụng chạm để điều khiển các chức năng.
- Module ghi âm INMP441: Module ghi âm tích hợp micro. Nó cho phép ghi âm âm thanh từ môi trường xung quanh.
- Bộ mã hoá âm thanh MAX98357: Một bộ khuếch đại âm thanh mono Class D, được sử dụng để mã hoá âm thanh ghi âm và phát lại thông qua loa. Nó cung cấp công suất đầu ra và chất lượng âm thanh tốt.
- Rotary Encoder: Một bộ mã hóa quay được sử dụng để điều chỉnh âm lượng của hệ thống. Bằng cách xoay encoder, người dùng có thể tăng hoặc giảm âm lượng của âm thanh được phát ra.
- MicroSD Adapter: Một module mở rộng được sử dụng để kết nối và đọc/ghi dữ liệu từ thẻ nhớ microSD. Nó được sử dụng để lưu trữ các file ghi âm và cho phép đọc dữ liệu từ thẻ nhớ.

Hệ thống này hoạt động bằng cách sử dụng ESP32 để điều khiển các module và linh kiện khác. Đầu tiên, màn hình ILI9341 được setup có 2 option: Ghi âm (Recording) và nhập thông báo – lịch trình (Scheduling/Annouce). Khi người dùng chạm vào màn hình trong toạ độ của nút ghi âm, module INMP441 sẽ bắt đầu ghi âm âm thanh từ micro và mã hoá dữ liệu âm thanh được thu âm bằng MAX98357. Dữ liệu ghi âm sẽ được lưu trữ trên thẻ nhớ microSD thông qua MicroSD Adapter. Người dùng có thể sử dụng rotary encoder để điều chỉnh âm lượng của âm thanh được phát ra qua loa. Và khi người dùng nhấn vào màn hình trong toạ độ của chức năng nhập thông báo – lịch trình, màn hình sẽ chuyển đổi qua giao diện bảng thông báo bao gồm thời gian và bảng thông báo. Lúc này, chúng ta lên web để nhập lịch trình vào và nó sẽ hiển thị trên màn hình.

Mục đích của đề tài là để nhắc nhở lịch trình trong 1 ngày, sử dụng giọng nói đã ghi âm từ trước làm âm thanh phát ra thông báo. Ngoài ra, hệ thống còn có thể phát triển thành máy ghi âm hoặc máy phát nhạc.

MỤC LỤC

DANH MỤC HÌNH	1
DANH MỤC BẢNG	2
CHƯƠNG 1: TỔNG QUAN	3
1.1. Đặt vấn đề	3
1.2. Mục tiêu.....	3
1.3. Nội dung nghiên cứu	3
1.4. Giới hạn đề tài.....	4
1.5. Bố cục.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	5
2.1. Tổng quan về Google Firebase	5
2.1.1. Giới thiệu về Google Firebase	5
2.1.2. Chức năng của Google Firebase	5
2.1.3. Ưu điểm và nhược điểm của Google Firebase	6
2.2. Giới thiệu về giao thức truyền thông được sử dụng trong hệ thống.....	7
2.2.1. SPI.....	7
2.2.2. I2S.....	8
2.2.3. I2C.....	9
2.2.4. SDIO.....	9
2.2.5. UART.....	10
2.3. Giới thiệu về linh kiện sử dụng trong hệ thống.....	11
2.3.1. Giới thiệu về ESP32	11
2.3.2. Giới thiệu về module MAX98357.....	12
2.3.3. Giới thiệu về module INMP441	14
2.3.4. Giới thiệu về module Rotary Encoder.....	15
2.3.5. Giới thiệu về module ILI9341	16
2.3.6. Giới thiệu về module MicroSD Adapter.....	18
CHƯƠNG 3: THIẾT KẾ VÀ THỰC THI HỆ THỐNG	20
3.1. Giới thiệu hệ thống.....	20
3.1.1. Giới thiệu và tóm tắt hệ thống.....	20
3.1.2. Yêu cầu và chức năng	20
3.1.3. Sơ đồ tổng quát của hệ thống.....	21

3.2. Sơ đồ và chức năng của các khối trong hệ thống	22
3.2.1. Sơ đồ khối.....	22
3.2.2. Chức năng từng khối.....	22
3.3. Thiết kế chi tiết.....	23
3.3.1. Sơ đồ nguyên lý	23
3.3.2. Thiết kế từng khối.....	23
3.4. Thiết kế và thi công mạch.....	26
3.4.1. Thiết kế PCB.....	26
3.4.2. Thi công mạch.....	27
3.5. Thiết kế hộp cho hệ thống	29
CHƯƠNG 4: KIỂM TRA VÀ ĐÁNH GIÁ.....	30
4.1. Kết quả quá trình thực hiện đề tài	30
4.1.1. Kết quả đạt được	30
4.1.2. Kết quả mô hình module thực tế	30
4.1.3. Kết quả gửi dữ liệu lên Google Firebase.....	31
4.2. Nhận xét và đánh giá chung.....	31
CHƯƠNG 5: HƯỚNG PHÁT TRIỂN	33
TÀI LIỆU THAM KHẢO	34

DANH MỤC HÌNH

Hình 2.1. Giao thức SPI

Hình 2.2. Sơ đồ thời gian của I2S

Hình 2.3. Giao thức I2C

Hình 2.4. ESP32 DevKitC và DOIT ESP32 DEVKIT V1

Hình 2.5. Sơ đồ chân của ESP32 DevKit V1 – DOIT và ESP32 DevKitC

Hình 2.6. Ảnh thực tế của module MAX98357

Hình 2.7. Ảnh thực tế và sơ đồ chân của module INMP441

Hình 2.8. Ảnh thực tế của module Rotary Encoder

Hình 2.9. Ảnh thực tế của module ILI9341

Hình 2.10. Sơ đồ chân của ILI9341

Hình 2.11. Ảnh thực tế của Module MicroSD Adapter

Hình 3.1. Sơ đồ tổng quát của hệ thống

Hình 3.2. Sơ đồ khối của hệ thống

Hình 3.3. Sơ đồ nguyên lý của mạch

Hình 3.4. Sơ đồ khối nguồn

Hình 3.5. Mạch in

Hình 3.6. Mặt trên của PCB

Hình 3.7. Mặt dưới của PCB

Hình 3.8. Bản mạch full linh kiện (mặt trên)

Hình 3.9. Bản mạch full linh kiện (mặt dưới)

Hình 3.10. Phần thứ nhất của hộp

Hình 3.11. Phần thứ hai của hộp

Hình 4.1. Module thực tế

Hình 4.2. Giao diện web để cập nhật lịch trình và thiết lập thời gian

Hình 4.3. Kết quả hiển thị trên Google Firebase

DANH MỤC BẢNG

Bảng 2.4. Thông số kỹ thuật của module Rotary Encoder

Bảng 2.5. Thông số kỹ thuật của ILI9341

Bảng 2.6. Thông số kỹ thuật của MicroSD Adapter

Bảng 3.1. Bảng nối chân từ ESP32 tới LCD TFT

Bảng 3.2. Bảng nối chân từ ESP32 tới module MAX98357

Bảng 3.3. Bảng nối chân từ ESP32 tới module INMP441

CHƯƠNG 1: TỔNG QUAN

1.1. Đặt vấn đề

Âm thanh và quản lý thời gian là hai yếu tố quan trọng trong cuộc sống hàng ngày của chúng ta. Từ việc kiểm soát âm lượng, ghi âm thông tin quan trọng đến quản lý lịch trình và công việc, chúng ta luôn cần có những công cụ và giải pháp tiện ích để tạo ra trải nghiệm tốt hơn và tăng hiệu suất làm việc, cải thiện chất lượng đời sống.

Vì lí do đó, nhóm chúng em đã lên ý tưởng thực hiện đề tài “Thiết kế hệ thống Todo – List” nhằm cho người dùng có thể ghi âm và thêm lời nhắc bằng file ghi âm đó, sau đó có thể phát ra đúng thời gian lịch trình được thiết lập sẵn trên thời gian biểu, nhờ vậy người dùng có thể quản lí được quỹ thời gian hợp lý hơn. Với sự kết hợp của các module và vi điều khiển, đề tài này mong muốn mang đến một giải pháp toàn diện và tiện ích cho việc điều khiển âm thanh, ghi âm và quản lý lịch trình. Chúng em hy vọng rằng hệ thống sẽ đáp ứng được nhu cầu và mang lại trải nghiệm tốt cho người dùng.

1.2. Mục tiêu

Mục tiêu của đề tài “Thiết kế hệ thống TODO – LIST”:

- Phát triển một hệ thống có khả năng điều khiển âm lượng, ghi âm âm thanh và phát lại âm thanh thông qua giao diện người dùng.
- Cho phép người dùng điều chỉnh âm lượng âm thanh bằng cách sử dụng Rotary Encoder, ghi âm âm thanh thông qua cảm biến INMP441 và lưu trữ các tệp ghi âm vào thẻ nhớ MicroSD. Sau đó, người dùng có thể phát lại âm thanh đã ghi thông qua loa và màn hình cảm ứng.
- Thông qua giao diện giao tiếp, người dùng có thể điền lịch trình vào bảng thông báo và hẹn thời gian phát file âm thanh đã ghi từ trước như một lời nhắc nhở.

1.3. Nội dung nghiên cứu

- Nội dung 1: Tổng quan về hệ thống ghi âm và thao tác người dùng
- Nội dung 2: Tổng quan về các giao thức truyền thông
- Nội dung 3: Hoàn thiện hệ thống
- Nội dung 4: Kết quả thực nghiệm

1.4. Giới hạn đề tài

Kiến thức và kỹ năng: Để triển khai đề tài này, cần có kiến thức và kỹ năng về lập trình vi điều khiển, việc làm việc với các module và việc kết nối chúng với ESP32. Điều này đòi hỏi cần có hiểu biết về vi điều khiển, giao tiếp chuẩn (như I²C, SPI), lập trình Arduino hoặc ESP-IDF và các ngôn ngữ lập trình liên quan.

Phần cứng và kết nối: Sử dụng nhiều module trong một đề tài có thể đặt ra thách thức về phần cứng và kết nối. Bạn cần đảm bảo rằng tất cả các module có thể tương tác và kết nối với nhau một cách hợp lý. Đồng thời, cũng cần kiểm tra và xử lý các mâu thuẫn phần cứng như xung đột chân, cung cấp điện phù hợp và đảm bảo đủ nguồn điện cho tất cả các module.

Tài nguyên giới hạn: ESP32 có một số tài nguyên giới hạn như bộ nhớ và xung nhịp. Khi kết hợp nhiều module và chức năng, bạn cần quản lý tài nguyên một cách hiệu quả để không vượt quá giới hạn của ESP32 và đảm bảo hoạt động ổn định của ứng dụng.

Quản lý dữ liệu và tích hợp: Với việc sử dụng các module như INMP441, MAX98357 và MicroSD Adapter, bạn cần quản lý dữ liệu âm thanh và tích hợp chúng một cách chính xác vào ứng dụng của mình. Điều này bao gồm ghi âm, mã hoá âm thanh, lưu trữ dữ liệu và xử lý dữ liệu âm thanh.

Thời gian và phạm vi: Đề tài này có thể yêu cầu nhiều thời gian và công sức để nghiên cứu, triển khai và kiểm thử. Bạn cần xác định rõ phạm vi và thời gian thực hiện để đảm bảo có đủ tài nguyên và kế hoạch để hoàn thành dự án.

1.5. Bố cục

Chương 1: Tổng quan

Chương này trình bày về vấn đề của đề tài, mục tiêu hướng đến, nội dung nghiên cứu, giới hạn và bố cục của đề tài.

Chương 2: Cơ sở lý thuyết

Chương này trình bày về tổng quan hệ thống, các giao thức, các linh kiện được sử dụng trong hệ thống, các lý thuyết liên quan đến đề tài

Chương 3: Thiết kế và thực thi hệ thống

Chương 4: Kiểm tra và đánh giá

Chương 5: Hướng phát triển

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về Google Firebase

2.1.1. Giới thiệu về Google Firebase

Google Firebase là một nền tảng phát triển ứng dụng đám mây (cloud) do Google cung cấp. Nó cung cấp một loạt các dịch vụ và công cụ để xây dựng, triển khai và quản lý ứng dụng di động và web. Firebase kết hợp các tính năng BaaS (backend-as-a-service) và PaaS (platform-as-a-service), giúp nhà phát triển xây dựng ứng dụng di động và web dễ dàng và hiệu quả, giúp giảm thiểu công việc quản lý hạ tầng và tăng tốc quá trình phát triển.

2.1.2. Chức năng của Google Firebase

- **Firebase Authentication:** Dịch vụ xác thực người dùng, cung cấp các phương thức đăng nhập, đăng ký và quản lý người dùng.
- **Firebase Realtime Database:** Cơ sở dữ liệu thời gian thực, cho phép lưu trữ và đồng bộ dữ liệu trực tiếp giữa các thiết bị và ứng dụng.
- **Firebase Cloud Firestore:** Cơ sở dữ liệu đám mây linh hoạt, có khả năng tổ chức dữ liệu theo cấu trúc và hỗ trợ truy vấn phức tạp.
- **Firebase Storage:** Dịch vụ lưu trữ đám mây, cho phép lưu trữ và quản lý các tệp tin, hình ảnh, video và dữ liệu tương tự.
- **Firebase Cloud Messaging:** Dịch vụ thông báo đám mây, cho phép gửi thông báo đến các thiết bị di động và máy tính để bàn.
- **Firebase Hosting:** Dịch vụ lưu trữ và phân phối ứng dụng web tĩnh, giúp triển khai và quản lý trang web dễ dàng.
- **Firebase Functions:** Dịch vụ viết mã backend, cho phép triển khai và chạy các hàm (functions) trên máy chủ Firebase.

Ngoài ra, Firebase còn cung cấp các tính năng khác như phân tích và theo dõi hiệu suất ứng dụng, kiểm tra phân tích A/B, quảng cáo và tài chính, v.v. Firebase giúp đơn giản hóa việc phát triển ứng dụng, cung cấp một hệ sinh thái mạnh mẽ và linh hoạt để xây dựng và quản lý ứng dụng.

2.1.3. Ưu điểm và nhược điểm của Google Firebase

Ưu điểm

- Dễ sử dụng: Firebase cung cấp một giao diện người dùng thân thiện và các API dễ sử dụng, giúp nhà phát triển nhanh chóng triển khai và quản lý ứng dụng.
- Tích hợp hoàn hảo: Firebase tích hợp tốt với các sản phẩm và dịch vụ khác của Google như Google Cloud Platform, Google Analytics, Google Ads, v.v. Điều này giúp tăng tính toàn vẹn và mở rộng của ứng dụng.
- Real-time data synchronization: Firebase Realtime Database và Firebase Cloud Firestore cho phép đồng bộ dữ liệu theo thời gian thực giữa các thiết bị và ứng dụng, giúp tạo trải nghiệm tương tác và cập nhật nhanh chóng.
- Hỗ trợ nền tảng đa thiết bị: Firebase hỗ trợ phát triển ứng dụng trên nhiều nền tảng như iOS, Android và web, giúp tạo ra ứng dụng đa nền tảng dễ dàng.
- Dịch vụ phía máy chủ: Firebase Functions cho phép viết mã backend và triển khai các hàm trên máy chủ Firebase, giúp xử lý logic

Nhược điểm

- Giới hạn quy mô: Firebase có giới hạn về quy mô dữ liệu và sự mở rộng so với các dịch vụ đám mây khác như AWS hay Google Cloud Platform. Điều này có thể gây hạn chế cho các ứng dụng lớn và có tải trọng cao.
- Phụ thuộc vào hệ sinh thái Google: Firebase được phát triển bởi Google, do đó nó có sự phụ thuộc vào hệ sinh thái của Google và các sản phẩm của họ. Điều này có thể tạo ra rào cản khi muốn tích hợp với các dịch vụ hoặc công nghệ không thuộc hệ sinh thái Google.
- Chi phí: Một số dịch vụ của Firebase có chi phí sử dụng, đặc biệt là khi ứng dụng có quy mô lớn hoặc có lưu lượng truy cập cao. Điều này cần được xem xét khi tính toán ngân sách phát triển ứng dụng.
- Hạn chế về kiểm soát: Do Firebase là một dịch vụ đám mây, nên việc kiểm soát hoàn toàn các khía cạnh hạ tầng và dữ liệu có thể bị giới hạn. Điều này có thể tạo ra một số hạn

phức tạp và tương tác với các dịch vụ bên ngoài.

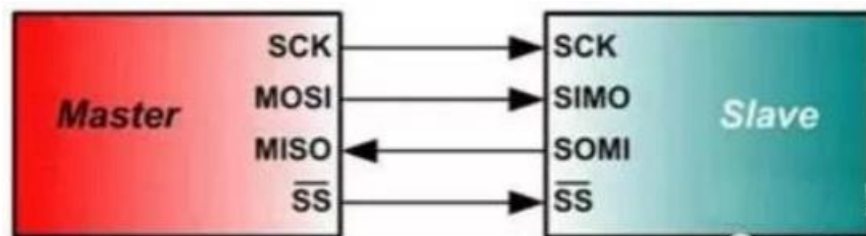
chế trong việc quản lý và kiểm soát ứng dụng.

2.2. Giới thiệu về giao thức truyền thông được sử dụng trong hệ thống

2.2.1. SPI

SPI (Serial Peripheral Interface) là một giao thức truyền thông đồng bộ do MOTOROLA đề xuất, được sử dụng để kết nối và truyền dữ liệu giữa vi điều khiển (master) và các thiết bị ngoại vi (slaves). Giao thức SPI bao gồm các chân giao tiếp như sau:

- MOSI (Master Out, Slave In): Chân MOSI được sử dụng để truyền dữ liệu từ master đến slave.
- MISO (Master In, Slave Out): Chân MISO được sử dụng để truyền dữ liệu từ slave đến master.
- Master điều khiển quá trình truyền dữ liệu bằng cách tạo ra một clock (SCK - Serial Clock) và chọn thiết bị slave (bằng cách kích hoạt chân SS - Slave Select) để truyền dữ liệu tới. Các thiết bị slave chỉ truyền dữ liệu khi được kích hoạt bởi master thông qua chân SS.



Hình 2.1. Giao thức SPI

SPI sử dụng một số cài đặt chế độ truyền thông khác nhau, bao gồm số lượng bit trên mỗi frame, thứ tự bit truyền, và cách chọn thiết bị slave. Ngoài ra, nó cũng hỗ trợ tốc độ truyền dữ liệu cao.

Giao thức SPI rất linh hoạt và được sử dụng để kết nối và truyền dữ liệu giữa vi điều khiển và các thiết bị như cảm biến, màn hình, bộ nhớ ngoại vi, module giao tiếp, và nhiều loại linh kiện khác.

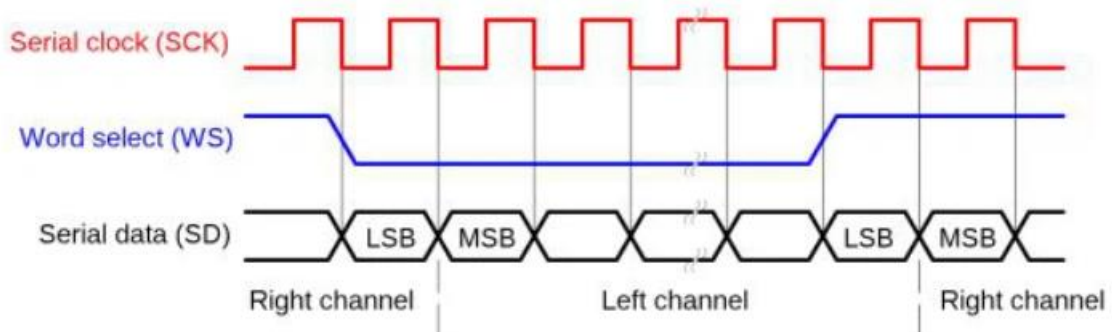
2.2.2. I2S

I2S là một giao thức truyền dữ liệu âm thanh số giữa các thiết bị điện tử, được phát triển bởi Philips Semiconductor (hiện là NXP Semiconductors) và được sử dụng rộng rãi trong các ứng dụng âm thanh như thiết bị ghi và phát âm thanh, hệ thống âm thanh trong ô tô, thiết bị âm thanh gia đình và nhiều ứng dụng khác.

I2S có 3 tín hiệu chính:

- BCLK: tức là tương ứng với mỗi bit dữ liệu âm thanh số, SCLK có 1 xung. Tần số của SCLK = $2 \times \text{tần số lấy mẫu} \times \text{số bit lấy mẫu}$.
- LRCLK (Left Right Clock) hoặc WS (Word Select): được sử dụng để chuyển đổi dữ liệu của kênh trái và phải. Khi LRCLK có mức logic cao, nó chỉ định dữ liệu âm thanh thuộc kênh trái, và khi LRCLK có mức logic thấp, nó chỉ định dữ liệu âm thanh thuộc kênh phải. LRCLK cũng quy định số lượng bit trên mỗi frame và tần số mẫu (sampling rate) của âm thanh.
- SD (Serial Data): là dữ liệu âm thanh được truyền theo chuẩn I2S. Dữ liệu âm thanh được chia thành các khung (frame) và truyền theo dạng tuần tự (serial) qua tín hiệu SD. Mỗi bit âm thanh được gói gọn trong một frame và được truyền theo thứ tự từ bit cao nhất đến bit thấp nhất hoặc ngược lại, tùy thuộc vào cấu hình của giao thức I2S.

Các tín hiệu BCLK, LRCLK và SD được sử dụng trong giao thức I2S để truyền dữ liệu âm thanh giữa các thiết bị như vi điều khiển, chip âm thanh, codec âm thanh và các thiết bị âm thanh khác.



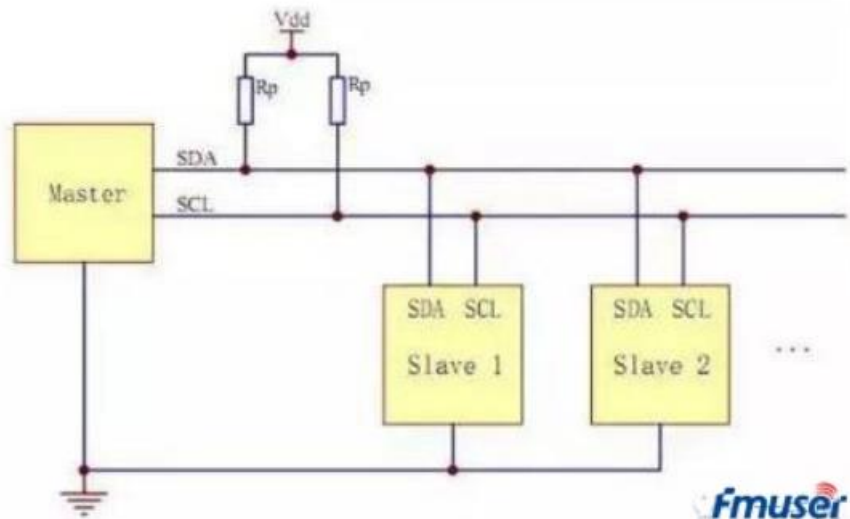
Hình 2.2. Sơ đồ thời gian của I2S

2.2.3. I2C

I2C một giao thức nối tiếp do Philips Semiconductors phát minh vào năm 1982, được sử dụng cho giao diện hai dây để kết nối các thiết bị tốc độ thấp như vi điều khiển, EEPROM, bộ chuyển đổi ADC và DAC, giao diện I/O và các thiết bị ngoại vi tương tự khác trong hệ thống. Nó được sử dụng để truyền dữ liệu tuần tự giữa các thiết bị với tốc độ truyền nhanh và hỗ trợ nhiều thiết bị trên cùng một bus.

I2C là một giao thức truyền thông hai dây gồm SCL (Serial Clock) và SDA (Serial Data) để truyền thông tin giữa bus và thiết bị, giao tiếp nối tiếp giữa vi điều khiển và các thiết bị bên ngoài, hoặc truyền dữ liệu hai chiều giữa thiết bị chính và thiết bị phụ. I2C thường được sử dụng để truyền tín hiệu điều khiển.

Giao thức I2C có khả năng kết nối nhiều thiết bị ngoại vi với số dây ít hơn so với giao thức SPI, làm cho nó phổ biến trong các ứng dụng điện tử. Nó cung cấp khả năng truyền dữ liệu đơn giản và dễ dàng mở rộng hệ thống thông qua việc thêm các thiết bị ngoại vi mới.



Hình 2.3. Giao thức I2C

2.2.4. SDIO

SDIO (Secure Digital Input/Output) là một giao thức truyền thông dựa trên giao diện bus để kết nối các thiết bị ngoại vi với các thiết bị lưu trữ dựa trên thẻ SD (Secure Digital). Ngoài khả năng kết nối với thẻ SD, nó cũng có thể được kết

nối với các thiết bị hỗ trợ giao diện SDIO. Nó cho phép truyền dữ liệu giữa thiết bị chủ (host device) như vi điều khiển hoặc máy tính và các thiết bị như thẻ nhớ SD, Wi-Fi module, Bluetooth module và các thiết bị khác.

Giao thức SDIO sử dụng giao diện bus có đường dẫn nhiều dây (multibit) để truyền dữ liệu và điều khiển. Nó hỗ trợ truyền dữ liệu song song, trong đó các bit dữ liệu được truyền đồng thời qua các dây truyền riêng biệt. Giao thức này cũng hỗ trợ chế độ nối tiếp, trong đó các bit dữ liệu được truyền qua một dây truyền duy nhất theo thời gian.

Giao thức SDIO được sử dụng chủ yếu trong các ứng dụng di động như điện thoại di động, máy tính bảng, máy ảnh kỹ thuật số và các thiết bị nhúng khác. Nó cho phép việc mở rộng khả năng lưu trữ, kết nối mạng và giao tiếp không dây thông qua các module ngoại vi SDIO.

Giao thức SDIO đòi hỏi việc cài đặt phần mềm phù hợp và hỗ trợ phần cứng từ thiết bị chủ và thiết bị ngoại vi. Các tốc độ truyền dữ liệu và chế độ truyền dữ liệu cụ thể cũng được xác định bởi các thiết bị tham gia trong hệ thống.

2.2.5. *UART*

Giao thức UART (Universal Asynchronous Receiver/Transmitter) là một giao thức truyền thông không đồng bộ thông qua cổng nối tiếp (serial port). Nó được sử dụng để truyền dữ liệu giữa các thiết bị thông qua hai dây truyền, một dây truyền (TX - Transmit) và một dây nhận (RX - Receive).

Giao thức UART không đồng bộ có nghĩa là không có tín hiệu đồng hồ chung giữa các thiết bị gửi và nhận. Thay vào đó, dữ liệu được truyền đi theo từng byte và được đồng bộ hóa bằng các bit Start và Stop. Mỗi byte dữ liệu cũng có thể đi kèm với một số bit kiểm tra (parity bit) để kiểm tra tính chính xác của dữ liệu.

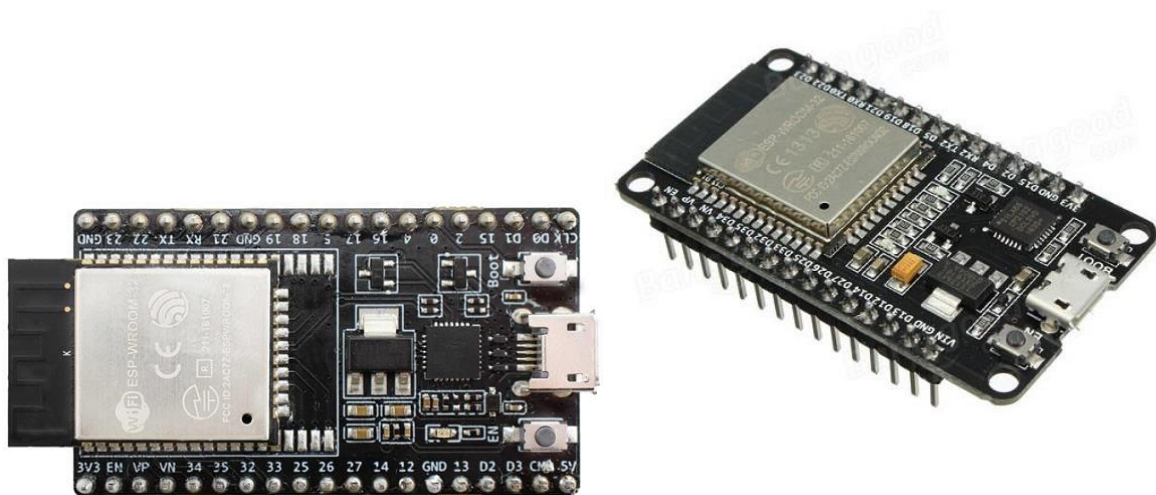
UART được sử dụng rộng rãi trong việc kết nối và truyền thông giữa các thiết bị điện tử, chẳng hạn như vi điều khiển, cảm biến, mạch điện tử và các thiết bị ngoại vi. Nó cung cấp một giao diện đơn giản và đáng tin cậy để truyền dữ liệu

điện tử mà không đòi hỏi tốc độ truyền cao như các giao thức truyền thông khác như SPI và I2C.

2.3. Giới thiệu về linh kiện sử dụng trong hệ thống

2.3.1. Giới thiệu về ESP32

ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp. Có thể lập trình cho ESP32 bằng nhiều ngôn ngữ khác nhau như C/C++, Python, NodeJs, Lua, ...



Hình 2.4. ESP32 DevKitC và DOIT ESP32 DEVKIT V1

ESP32	Thông số kỹ thuật
Bộ xử lý	CPU
Hệ thống xung nhịp	CPU Clock, RTC Clock, Audio PLL Clock
Số chân GPIO	34 GPIO pad vật lý
Kênh SAR ADC 12 bit	18 kênh
Kênh DAC 8 bit	2 kênh
Tốc độ xử lý	160MHz – 240MHz
Tốc độ xung nhịp đọc flash chip	40MHz – 80MHz
ROM	448Kbyte ROM
External FLASH	4MB

lượng âm thanh tốt cho các ứng dụng âm thanh như loa, hệ thống âm thanh nhúng, máy nghe nhạc, và các thiết bị di động.



Hình 2.6. Ảnh thực tế của module MAX98357

Sơ đồ chân của MAX98357:

- LRC: Đồng bộ hoá kênh trái/phải
- BCLK: Tín hiệu clock bit
- DIN: Dữ liệu âm thanh đầu vào
- GAIN: Kênh chọn GAIN
- SD: Kênh chọn ShutDown (SD_Mode ở mức thấp thì module sẽ ở trạng thái ShutDown)
- GND: GND
- Vin: DC 2.5V – 5V

Thông số kĩ thuật:

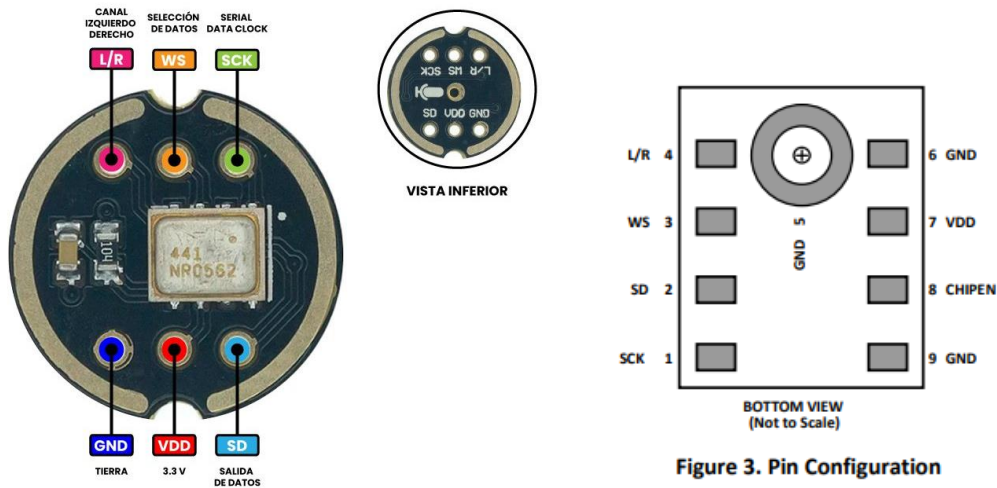
Điện áp hoạt động	2.5V - 5.5V
Công suất	3W
Công suất đầu ra	<ul style="list-style-type: none"> • Tối đa 3,2W cho tải trở kháng 4Ω và nguồn cấp 5V.

	<ul style="list-style-type: none"> Tối đa 1,7W cho tải trở kháng 8Ω và nguồn cấp 5V.
Dải tần số âm thanh đầu vào	20Hz - 20kHz
Giao tiếp	I2S

Bảng 2.2. Thông số kĩ thuật của module MAX98357

2.3.3. Giới thiệu về module INMP441

Module INMP441 là một module cảm biến âm thanh I2S Omnidirectional Microphone được sử dụng như một Microphone đa hướng giúp thu âm thanh từ môi trường, chuyển thành tín hiệu số (Digital) và truyền tới Vi điều khiển qua giao tiếp I2S, thích hợp với các ứng dụng như thu âm và xử lý âm thanh trong các ứng dụng điện tử.



Hình 2.7. Ảnh thực tế và sơ đồ chân của module INMP441

Sơ đồ chân:

- L/R: Chân chỉ định kênh trái phải, cho phép lựa chọn kênh đầu ra của microphone trong khung I²S. Khi chân này ở mức thấp, tín hiệu được đưa ra ở kênh trái, còn khi chân này ở mức cao, tín hiệu sẽ được đưa ra ở kênh phải.
- WS: chân này xác định các từ dữ liệu trong khung truyền của giao diện. Khi chân WS được đặt ở mức thấp, các bit dữ liệu được truyền trong khung I²S sẽ được coi là một luồng dữ liệu cho kênh âm thanh bên trái. Và khi chân WS được đặt ở mức cao, các bit dữ liệu được truyền trong khung I²S sẽ được coi là một luồng dữ liệu cho kênh âm thanh bên phải.
- SCK: Tín hiệu clock cho giao thức I²S
- SD: sử dụng giao thức I²S, Chân này sẽ chuyển sang trạng thái "tri-state" khi không hoạt động, tức là không đưa tín hiệu ra kênh đầu ra tương ứng. Để đảm

bảo việc xả tải đường truyền khi tất cả các microphone trên bus đã chuyển sang trạng thái "tri-state", chân SD nên có một trở kéo xuống (pulldown resistor) 100 kΩ để xả tải đường truyền trong thời gian đó.

- GND: GND
- VDD: Nguồn cung cấp từ 1.8V – 3.3V

Thông số kỹ thuật:

Điện áp sử dụng	1.8V – 3.3V
Giao diện Digital I²S	Dữ liệu 24-bit có độ chính xác cao
SNR (Signal-to-noise Ratio)	61dBA Đảm bảo chất lượng âm thanh tốt
Độ nhạy	-26 dBFS Cho phép ghi âm các âm thanh nhỏ và chi tiết
Tần số	60Hz – 15kHz
Cường độ dòng điện tiêu thụ	1.4mA
PSR (Power Supply Rejection) Hiệu suất kháng nhiễu	-75dBFS Giảm ảnh hưởng của nhiễu nguồn lên chất lượng âm thanh

Bảng 2.3. Thông số kỹ thuật của module INMP441

2.3.4. Giới thiệu về module Rotary Encoder

Rotary Encoder (còn được gọi là bộ mã hóa quay) là một thiết bị điện tử được sử dụng để ghi lại và định vị các vòng quay hoặc chuyển động quay. Nó được sử dụng rộng rãi trong các ứng dụng như điều khiển và ghi lại vị trí, điều khiển âm lượng, điều khiển menu và các ứng dụng tương tự khác.



Hình 2.8. Ảnh thực tế của module Rotary Encoder

Sơ đồ chân:

- CLK: Chân CLK tạo ra các xung clock khi encoder được quay.
- DT: Chân Data, cung cấp chuỗi xung.
- SW: Chân Switch là chân nút nhấn.
- VCC: Nguồn cung cấp từ 3.3V – 5V
- GND: GND

Thông số kỹ thuật:

Độ rộng rãnh	5mm
Điện áp hoạt động	3.3V – 5V
Kích thước	3.2cm x 1.4cm
Dòng output	15mA

Bảng 2.4. Thông số kỹ thuật của module Rotary Encoder

2.3.5. Giới thiệu về module ILI9341

ILI9341 là một loại màn hình LCD TFT (Thin Film Transistor) có kích thước 2,8 inch và độ phân giải 320x240 pixel giao tiếp SPI được sử dụng trong các ứng dụng điều khiển cảm ứng và hiển thị, giúp xây dựng giao diện điều khiển cảm ứng trên màn hình. Nó được sử dụng rộng rãi trong các ứng dụng nhúng và dự án điện tử để hiển thị thông tin.

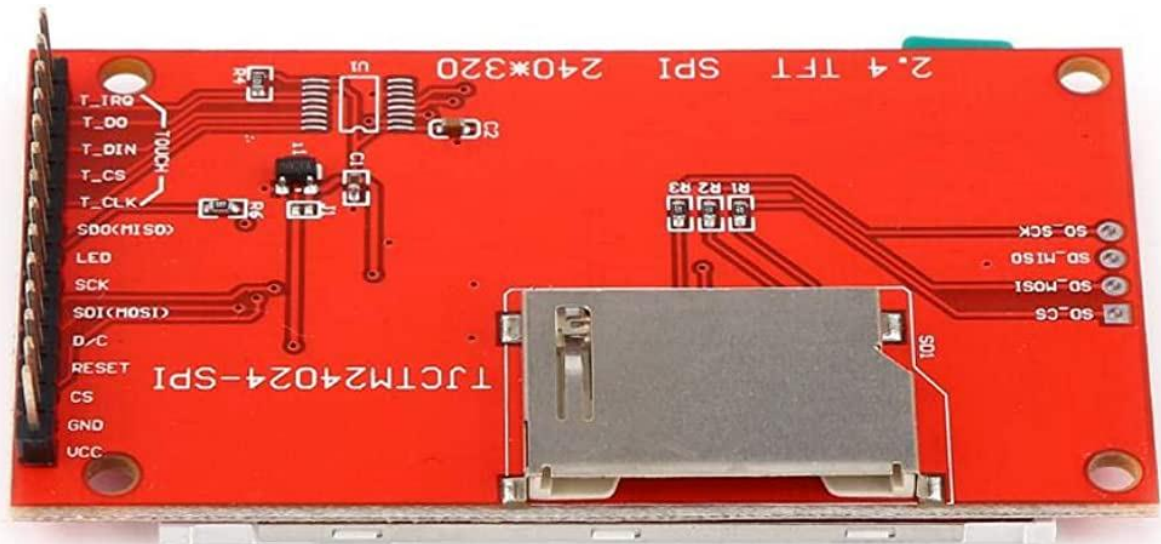


Hình 2.9. Ảnh thực tế của module ILI9341

Sơ đồ chân của ILI9341:

- VCC: Nguồn cung cấp từ 3.3V – 5V DC
- GND: GND
- CS: Chân Select
- RST: Chân Reset
- DC: Chân điều khiển dữ liệu Data/Command
- SDI (MOSI): Chân dữ liệu đầu vào (Serial Data In / Master Out – Slave In)
- SCK: Chân CLK
- LED: Đèn nền LED
- SDO (MISO): Chân dữ liệu đầu ra (Serial Data Out / Master In – Slave Out)

- T_CLK: Chân cảm ứng (Clock)
- T_CS: Chân cảm ứng chọn
- T_DIN: Chân cảm ứng dữ liệu đầu vào
- T_DO: Chân cảm ứng dữ liệu đầu ra
- T_IRQ: Chân ngắt cảm ứng



Hình 2.10. Sơ đồ chân của ILI9341

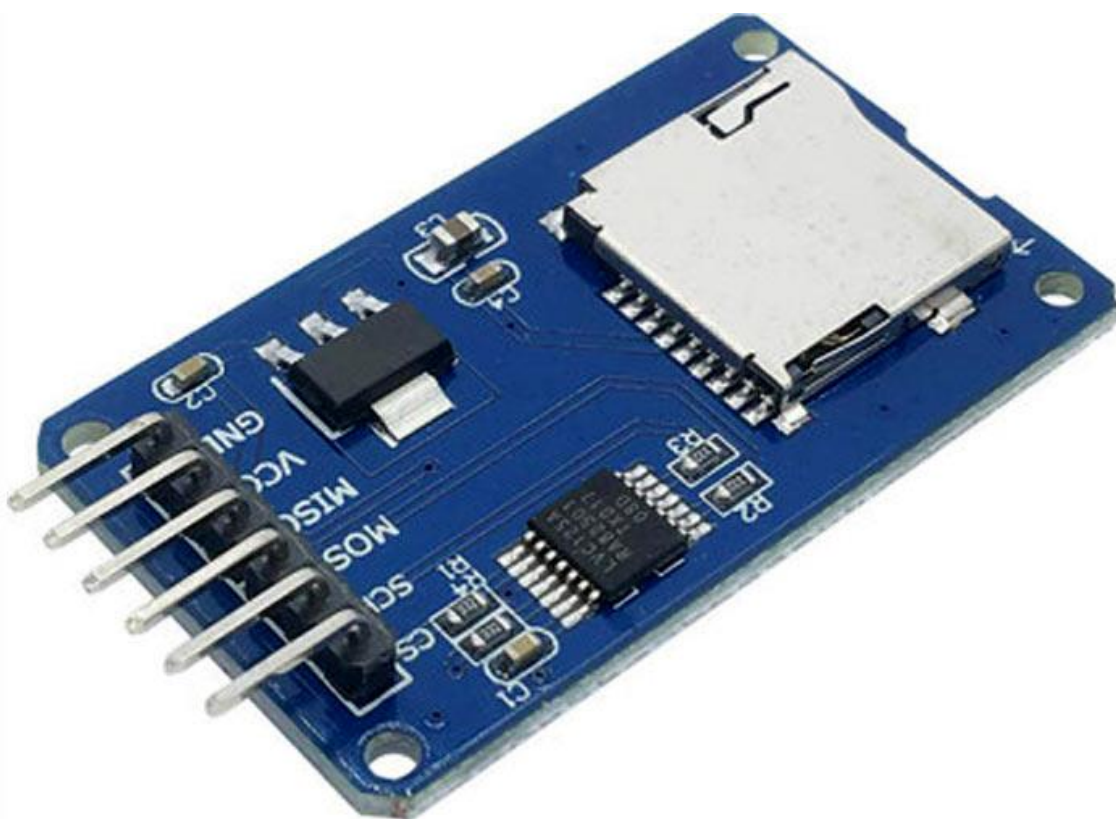
Thông số kỹ thuật:

Điện áp sử dụng	3.3V – 5V
Điện áp giao tiếp	TTL 3.3 ~ 5V DC
IC Driver hiển thị	ILI9341 giao tiếp SPI
Kích thước màn hình	2.8 inch
Độ phân giải	320 x 240 pixels
IC Driver cảm ứng	XPT2046 giao tiếp SPI
Kích thước hiển thị	46(W) x 65(H) mm
Kích thước toàn màn hình	86 x 50 mm
	Tích hợp khe thẻ nhớ SD giao tiếp SPI

Bảng 2.5. Thông số kỹ thuật của ILI9341

2.3.6. Giới thiệu về module MicroSD Adapter

Module MicroSD Adapter là một module mở rộng được sử dụng để kết nối và đọc/ghi dữ liệu từ thẻ nhớ microSD trên các hệ thống nhúng.



Hình 2.11. Ảnh thực tế của Module MicroSD Adapter

Sơ đồ chân:

- GND: GND
- VCC: Nguồn cung cấp 4.5V – 5.5V
- MISO: Chân dữ liệu đầu ra (Master In Slave Out)
- MOSI: Chân dữ liệu đầu vào (Master Out Slave In)
- SCK: Chân CLOCK
- CS: Chân Select

Thông số kỹ thuật:

Điện áp hoạt động	4.5V – 5.5V
Dòng điện	0.2 – 200mA
Bộ điều khiển điện áp	3.3V
Kích thước	42 x 24 x 12 mm
	<ul style="list-style-type: none"> • Hỗ trợ thẻ nhớ SD lên đến 2GB • Hỗ trợ thẻ nhớ SDHC lên đến 32GB

Bảng 2.6. Thông số kỹ thuật của MicroSD Adapter

CHƯƠNG 3: THIẾT KẾ VÀ THỰC THI HỆ THỐNG

3.1. Giới thiệu hệ thống

3.1.1. Giới thiệu và tóm tắt hệ thống

Hệ thống được giới thiệu là một ứng dụng điều khiển âm thanh, ghi âm và quản lý lịch trình thông qua vi điều khiển ESP32 và các module kết hợp. Hệ thống cho phép người dùng điều khiển âm lượng, ghi âm âm thanh và quản lý lịch trình thông qua một giao diện người dùng trực quan.

Vi điều khiển ESP32 được sử dụng làm khối xử lý trung tâm của hệ thống. Nó kết hợp với module màn hình cảm ứng ILI9341, module ghi âm INMP441, module mã hoá âm thanh MAX98357A, và module Rotary Encoder để tạo thành một hệ thống hoàn chỉnh.

Người dùng có thể điều chỉnh âm lượng bằng cách sử dụng Rotary Encoder, với thông tin hiển thị trên màn hình ILI9341. Họ cũng có thể thực hiện chức năng ghi âm bằng cách chạm vào màn hình và ghi lại âm thanh thông qua module INMP441. Dữ liệu ghi âm được lưu trữ trên thẻ nhớ MicroSD thông qua module MicroSD Adapter.

Ngoài ra, hệ thống kết nối với Firebase để lấy thông tin lịch trình và công việc. Người dùng có thể xem thông tin lịch trình và cập nhật nó thông qua giao diện người dùng trên màn hình ILI9341. Tất cả dữ liệu và tương tác của người dùng được xử lý và điều khiển bởi vi điều khiển ESP32.

Hệ thống cung cấp một giao diện đơn giản và thuận tiện cho người dùng để điều khiển âm thanh, ghi âm và quản lý lịch trình. Nó mang lại trải nghiệm tốt và đáp ứng nhu cầu của người dùng trong việc quản lý âm thanh và lịch trình cá nhân.

3.1.2. Yêu cầu và chức năng

- Điều khiển âm thanh: Hệ thống cho phép điều khiển âm lượng của âm thanh phát ra qua module MAX98357A. Sử dụng Rotary Encoder để tăng/giảm âm lượng một cách thuận tiện và chính xác.

- Ghi âm: Sử dụng module ghi âm INMP441 để thu âm và ghi lại âm thanh. Sử dụng màn hình cảm ứng ILI9341 để hiển thị các tùy chọn và ghi âm. Kết hợp chức năng chạm để bắt đầu/giữ ghi âm và dừng ghi âm.
- Quản lý lịch trình: Kết nối với Firebase để lấy thông tin lịch trình và công việc.
- Lưu trữ dữ liệu: Sử dụng module MicroSD Adapter để lưu trữ các file ghi âm và dữ liệu lịch trình. Hỗ trợ đọc/ghi dữ liệu từ thẻ nhớ MicroSD và thao tác với các file trên đó.
- Giao diện người dùng: Sử dụng màn hình cảm ứng ILI9341 để hiển thị thông tin và tương tác với người dùng. Cung cấp các nút điều khiển và chức năng để thao tác với âm thanh, ghi âm và lịch trình.

3.1.3. Sơ đồ tổng quát của hệ thống



Hình 3.1. Sơ đồ tổng quát của hệ thống

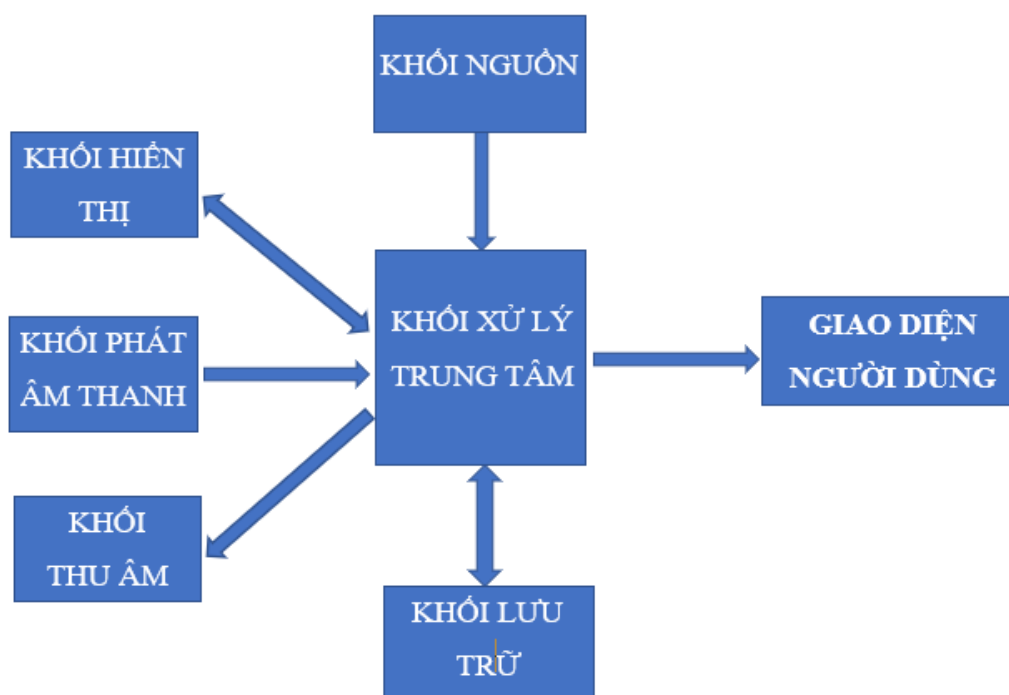
Bộ xử lý có nhiệm vụ là xử lý dữ liệu, quản lý và điều khiển các hoạt động của hệ thống.

Các module hoạt động với các chức năng riêng biệt, chúng nhận các tín hiệu điều khiển từ bộ xử lý.

Giao diện người dùng được sử dụng để tương tác và trao đổi thông tin giữa người dùng và hệ thống máy tính. Nó cung cấp một phương tiện trực quan và thân thiện để người dùng tương tác với các ứng dụng, dịch vụ và chức năng của hệ thống.

3.2. Sơ đồ và chức năng của các khối trong hệ thống

3.2.1. Sơ đồ khối



Hình 3.2. Sơ đồ khối của hệ thống

3.2.2. Chức năng từng khối

Khối xử lý trung tâm là một trong những thành phần quan trọng nhất của một hệ thống. Nhiệm vụ chính của khối là thực hiện các hoạt động tính toán và điều khiển các thành phần khác trong hệ thống. Trong hệ thống Todo-List này, sử dụng vi điều khiển ESP32 làm khối xử lý trung tâm.

Khối nguồn chịu trách nhiệm cung cấp nguồn điện cho các thành phần khác của hệ thống. Nó chuyển đổi nguồn điện từ nguồn cung cấp (như nguồn điện AC từ ổ cắm điện) thành nguồn điện DC ổn định và phù hợp để cung cấp cho các linh kiện bên trong hệ thống.

Khối hiển thị là một thành phần của hệ thống điện tử dùng để hiển thị thông tin cho người dùng. Khối hiển thị bao gồm màn hình LCD TFT. Khối hiển thị nhận thông tin từ khối xử lý trung tâm và hiển thị nội dung theo yêu cầu.

Khối phát âm thanh là nơi loa nhận tín hiệu âm thanh từ nguồn và chuyển đổi thành âm thanh phát ra. Khối phát âm thanh bao gồm các thành phần như bộ

khuếch đại âm thanh, loa trầm, loa giữa và loa tweeter để tái tạo các tần số khác nhau của âm thanh.

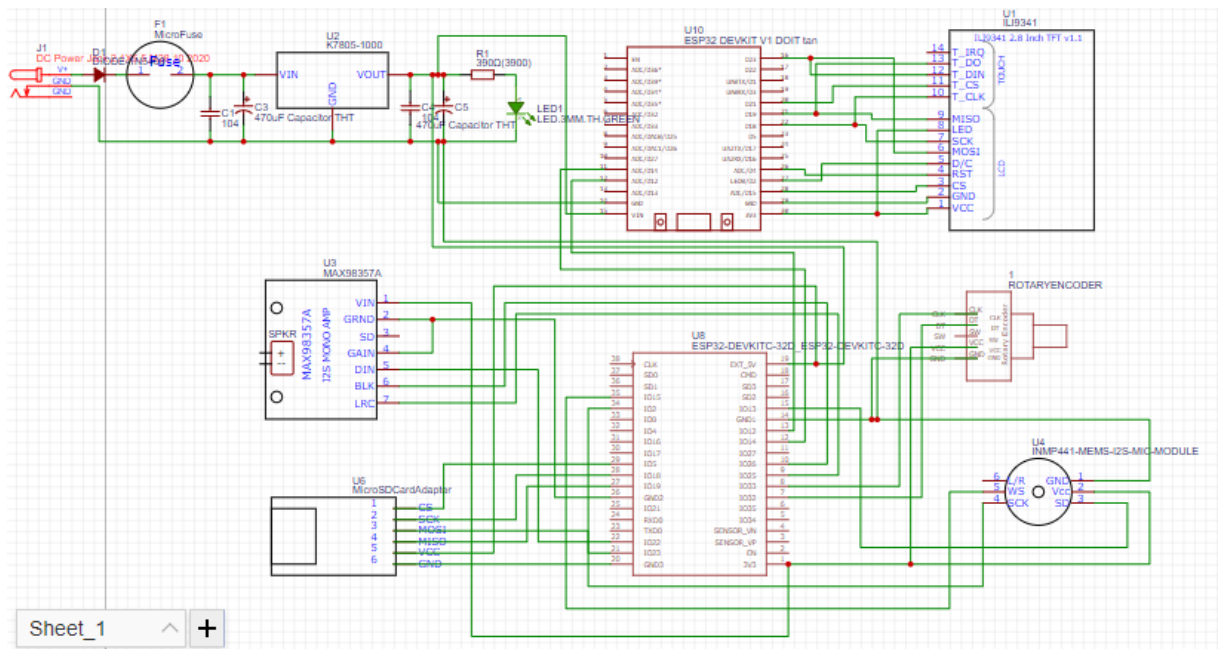
Khối thu âm bao gồm mic thu âm là thiết bị thu âm âm thanh từ môi trường xung quanh. Nó chuyển đổi âm thanh thành tín hiệu điện để được xử lý hoặc ghi lại bằng các thiết bị khác.

Khối lưu trữ được sử dụng để lưu trữ và truy cập dữ liệu và các tệp tin khác. Nó có khả năng lưu trữ dữ liệu trong thời gian dài mà không cần nguồn điện liên tục. Khối lưu trữ được kết nối với hệ thống thông qua giao thức SPI để truyền nhận dữ liệu.

Giao diện người dùng được sử dụng để tương tác và trao đổi thông tin giữa người dùng và hệ thống máy tính. Nó cung cấp một phương tiện trực quan và thân thiện để người dùng tương tác với các ứng dụng, dịch vụ và chức năng của hệ thống.

3.3. Thiết kế chi tiết

3.3.1. Sơ đồ nguyên lý

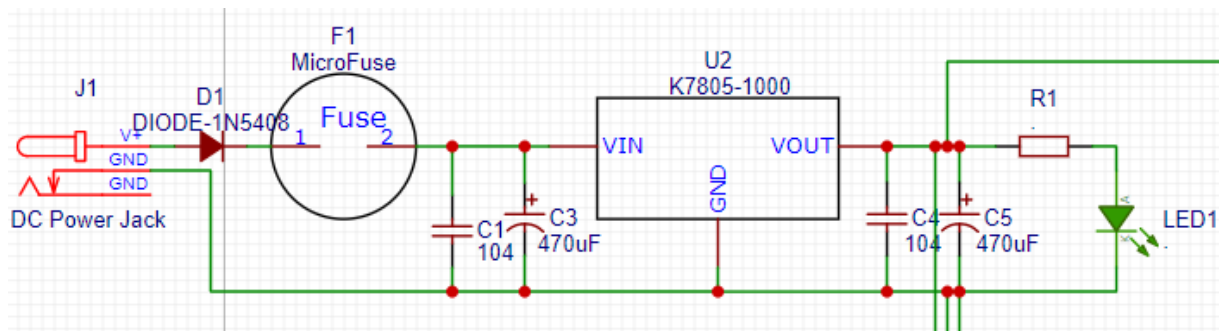


Hình 3.3. Sơ đồ nguyên lý của mạch

3.3.2. Thiết kế từng khối

- Khối nguồn:

Thiết kế nguồn để cấp nguồn điện 5V 1A cho 2 vi điều khiển ESP32 sử dụng ổn áp 7805, là một loại linh kiện điện tử được sử dụng rộng rãi trong các ứng dụng điện tử để cung cấp nguồn điện ổn định với giá trị đầu ra là 5V. Ổn áp 7805 là một loại IC 3 chân, trong đó chân 1 là chân đầu ra V_{OUT} , chân 2 là chân đất GND, và chân 3 là chân đầu vào V_{IN} . Khi được kết nối với nguồn điện đầu vào có giá trị lớn hơn 5V, ổn áp 7805 sẽ tự động điều chỉnh điện áp đầu ra xuống 5V. Ổn áp 7805 cũng có khả năng cung cấp dòng điện tới 1A, giúp đáp ứng nhu cầu điện áp và dòng điện của nhiều ứng dụng điện tử.



Hình 3.4. Sơ đồ khối nguồn

- Khối xử lý trung tâm:

Sau khi được cấp nguồn, mạch sẽ bắt đầu hoạt động và led sẽ được bật để chỉ ra rằng mạch đang hoạt động. Đồng thời, vi điều khiển trung tâm ESP32 đóng vai trò là bộ vi xử lý trung tâm (CPU) trong hệ thống của bạn và chịu trách nhiệm điều khiển toàn bộ hoạt động của đề tài. Nó có khả năng xử lý và thực thi mã lệnh, quản lý các tác vụ, giao tiếp với các module và linh kiện khác, và điều khiển các module trong hệ thống.

Giao tiếp với màn hình ILI9341: ESP32 sử dụng các chân GPIO và giao thức SPI (Serial Peripheral Interface) để giao tiếp với màn hình ILI9341. Nó gửi dữ liệu hình ảnh và lệnh điều khiển từ ESP32 đến màn hình và nhận phản hồi từ màn hình để điều khiển hiển thị và tương tác người dùng.

Giao tiếp với cảm biến INMP441: ESP32 sử dụng giao thức I2S (Inter-Integrated Sound) để giao tiếp với cảm biến INMP441. Nó nhận dữ liệu âm thanh từ cảm biến và có thể thực hiện các chức năng như ghi âm, xử lý âm thanh và phân tích.

Giao tiếp với module mã hoá âm thanh MAX98357: ESP32 sử dụng giao thức I2S để giao tiếp với module mã hoá âm thanh MAX98357. Nó gửi dữ liệu âm thanh đến module để phát ra loa và điều chỉnh âm lượng.

Giao tiếp với Rotary Encoder: ESP32 sử dụng các chân GPIO và giao thức GPIO để giao tiếp với Rotary Encoder. Nó nhận tín hiệu từ Rotary Encoder để điều chỉnh âm lượng âm thanh hoặc thực hiện các chức năng khác tùy thuộc vào thiết kế của đề tài.

Giao tiếp với MicroSD Adapter: ESP32 sử dụng giao thức SPI để giao tiếp với MicroSD Adapter. Nó gửi và nhận dữ liệu từ thẻ nhớ MicroSD để lưu trữ và truy xuất các tệp ghi âm hoặc dữ liệu khác.

- Khởi hiển thị:

Ta sử dụng màn hình LCD TFT có cảm ứng để hiển thị và nhập dữ liệu. Kết nối LCD với vi điều khiển ESP32 như sau:

ILI9341	ESP32
VCC	3V3
GND	GND
CS	15
RST	4
DC	2
MOSI	23
SCK	18
LED	3V3
MISO	19
T_CLK	18
T_CS	21
T_DIN	23
T_DO	19
T_IRQ	

Bảng 3.1. Bảng nối chân từ ESP32 tới LCD TFT

- Khởi phát âm thanh:

Ta sử dụng module MAX98357 để khuếch đại âm thanh thông qua giao tiếp I2S, sau đó gửi tín hiệu âm thanh đến loa. Kết nối MAX98357 với vi điều khiển ESP32 như sau:

MAX98357A	ESP32
VCC	3.3V
GND	GND
SD	
GAIN	GND

Din	G22
BLCK	G26
LRC	G25

Bảng 3.2. Bảng nối chân từ ESP32 tới module MAX98357

- **Khôi thu âm:**

Module INMP441 là một module âm thanh số chất lượng cao, được sử dụng để thu âm và xử lý âm thanh trong các ứng dụng điện tử. Kết nối INMP441 với vi điều khiển ESP32 như sau:

INMP441	ESP32
VCC	3V3
GND	GND
SD	13
L/R	
WS	33
SCK	15

Bảng 3.3. Bảng nối chân từ ESP32 tới module INMP441

3.4. Thiết kế và thi công mạch

3.4.1. Thiết kế PCB

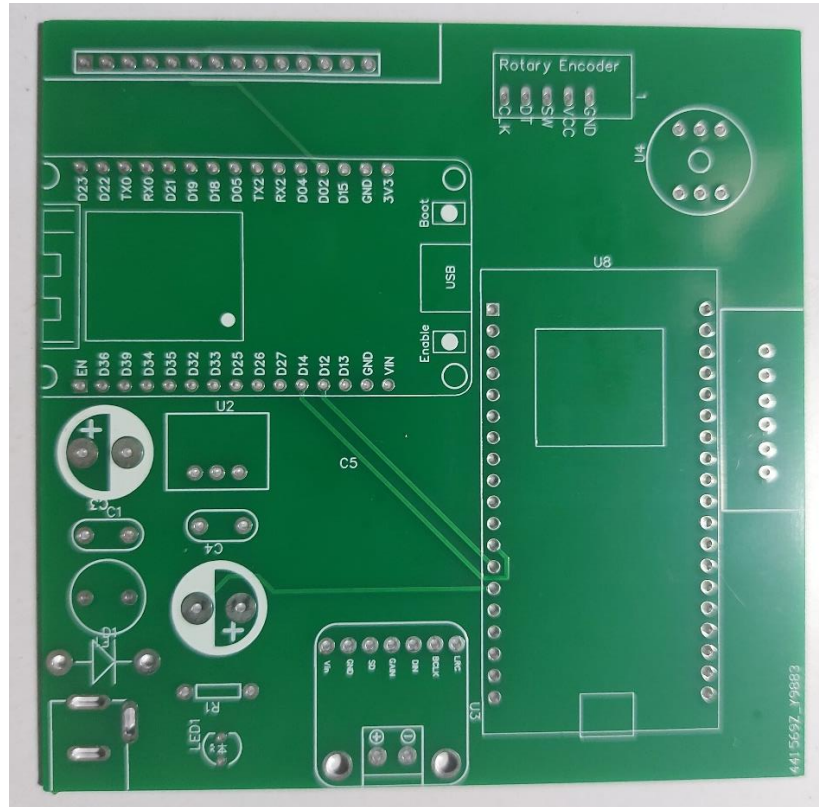


Hình 3.5. Mạch in

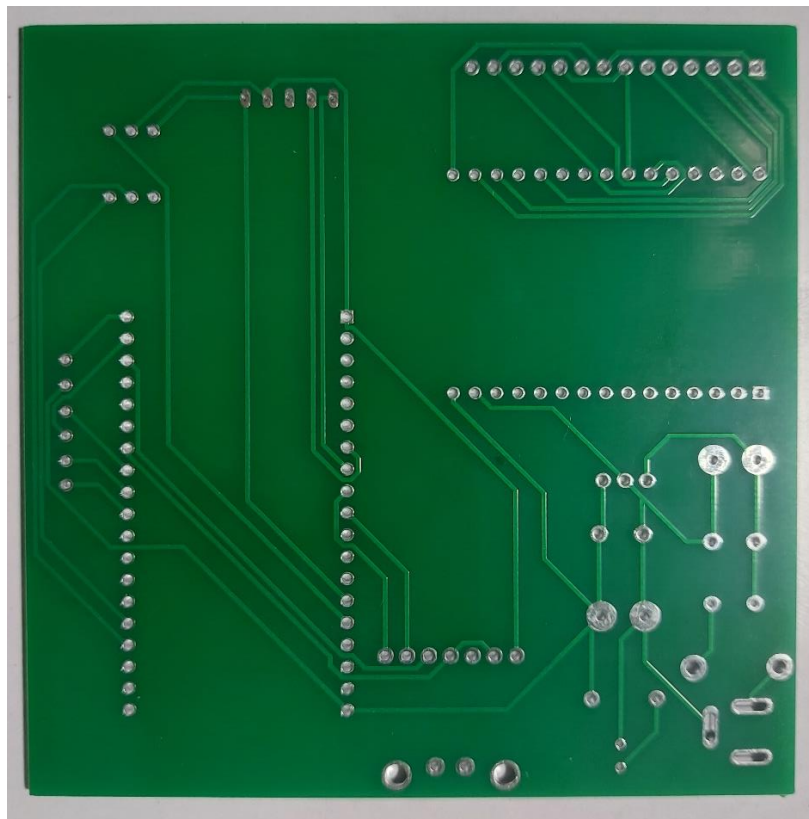
Các thông số của mạch in:

- Kích thước board: 9cm x 9cm.
- PCB được vẽ 2 lớp, tập chung chính đi dây ở lớp Bottom và một số dây ở lớp Top.
- Bề rộng của đường dây trên board là 20mil.

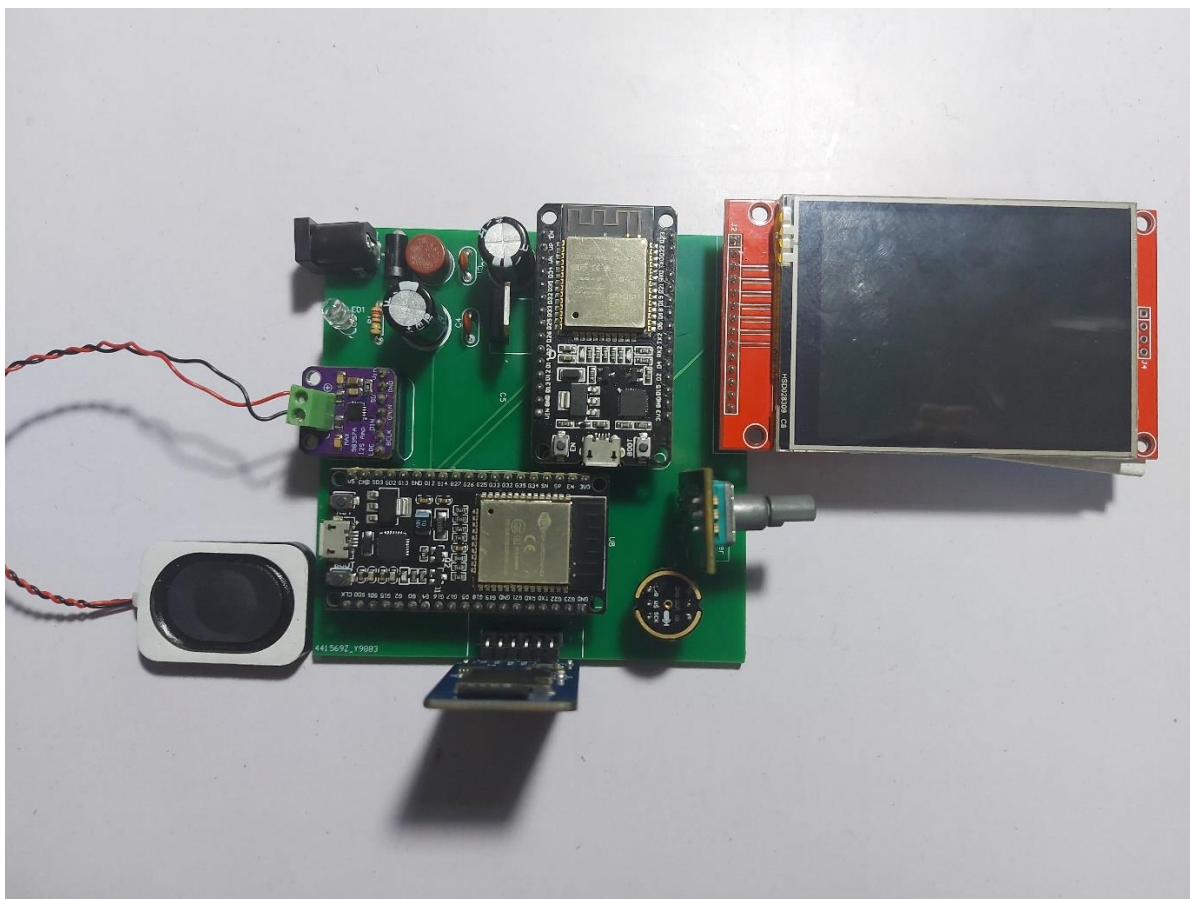
3.4.2. Thi công mạch



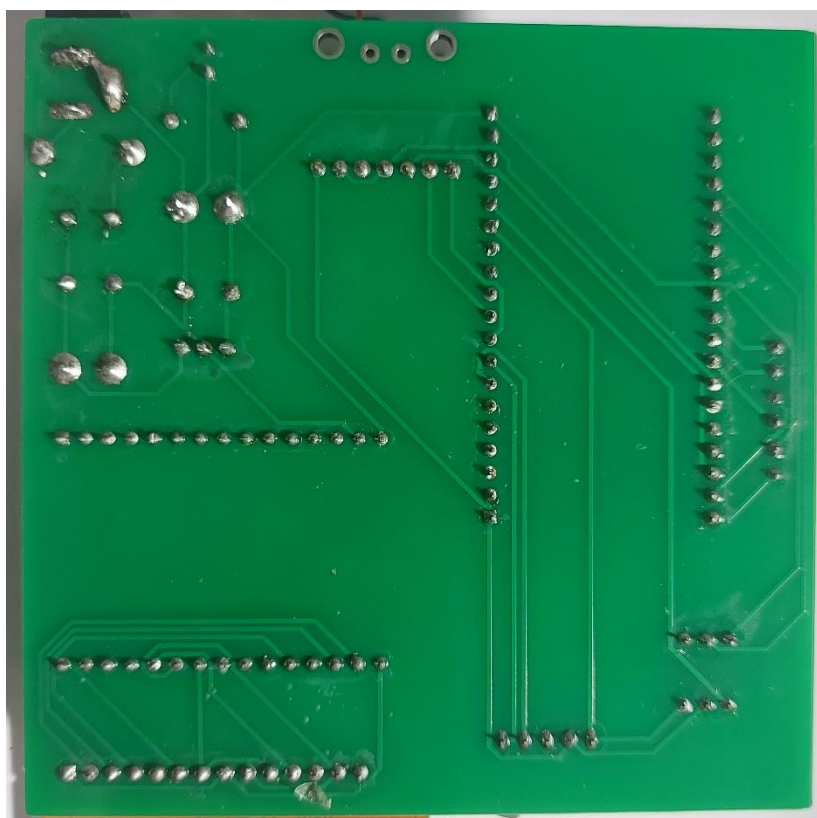
Hình 3.6. Mặt trên của PCB



Hình 3.7. Mặt dưới của PCB



Hình 3.8. Bản mạch full linh kiện (mặt trên)

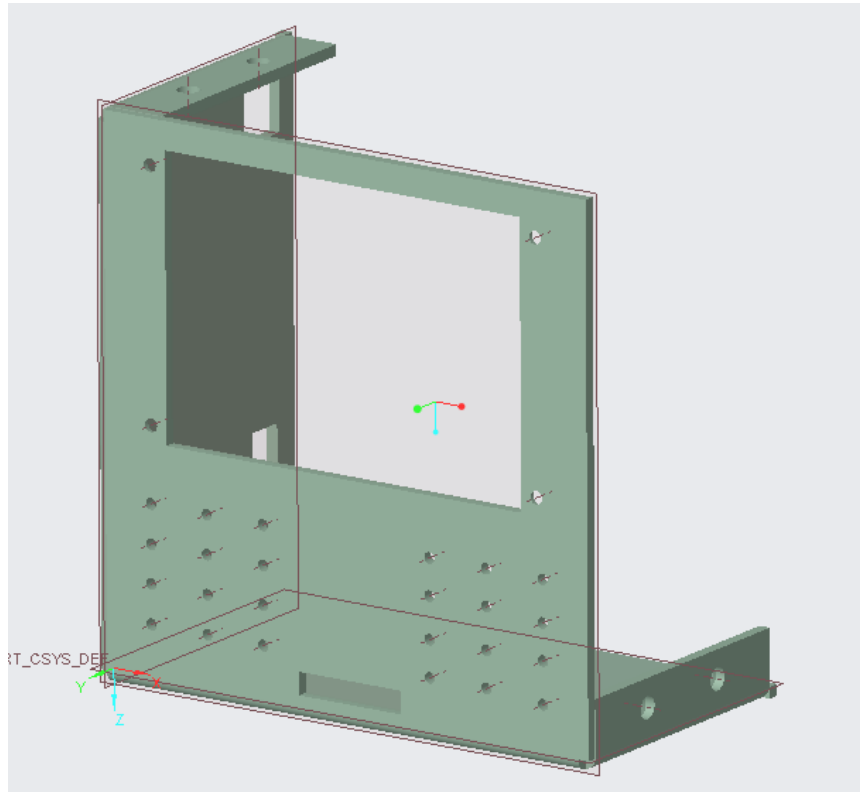


Hình 3.9. Bản mạch full linh kiện (mặt dưới)

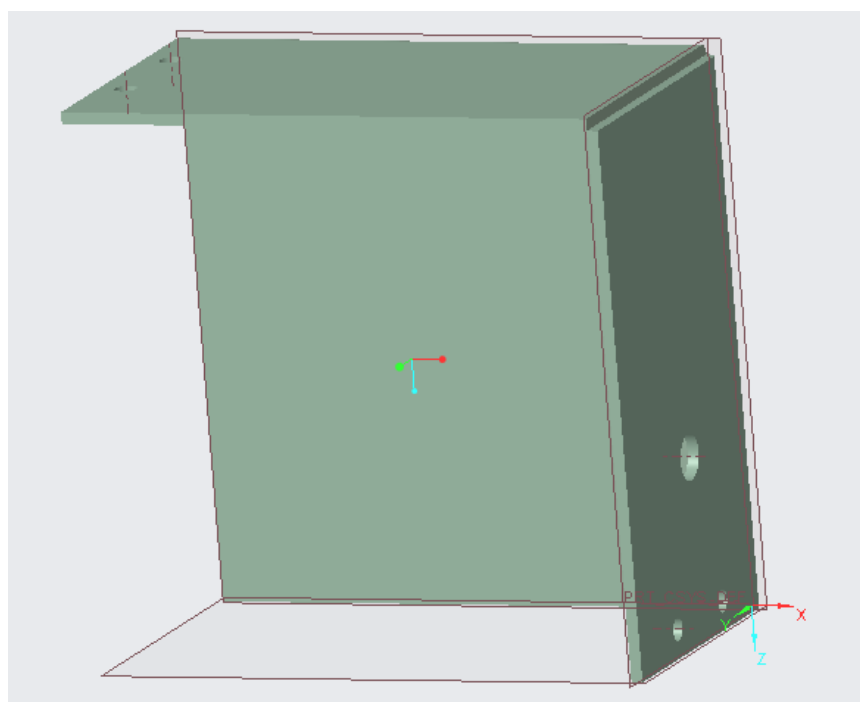
3.5. Thiết kế hộp cho hệ thống

Sử dụng phần mềm Creo Parametric phiên bản 8.0.0.0 để thiết kế hộp đóng gói cho thiết bị.

Dưới đây là bản vẽ 3D hộp đóng gói cho thiết bị:



Hình 3.10. Phần thứ nhất của hộp



Hình 3.11. Phần thứ hai của hộp

CHƯƠNG 4: KIỂM TRA VÀ ĐÁNH GIÁ

4.1. Kết quả quá trình thực hiện đề tài

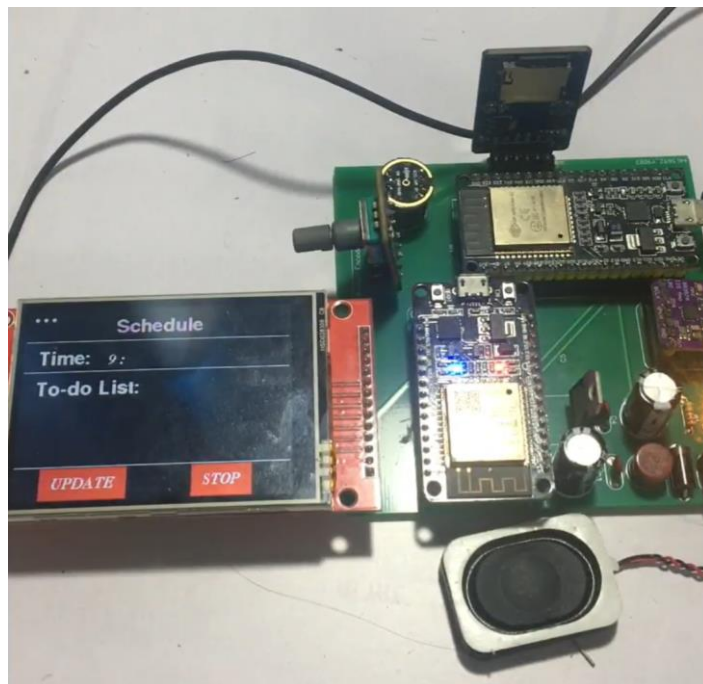
4.1.1. Kết quả đạt được

Sau thời gian nghiên cứu và phát triển đề tài, nhóm em đã đạt được một số kết quả. Từ đó có thể đưa ra được đánh giá, nhận xét và những hạn chế cũng như hướng phát triển để sản phẩm có thể đưa vào thực tế.

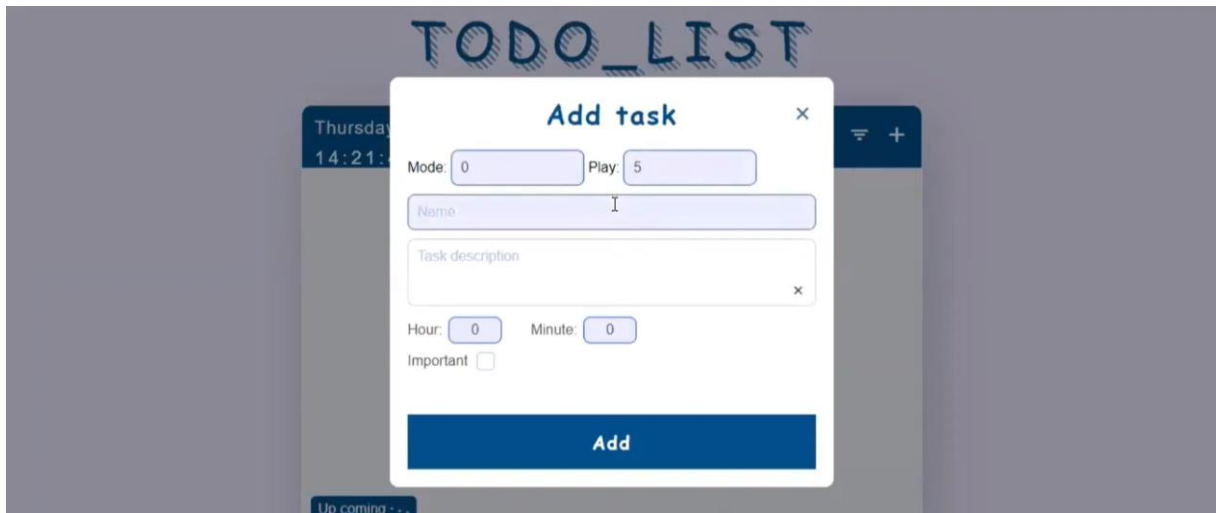
Kết quả đạt được:

- Có thể dễ dàng điều chỉnh âm lượng của âm thanh một cách thuận tiện và chính xác bằng Rotary Encoder.
- Có thể lưu trữ các tệp ghi âm vào thẻ nhớ thông qua MicroSD Adapter, giúp người dùng dễ dàng quản lý và phân loại các bản ghi âm.
- Giúp chúng ta thu âm âm thanh với độ nhạy cao và chất lượng âm thanh tốt, kết hợp với mã hoá âm thanh trên MAX98357A, chúng ta có thể phát lại âm thanh đã ghi với chất lượng cao thông qua loa.
- Giao tiếp với màn hình ILI9341 thông qua chạm cho phép người dùng dễ dàng tương tác để thêm lịch trình và điều chỉnh các thiết lập âm thanh.

4.1.2. Kết quả mô hình module thực tế

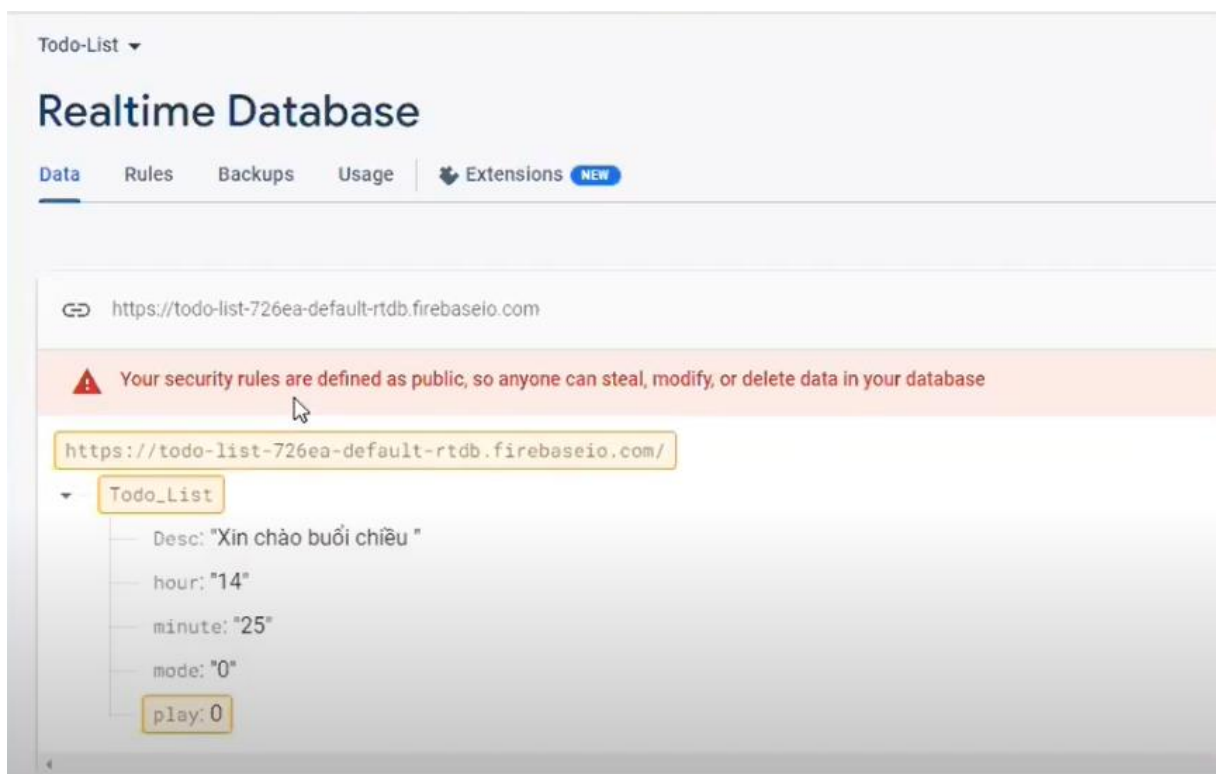


Hình 4.1. Module thực tế



Hình 4.2. Giao diện web để cập nhật lịch trình và thiết lập thời gian

4.1.3. Kết quả gửi dữ liệu lên Google Firebase



Hình 4.3. Kết quả hiển thị trên Google Firebase

4.2. Nhận xét và đánh giá chung

Đề tài của chúng em có thể rất hữu ích trong đời sống thực tế, giúp người dùng có thể quản lý được thời gian biểu. Tuy nhiên, hệ thống vẫn còn nhiều hạn chế nên chưa tối ưu và còn nhiều yêu cầu chưa đáp ứng được, cần có thêm thời gian nghiên cứu và chi phí để hoàn thiện.

Một số hạn chế nhóm chúng em nhận thấy trong lúc thực hiện đề tài như sau:

- Chưa tối ưu được bộ nhớ của vi điều khiển ESP32.
- Module INMP441 thỉnh thoảng tín hiệu ghi âm sẽ không rõ.
- Chưa tối ưu được hệ thống, vẫn phải sử dụng Firebase để cập nhật dữ liệu.

CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Từ những hạn chế trong quá trình thực hiện đề tài, nhóm đưa ra một số ý tưởng phát triển của hệ thống, để có thể hoàn chỉnh và hoạt động hiệu quả hơn:

- Thay vì sử dụng web thì có thể viết một ứng dụng để người dùng có thể giao tác đơn giản hơn.
- Tích hợp các dịch vụ đám mây: Kết nối với các dịch vụ đám mây như Google Drive hoặc Dropbox để tự động sao lưu và chia sẻ các file ghi âm.
- Tích hợp điều khiển giọng nói: Phát triển khả năng điều khiển âm thanh thông qua giọng nói. Điều này có thể đòi hỏi việc sử dụng các công nghệ nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên.
- Tích hợp tính năng xử lý tín hiệu âm thanh: Nghiên cứu và triển khai các thuật toán xử lý tín hiệu âm thanh để tăng cường chất lượng âm thanh, như echo cancellation, noise reduction, hoặc audio effects.

TÀI LIỆU THAM KHẢO

- [1]. [Online] FPT CLOUD, Firebase là gì? Ưu nhược điểm & Các dịch vụ của Firebase. Publish: 31st December, 2021. Link: <https://fptcloud.com/firebase-la-gi/>
- [2]. [Online] InvenSense, Omnidirectional Microphone with Bottom Port and I2 S Digital Output INMP441. Link: <https://datasheetspdf.com/pdf-file/1350167/InvenSense/INMP441/1>
- [3]. [Online] ILITEK, a-Si TFT LCD Single Chip Driver 240RGBx320 Resolution and 262K color. Publish: 14th July, 2022. Link: <https://datasheetspdf.com/pdf-file/769801/ILITEK/ILI9341/1>
- [4]. [Online] HSHOP, Màn Hình LCD TFT Cảm Ứng Điện Trở 2.8 Inch ILI9341 Giao Tiếp SPI. Publish: 6th June, 2020. Link: <https://hshop.vn/products/man-hinh-lcd-tft-cam-ung-dien-tro-2-8-inch-ili9341-giao-tiep-spi>
- [5]. [Online] COMPONENTS 101, Micro SD Card Adapter Module. Publish: 11th June, 2021. Link: <https://components101.com/modules/micro-sd-card-module-pinout-features-datasheet-alternatives>
- [6]. [Online] FMUSER, Giới thiệu về SPI, I2C, UART, I2S, GPIO, SDIO, CAN. Publish: 15th September, 2021. Link: <https://vi.fmuser.org/news/IPTV-encoder/Introduction-to-SPI-I2C-UART-I2S-GPIO-SDIO-CAN/>
- [7]. [Online] DroneBot Workshop, ESP32 Sound - Working with I2S. Publish: 22nd May, 2022. Link: <https://www.youtube.com/watch?v=m-MPBjScNRk&t=2362s>
- [8]. [Online] That Project, ESP32 | INMP441 | Tutorial - [Part.4] Capturing audio from i2s mic to save WAV file (I2S interface). Publish: 17th May, 2020. Link: <https://www.youtube.com/watch?v=qmruNKeIN-o&t=950s>
- [9]. [Online] AFYU, I2S MAX98357 Bluetooth Speaker project for ESP32 microcontroller / Programming Arduino IDE. Publish: 15th February, 2023. Link: <https://www.youtube.com/watch?v=24pk28oqVQI&t=43s>