ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH

SINH VIÊN THỰC HIỆN

<LÊ THÀNH LỢI> – <22520799>

<ĐÀO TIẾN HÙNG> – <22520499>

BÁO CÁO ĐỒ ÁN MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN

<GAME TETRIS>

GIẢNG VIÊN HƯỚNG DẪN

<PHẠM MINH QUÂN>

TP. HỒ CHÍ MINH, <2024>

LỜI MỞ ĐẦU

Ngày nay, sự phát triển mãnh mẽ của cuộc cách mạng công nghiệp 4.0, cuộc sống của con người chúng ta đã thay đổi ngày một tốt hơn, với những trang thiết bị phục vụ cho công cuộc công nghiệp hóa – hiện đại hóa. Để góp phần vào điều đó, không thể không nhắc đến những đóng góp từ những hệ thống nhúng và công nghệ internet vạn vật. Từ những ngày đầu phát triển, các vi xử lý – vi điều khiển đã cho thấy sự ưu việt của nó, và đến ngày nay, điều này còn được khẳng định thêm. Những thành tựu của nó đã biến những điều tưởng chừng như không thể trước đây thành hiện thực ở hiện tại, góp phần giúp đời sống con người được cải thiện và nâng cao.

Để góp phần làm sáng tỏ hiệu quả của những ứng dụng trong thực tế của môn vi xử lý – vi điều khiển, sau một thời gian học tập, được thầy giảng dạy và cung cấp kiến thức, cùng với sự nổ lực của bản thân, chúng em đã *"Thiết kế máy chơi game tetris"*. Do thời gian, kiến thức và kinh nghiệm còn có hạn nên sẽ không tránh khỏi những sai sót. Chúng em rất mong nhận được sự giúp đỡ và tham khảo ý kiến từ thầy cùng các bạn để có thể hoàn thiện đồ án.

LỜI CẢM ƠN

Trong suốt quá trình gian thực hiện đồ án, từ lúc bắt đầu đến lúc thực hiện thành công bản mô phỏng, chúng em đã gặp rất nhiều khó khăn, thử thách và được thầy Phạm Minh Quân chỉ bảo, giúp đỡ rất nhiều. Chúng em xin gửi lời cảm ơn sâu sắc và chân thành nhất đến thầy. Thầy đã hỗ trợ chúng em về mặt kiến thức cần thiết cũng như giải đáp tất cả các thắc mắc trong quá trình hiện đồ án của chúng em.

Với kinh nghiệm còn hạn chế, đồ án của chúng em không thể tránh khỏi được những thiếu sót. Chúng em rất mong nhận được những ý kiến của thầy để có thể bổ sung, hoàn thiện hơn ở đồ án cuối kỳ, cũng như bổ sung, nâng cao kiến thức của mình.

Chúng em xin chân thành cảm ơn.

MỤC LỤC

DANH MỤC HÌNH

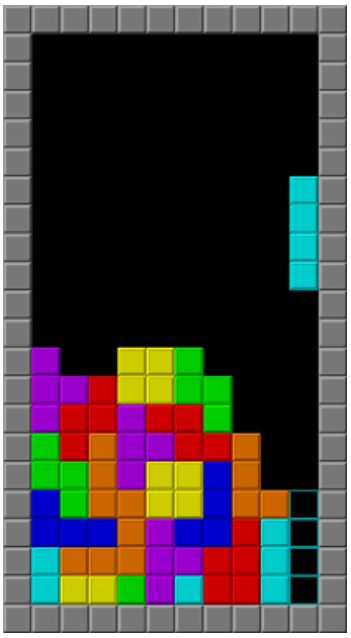
DANH MỤC BẢNG

Chương 1.TỔNG QUAN

1.1. Giới thiệu về Tetris.

1.1.1. Tổng quan về Tetris

Tetris (tiếng Nga: Тетрис) [1], hay còn gọi là trò chơi xếp hình, là một trò chơi điện tử đầu tiên được thiết kế và phát triển bởi nhà khoa học máy tính người Liên Xô Alexey Pajitnov. Trò chơi được tạo ra vào ngày 6 tháng 6 năm 1984.



Hình 1.1: Một màn hình game Tetris điển hình

Trò chơi có bảy loại khối hình: I (thẳng đứng), J, L, O (vuông), S, T, Z. Ta thấy mỗi khối gạch được cấu tạo từ 4 hình vuông nhỏ xếp lại với nhau. Ta có thể coi các khối gạch đó như là những hình chữ nhật có kích thước khác nhau.

Các hình khác được tạo ra khi xoay các khối cơ bản này các góc tương ứng 90 độ, 180 độ, 270 độ.

Một chuỗi ngẫu nhiên của Tetriminos rơi xuống sân chơi (một trục đứng hình chữ nhật, được gọi là "tốt" hay "ma trận").

1.1.2. Luật chơi của Tetris

Mục tiêu của trò chơi là di chuyển các khối gạch đang rơi từ từ xuống trong kích thước hình chữ nhật 24 hàng x 10 cột (trên màn hình). Chỗ nào có gạch rồi thì không di chuyển được tới vị trí đó. Người chơi xếp những khối hình sao cho khối hình lấp đầy 1 hàng ngang để ghi điểm và hàng ngang ấy sẽ biến mất.

Ta có hai bảng:

- Một bảng chính gồm 18 dòng và 10 cột. Là màn hình hiển thị trò chơi
- Một bảng Next thể hiện những khối gạch tiếp theo sẽ được cho vào màn chơi khi khối gạch hiện tại đã đặt xong.

Một nhóm 4 khối sẽ rơi từ phía trên cùng của màn hình, di chuyển các khối và xoay chúng cho đến khi chúng rơi xuống phía dưới cùng của màn hình, sau đó nhóm 4 khối tiếp theo sẽ rơi xuống.

Nếu để cho những khối hình cao quá màn hình, trò chơi sẽ kết thúc.

Trò chơi kết thúc khi khối gạch không rơi xuống được nữa.

Tất cả các Tetriminos có khả năng hoàn thành một và hai dòng. J, L có thể có ba. Chỉ có Tetrimino chữ I có khả năng để xóa bốn dòng cùng một lúc, và điều này được gọi là một "Tetris". Xóa nhiều nhất chỉ được 4 hàng/1 lần.

Phím tắt:

- Phím mũi tên lên: xoay khối.
- Phím mũi tên trái: di chuyển sang trái.
- Phím mũi tên phải: di chuyển sang phải.
- Phím mũi tên xuống: tăng tốc độ rơi.

1.2. Lý do chọn đề tài

Tetris là một trong những trò chơi điện tử kinh điển, được nhiều người yêu thích và dễ dàng tiếp cận. Việc phát triển một máy chơi game Tetris sử dụng vi điều khiển không chỉ giúp chúng ta hiểu rõ hơn về nguyên lý hoạt động của vi điều khiển mà còn tạo ra một sản phẩm thực tiễn có tính ứng dụng cao. Đề tài này được chọn vì các lý do sau:

Khả năng học hỏi và phát triển kỹ năng: Làm việc với vi điều khiển giúp sinh viên nắm bắt và thành thạo kỹ năng lập trình và điều khiển vi xử lý, từ đó củng cố

kiến thức lý thuyết đã học. Việc thực hành lập trình trên vi điều khiển cho phép sinh viên áp dụng các kiến thức về lập trình, quản lý tài nguyên hệ thống, và xử lý giao diện. Hơn nữa, sinh viên sẽ học được cách tích hợp phần cứng và phần mềm, điều khiển các thiết bị ngoại vi và xử lý dữ liệu từ các cảm biến, từ đó nâng cao kỹ năng thực hành và khả năng giải quyết vấn đề.

Tính thực tiễn và ứng dụng cao: Tạo ra một sản phẩm cụ thể như máy chơi game Tetris giúp sinh viên hiểu rõ hơn về quy trình thiết kế, lắp ráp và lập trình phần cứng. Qua đó, sinh viên có cơ hội thực hành các bước thiết kế hệ thống điện tử, từ việc vẽ sơ đồ mạch điện, chọn lựa các linh kiện phù hợp, đến việc lắp ráp và kết nối chúng. Bên cạnh đó, việc phát triển phần mềm điều khiển game Tetris trên vi điều khiển giúp sinh viên hiểu sâu hơn về logic game, quản lý trạng thái và xử lý sự kiện trong game.

Khả năng mở rộng và phát triển: Dự án này có thể được mở rộng thêm các tính năng mới, từ đó tạo điều kiện cho sự sáng tạo và phát triển thêm nhiều kỹ năng khác.

1.3. Mục tiêu nghiên cứu

Ứng dụng các kiến thức đã học trong môn Vi xử lý – vi điều khiển và kiến thức tìm hiểu về vi điều khiển Esp32, từ đó xây dựng máy game Tetris. Cụ thể hơn:

- Thiết kế và xây dựng phần cứng của máy chơi game.
- Lập trình phần mềm điều khiển và logic game Tetris trên vi điều khiển.
- Tích hợp phần cứng và phần mềm để máy chơi game hoạt động trơn tru.
- Đánh giá và thử nghiệm hiệu suất của máy chơi game, từ đó rút ra các kinh nghiệm và kiến thức hữu ích.

1.4. Nội dung thực hiện

- *Khảo sát và nghiên cứu:* Tìm hiểu về cấu trúc và nguyên lý hoạt động của game Tetris, các con vi điều khiển, và các thành phần phần cứng cần thiết.
- *Thiết kế hệ thống:* Vẽ sơ đồ mạch điện, thiết kế giao diện và cấu trúc tổng thể của máy chơi game.
- *Phát triển phần mềm:* Viết code điều khiển vi điều khiển, phát triển logic game Tetris, và tạo giao diện người dùng.

- *Lắp ráp và tích hợp:* Lắp ráp các thành phần phần cứng, kết nối và tích hợp với phần mềm đã phát triển.
- *Kiểm tra và hiệu chỉnh:* Chạy thử nghiệm máy chơi game, kiểm tra lỗi, hiệu chỉnh và tối ưu hóa hệ thống.

1.5. Phương thức thực hiện

- Thiết kế sơ đồ hệ thống.
- Sử dụng vi điều khiển Esp32 để điều khiển, lập trình bằng ngôn ngữ C trên Wokwi, Arduino IDE.
- Sử dụng lý thuyết về ngắt ngoài để cài đặt tính năng nút nhấn thao tác với game.
- Sử dụng Oled để hiện thị giao diện thao tác với người dùng của game.
- Sử dụng nguyên lý áp điện để điều khiển Buzzer phát ra âm thanh.

1.6. Giới hạn đề tài.

1.6.1. Giới hạn về phần cứng

- *ESP32:* Sử dụng vi điều khiển ESP32 làm trung tâm điều khiển. Các tính năng chính của ESP32 bao gồm Wi-Fi, Bluetooth, và nhiều cổng GPIO, nhưng trong phạm vi đề tài này, sẽ chỉ sử dụng các tính năng cần thiết cho việc điều khiển game Tetris.
- *Màn hình hiển thị:* Sử dụng màn hình OLED để hiển thị game Tetris. Màn hình OLED có kích thước nhỏ gọn, độ phân giải phù hợp cho việc hiển thị các khối hình và giao diện người chơi của trò chơi.
- *Nút bấm điều khiển:* Sử dụng các nút bấm vật lý để điều khiển các thao tác trong game như di chuyển các khối hình, xoay khối, và thả khối nhanh. Số lượng nút bấm sẽ được giới hạn để đảm bảo tính đơn giản và dễ sử dụng.
- *Nguồn điện:* Sử dụng nguồn điện từ pin hoặc nguồn USB để cung cấp năng lượng cho hệ thống. Điều này giúp đảm bảo tính di động của máy chơi game.

1.6.2. Giới hạn về phần mềm

- *Ngôn ngữ lập trình:* Sử dụng ngôn ngữ C/C++ để lập trình ESP32. Các thư viện hỗ trợ như Arduino IDE sẽ được sử dụng để đơn giản hóa quá trình lập trình và điều khiển các thiết bị ngoại vi.

- *Logic game Tetris:* Xây dựng logic game cơ bản của Tetris, bao gồm các thao tác như di chuyển khối, xoay khối, và kiểm tra các dòng đã hoàn thành để xóa. Sẽ không phát triển các tính năng phức tạp hơn như chế độ chơi đa người, cấp độ khó khác nhau, hoặc các hiệu ứng đặc biệt.
- *Giao diện người dùng:* Giao diện đơn giản, tập trung vào hiển thị các khối hình và thông tin cơ bản như điểm số và cấp độ. Không tập trung vào phát triển giao diện đồ họa phức tạp.

1.6.3. Giới hạn về chức năng

- *Chức năng cơ bản:* Đảm bảo máy chơi game Tetris có thể hoạt động một cách mượt mà với các chức năng cơ bản của trò chơi. Không phát triển các tính năng nâng cao như lưu trữ điểm số trực tuyến, kết nối mạng, hoặc chơi đa người.
- *Tính di động:* Thiết kế máy chơi game có kích thước nhỏ gọn, dễ mang theo và sử dụng. Điều này có nghĩa là sẽ giới hạn việc sử dụng các linh kiện và phụ kiện phức tạp hoặc có kích thước lớn.
- *Khả năng mở rộng:* Trong phạm vi đề tài này, sẽ tập trung vào việc hoàn thiện các chức năng cơ bản. Các tính năng mở rộng sẽ được đề xuất như những hướng phát triển tương lai, không bắt buộc phải thực hiện trong phạm vi đề tài hiện tại.

1.6.4. Giới hạn về thời gian và nguồn lực

- *Thời gian thực hiện:* Đề tài được thực hiện trong khoảng thời gian giới hạn của học kỳ, do đó cần phải có kế hoạch cụ thể và chi tiết để đảm bảo hoàn thành đúng hạn. Sẽ không có đủ thời gian để phát triển và kiểm tra tất cả các tính năng mở rộng.
- *Nguồn lực:* Sử dụng các linh kiện và công cụ có sẵn hoặc dễ dàng mua được với chi phí hợp lý. Không sử dụng các thiết bị đắt tiền hoặc khó tìm.

Chương 2.CƠ SỞ LÝ THUYẾT

2.1. PWM

2.1.1. Định nghĩa

PWM (Pulse Width Modulation) [2] là một kỹ thuật điều chỉnh dòng điện hoặc điện áp cung cấp cho tải bằng cách bật và tắt tín hiệu một cách nhanh chóng. Giá trị trung bình của tín hiệu điện áp hoặc dòng điện được điều chỉnh bằng cách thay đổi tỷ lệ giữa thời gian tín hiệu bật (xung cao) và thời gian tín hiệu tắt (xung thấp) trong một chu kỳ cố định. Tỷ lệ này được gọi là "duty cycle".

2.1.2. Nguyên lý hoạt động

PWM hoạt động dựa trên việc tạo ra một chuỗi các xung điện với độ rộng xung thay đổi để điều chỉnh mức năng lượng trung bình cung cấp cho tải. Nguyên lý cơ bản của PWM có thể được giải thích qua hai thông số chính:

- *Tần số (Frequency):* Số lượng chu kỳ PWM diễn ra trong một giây. Tần số cao hơn thường giúp làm mịn tín hiệu đầu ra nhưng có thể yêu cầu phần cứng phức tạp hơn.
- *Chu kỳ hoạt động (Duty Cycle):* Tỷ lệ phần trăm của một chu kỳ mà tín hiệu PWM là cao. Duty cycle có thể thay đổi từ 0% (tín hiệu luôn ở mức thấp) đến 100% (tín hiệu luôn ở mức cao).

Công thức tính giá trị trung bình của điện áp đầu ra (V_avg) từ tín hiệu PWM như sau:

$V_{avg} = V_{max} \times \text{Duty Cycle}$

Trong đó, V_{max} là điện áp tối đa của tín hiệu PWM.

2.1.3. Ứng dụng

PWM được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau nhờ vào tính linh hoạt và hiệu quả của nó. Dưới đây là một số ứng dụng chính:

- *Điều khiển động cơ:* PWM được sử dụng để điều khiển tốc độ và hướng quay của động cơ DC. Bằng cách thay đổi duty cycle, có thể điều chỉnh lượng năng lượng trung bình cung cấp cho động cơ, từ đó điều chỉnh tốc độ quay.

- *Điều khiển độ sáng LED:* Đối với đèn LED, PWM có thể điều chỉnh độ sáng bằng cách thay đổi duty cycle. LED sẽ nhận được các xung điện nhanh, và mắt người sẽ nhìn thấy độ sáng trung bình dựa trên duty cycle.
- *Truyền dữ liệu:* Trong các hệ thống truyền thông số, PWM có thể được sử dụng như một phương pháp điều chế tín hiệu. Các giá trị số có thể được biểu diễn bằng các chuỗi xung có độ rộng khác nhau.
- *Điều khiển nhiệt độ:* PWM có thể được sử dụng để điều khiển nhiệt độ trong các hệ thống sưởi ấm bằng cách điều chỉnh năng lượng cung cấp cho bộ phận làm nóng.
- *Nguồn cung cấp xung (Switched-mode power supply):* Trong các bộ nguồn chuyển mạch, PWM được sử dụng để điều chỉnh điện áp đầu ra bằng cách điều khiển thời gian bật/tắt của các thiết bị chuyển mạch.

2.2. I2C

2.2.1. Định nghĩa

I2C (Inter-Integrated Circuit) [3] là một chuẩn giao tiếp truyền dữ liệu nối tiếp đa thiết bị (multi-master, multi-slave) được sử dụng để kết nối các vi điều khiển và các thiết bị ngoại vi khác. I2C được phát triển bởi Philips Semiconductor (nay là NXP Semiconductors) vào những năm 1980.

2.2.2. Nguyên lý hoạt động

I2C sử dụng hai dây để truyền dữ liệu:

- SDA (Serial Data Line): Dây truyền dữ liệu.
- SCL (Serial Clock Line): Dây đồng hồ.

Các thiết bị trên bus I2C được phân loại thành hai loại chính: master và slave. Một thiết bị master điều khiển quá trình truyền dữ liệu, và các thiết bị slave nhận và phản hồi dữ liệu. Một hệ thống I2C có thể có nhiều master và nhiều slave.

2.2.3. Ứng dụng

I2C được sử dụng rộng rãi trong nhiều thiết bị điện tử nhờ tính linh hoạt và khả năng kết nối nhiều thiết bị chỉ với hai dây. Dưới đây là một số ứng dụng chính:

- *Kết nối Cảm biến:* I2C thường được sử dụng để kết nối các cảm biến với vi điều khiển. Ví dụ: cảm biến nhiệt độ, cảm biến áp suất, và cảm biến gia tốc.

- *Giao tiếp giữa các IC:* I2C được sử dụng để kết nối các IC với nhau trong các thiết bị điện tử phức tạp như điện thoại di động, máy tính và thiết bị nhúng.
- *Điều khiển Hiển thị:* I2C thường được sử dụng để điều khiển các màn hình LCD, OLED và các loại hiển thị khác.
- *Thiết bị Lưu trữ:* EEPROM và các loại bộ nhớ flash thường sử dụng giao tiếp I2C để lưu trữ và truy xuất dữ liệu.
- *Thiết bị Âm thanh:* Các bộ giải mã âm thanh, điều khiển âm lượng và các thiết bị âm thanh khác sử dụng I2C để giao tiếp với vi điều khiển.

Chương 3.LINH KIỆN SỬ DỤNG

3.1. Thiết bị điều khiển

3.1.1. Vi xử lý sử dụng

Sau nhiều lần chọn lọc, so sánh các lợi ích mà mỗi vi xử lý mạng lại, nhóm đã chọn dùng vi xử lý ESP32 [4]. Vi xử lý này dễ dùng và thích hợp với nhu cầu sử dụng nhất. Vi xử lý có những điểm mạnh như:

I. Hiệu Năng Mạnh Mẽ

ESP32 là một vi xử lý mạnh mẽ với khả năng xử lý đa nhiệm và tốc độ nhanh, rất phù hợp để phát triển các ứng dụng yêu cầu xử lý phức tạp như game Tetris. Với CPU dual-core, ESP32 có thể xử lý nhiều tác vụ cùng lúc, đảm bảo trò chơi hoạt động mượt mà và đáp ứng nhanh chóng

II. Khả Năng Lập Trình Linh Hoạt

ESP32 hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau như Arduino, MicroPython và ESP-IDF, mang lại sự linh hoạt trong phát triển phần mềm. Điều này giúp chúng em dễ dàng lựa chọn công cụ phát triển phù hợp với kỹ năng và yêu cầu của dự án.

III. Hỗ Trợ Đa Dạng Cổng Giao Tiếp

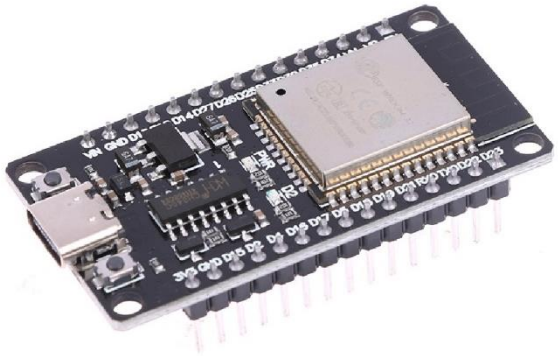
ESP32 cung cấp nhiều cổng giao tiếp như GPIO, SPI, I2C, UART, ADC, và DAC. Điều này rất quan trọng trong việc kết nối với các linh kiện ngoại vi như màn hình LED, nút bấm và loa, giúp xây dựng một hệ thống hoàn chỉnh cho game Tetris.

IV. Chi Phí Thấp và Dễ Tiếp Cận

Giá thành hợp lý và sự phổ biến rộng rãi của ESP32 giúp chúng em dễ dàng tiếp cận và sử dụng, đồng thời cộng đồng hỗ trợ mạnh mẽ cung cấp nhiều tài liệu và ví dụ hữu ích.

3.1.2. Tổng quan về ESP32

ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth. Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.



Hình 3.1: *Vi điều khiển Esp32*

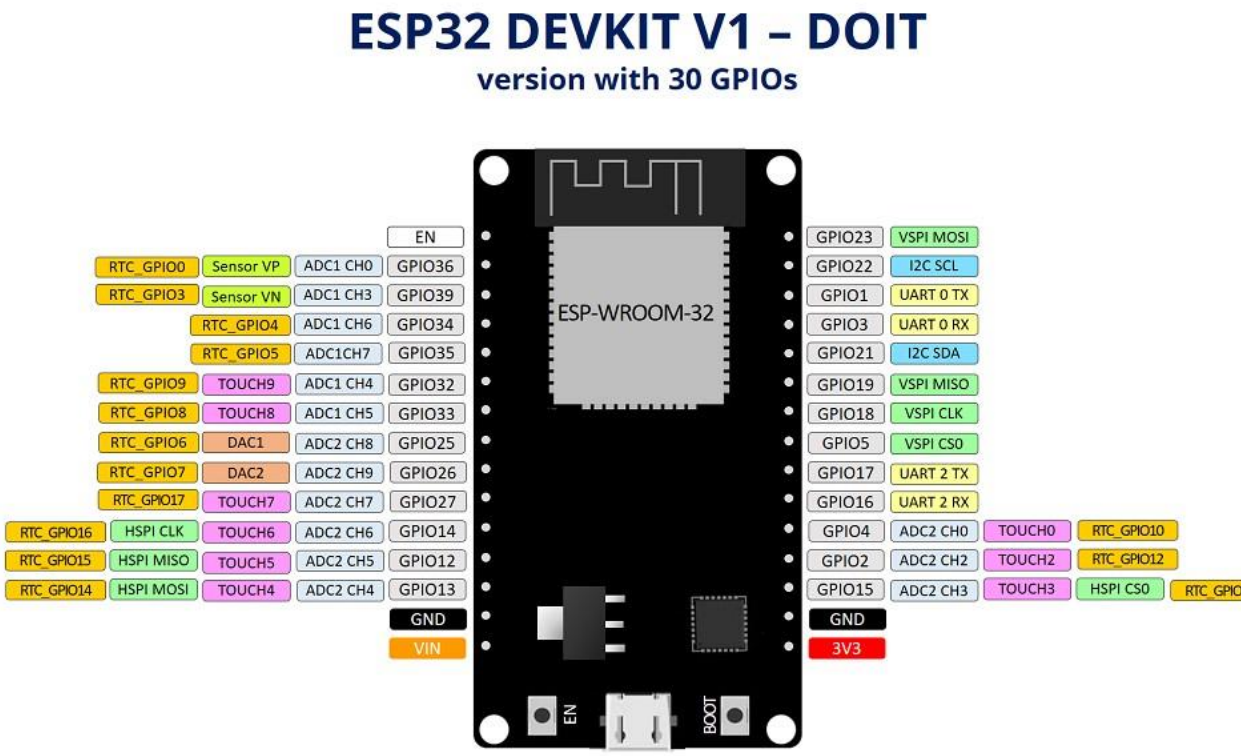
Nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng.

3.1.3. Các tính năng của ESP32

- Bộ xử lý:
  - CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH)
  - Bộ đồng xử lý (co-processor) công suất cực thấp (Ultra low power, viết tắt: ULP)
- Hệ thống xung nhịp: CPU Clock, RTC Clock và Audio PLL Clock
- Bộ nhớ nội: 448 KB bộ nhớ ROM và 520 KB bộ nhớ SRAM
- Kết nối không dây: Wi-Fi: 802.11 b/g/n và Bluetooth: v4.2 BR/EDR và BLE
- 34 GPIO pad vật lý
- Bảo mật:
  - Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WFA, WPA/WPA2 và WAPI.
  - Secure boot (tạm dịch: khởi động an toàn)
  - Mã hóa flash
  - 1024-bit OTP, lên đến 768-bit cho khách hàng
  - Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography
- Quản lý năng lượng:

- Bộ ổn áp nội với điện áp rơi thấp (internal low-dropout regulator)
- Miền nguồn riêng (individual power domain) cho RTC
- Dòng 5 µA cho chế độ deep sleep
- Trở lại hoạt động từ ngắt GPIO, timer, đo ADC, ngắt với cảm ứng điện dung

3.1.4. Sơ đồ chân ESP32

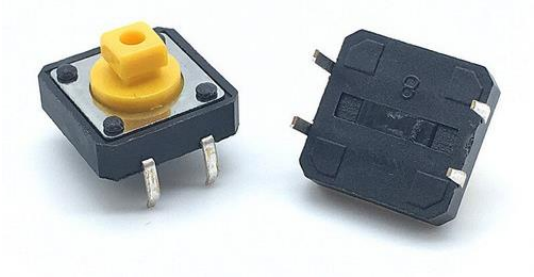


Hình 3.2: *Sơ đồ chân của Esp32-Devkit-V1 với 30 chân*

3.2. Thiết bị nhận tính hiệu

Sử dung nút bấm Tactile để điều khiển :

Điểm nổi bật của nút bấm Tactile là tính năng đáp ứng nhanh chóng khi được nhấn. Các nút bấm nhỏ và nhẹ của nút bấm Tactile được thiết kế để phản hồi ngay lập tức khi được nhấn.



Hình 3.3: *Push Button Tactile*

Dưới đây là các thông số kỹ thuật cơ bản của nút nhấn Tactile:

- Kích thước

  - *Đường kính nút:* 6mm

  - *Chiều cao:* 5mm

- Điện áp định mức: 12V DC
- Dòng điện định mức: Khoảng 50mA đến 100mA
- Điện trở tiếp xúc: Dưới 100 mΩ
- Điện trở cách điện: Trên 100 MΩ ở 500V DC
- Độ bền cơ học: Khả năng chịu được khoảng 100,000 đến 1,000,000 lần nhấn
- Lực nhấn: Khoảng 160gf ± 50gf (gram-force)
- Nhiệt độ hoạt động: Từ -20°C đến 70°C
- Vật liệu:

  - *Chân tiếp xúc:* làm bằng đồng hoặc hợp kim đồng, mạ bạc hoặc niken

  - *Thân nút:* nhựa chịu nhiệt

- Số chân: 4 chân
- Loại hoạt động: Momentary (khi nhấn giữ thì mạch được nối, khi thả ra thì mạch hở)
- Độ bền môi trường: Độ ẩm 85% RH ở 40°C trong 96 giờ

3.3. Thiết bị xuất tín hiệu

3.3.1. Màn hình OLED hiển thị

Màn hình OLED gồm những lớp như tấm nền, Anode, lớp hữu cơ, cathode. Và phát ra ánh sáng theo cách tương tự như đèn LED. Quá trình trên được gọi là phát lân quang điện tử. Những ưu điểm có thể kể đến trên màn hình OLED là những lớp hữu cơ nhựa mỏng, nhẹ mềm dẻo hơn những lớp tinh thể trên LED hay LCD nhờ vậy mà có thể ứng dụng OLED để chế tạo màn hình gập cuộn được. Độ sáng của OLED cũng tốt hơn LED và không cần đèn nền như trên LCD nên sử dụng pin ít hơn. Góc nhìn cũng cải thiện hơn những công nghệ tiền nhiệm, khoảng 170 độ.

Hình 3.4: *Màn hình OLED SSD1306*

Thông số kĩ thuật :

- Điện áp sử dụng: 3V3 đến 5V (DC)
- Công suất tiêu thụ: 0.04W
- Góc hiển thị: Lớn hơn 160 độ
- Độ phân giải của màn hình OLED: 128X64 pixel (Điểm ảnh)
- Độ rộng màn hình: 0.96inch
- Giao tiếp: I2C
- Driver: SSD1306

3.3.2. Buzzer phát âm thanh

Còi buzzer thụ động không có bộ dao động bên trong nên khi cấp một tần số từ 2khz – 5khz còi sẽ phát âm thanh thanh tùy theo tần số và thời gian, còi buzz thụ

động với những ưu điểm như: giá rẻ, tần số âm thanh có thể được kiểm soát giúp tạo ra nhiều hiệu ứng âm thanh.

Hình 3.5: *Buzzer thụ động*

Thông số kỹ thuật

- Điện áp: 5VDC
- Tần số hoạt động: 2Khz -5Khz
- Kích thước: 12*8.5mm
- Trọng lượng: 1g

3.4. Các linh kiện khác

3.4.1. Điện trở 10k Ohm

Điện trở 10k Ohm1/4W có kích thước nhỏ chiều dài chỉ 6.5mm, rộng 2mm, chân cắm 0,5mm. Nhiệt độ hoạt động từ -55oC đến 155oC và dải điện áp rộng thích hợp với nhiều mạch điện tử.

Hình 3.6: *Điện trở 10K Ohm*

Thông số kĩ thuật:

- Trị số: 10K Ohm
- Công suất: 1/4W (0,25W)
- Sai số: 5%
- Số vòng màu: 5

Công dụng:

- Điều chỉnh dòng điện và điện áp trong mạch điện tử
- Bảo vệ các linh kiện điện tử khỏi quá tải
- Tạo ra một điện áp hay dòng điện cần thiết
- Ghép nối các mạch điện tử với nhau

3.4.2. Transistor

Transistor A1015 là loại transistor thuận PNP đươc ứng dụng chủ yếu trong các mạch khuếch đại tần số âm thanh. Nó cũng có thể được sử dụng cho mục đích chuyển mạch giống như các bóng bán dẫn PNP khác. Khi sử dụng làm bộ khuếch đại cho mục đích chung tần số Âm thanh, có thể được vận hành trong vùng hoạt động.

Hình 3.7: *Transistor*

Thông số kỹ thuật :

- Loại Transistor: PNP
- Công suất: 0,4watts
- Điện áp định mức và dòng điện định mức: Vceo = 50v và Ic = 150mA

- Độ ồn thấp: 1dB
- Chất liệu vỏ: nhựa đúc
- Công suất tiêu tán: 400mW
- Mức tăng dòng điện một chiều: 400
- Collector Emitter và Collector Base điện áp sự cố: 50Vdc
- Điện áp bão hòa Emitter cơ sở: 1.1Vdc
- Điện áp cơ sở Emitter: 1.45Vdc
- Bổ sung cho 2SC1815 tạo thành 1 cặp thuận nghịch trong mạch đẩy kéo

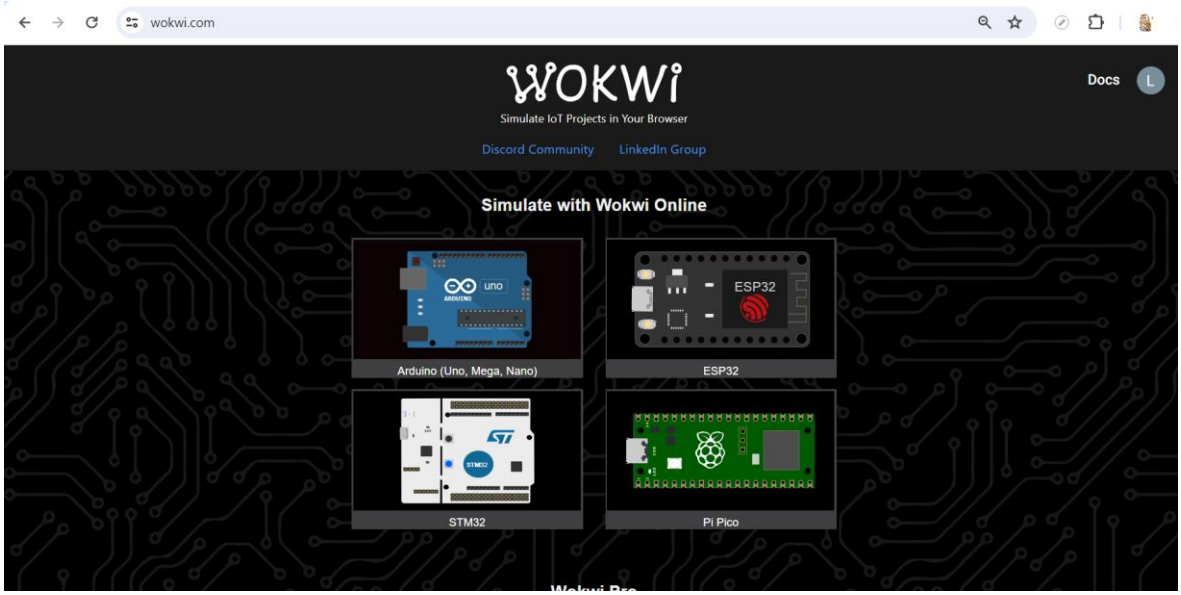
Nhiệt độ hoạt động và nhiệt độ lưu trữ: -55oC đến + 150oC

Chương 4.MÔ PHỎNG TRÊN PHẦN MỀM

4.1. Giới thiệu phần mềm woki

4.1.1. Wokwi là gì?

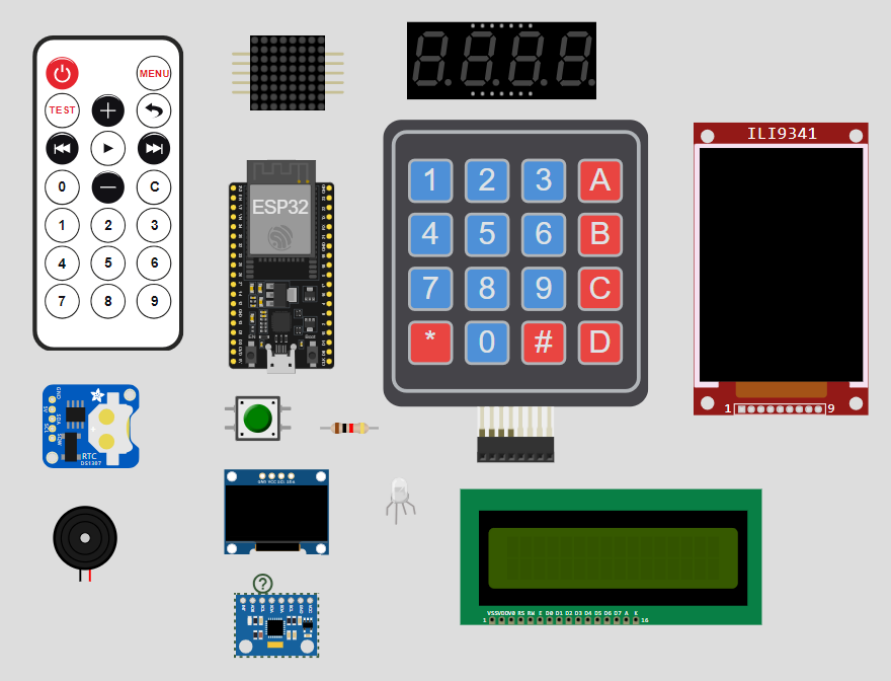
Wokwi [5] là một nền tảng trực tuyến cho phép người dùng mô phỏng và lập trình các dự án điện tử, đặc biệt là các dự án sử dụng vi điều khiển như Arduino, ESP32,... Nền tảng này cung cấp một môi trường ảo để thử nghiệm và phát triển các ứng dụng mà không cần phần cứng thực tế.



Hình 4.1: *Giao diện của Wokwi*

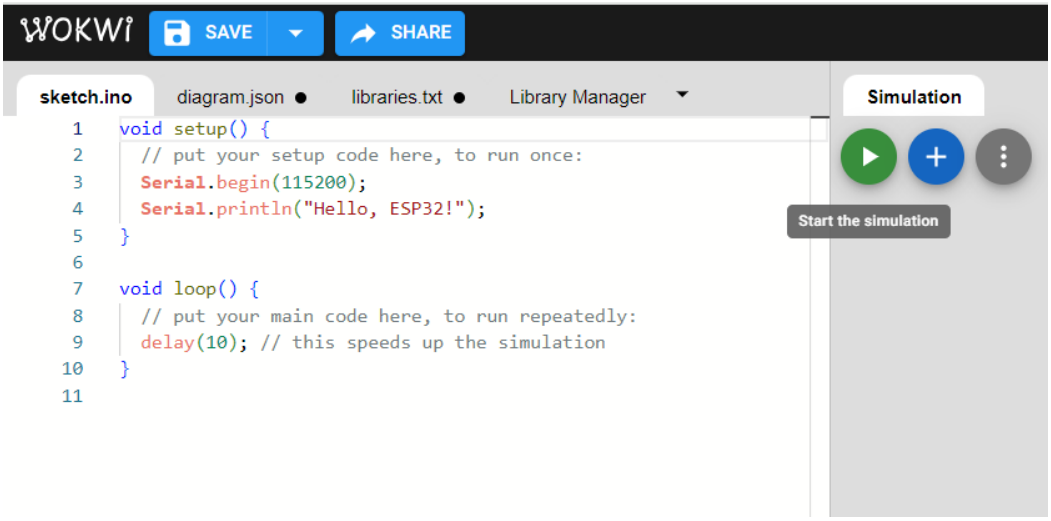
4.1.2. Các tính năng chính của Wokwi

- *Mô phỏng phần cứng:* Wokwi hỗ trợ mô phỏng nhiều loại vi điều khiển và các linh kiện điện tử khác nhau như đèn LED, nút nhấn, màn hình OLED, và các cảm biến. Điều này giúp người dùng kiểm tra và chạy thử các dự án một cách dễ dàng và nhanh chóng.



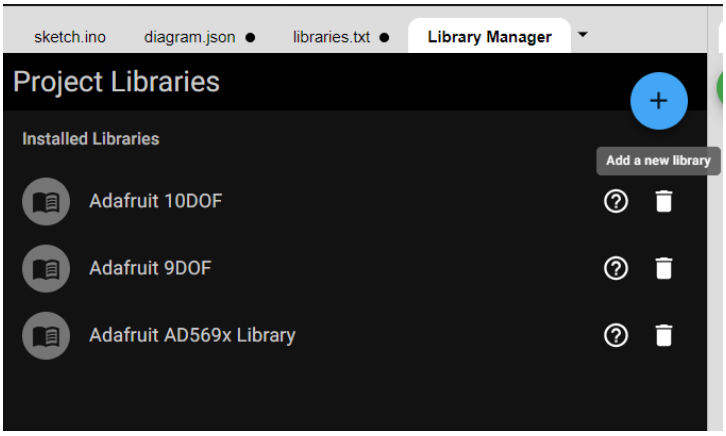
Hình 4.2: *Một số vi điều khiển và linh kiện được mô phỏng trên Wokwi*

- *Trình biên dịch trực tuyến:* Nền tảng cung cấp trình biên dịch trực tuyến cho phép người dùng viết, biên dịch và chạy mã nguồn trực tiếp trên trình duyệt mà không cần cài đặt phần mềm bổ sung.

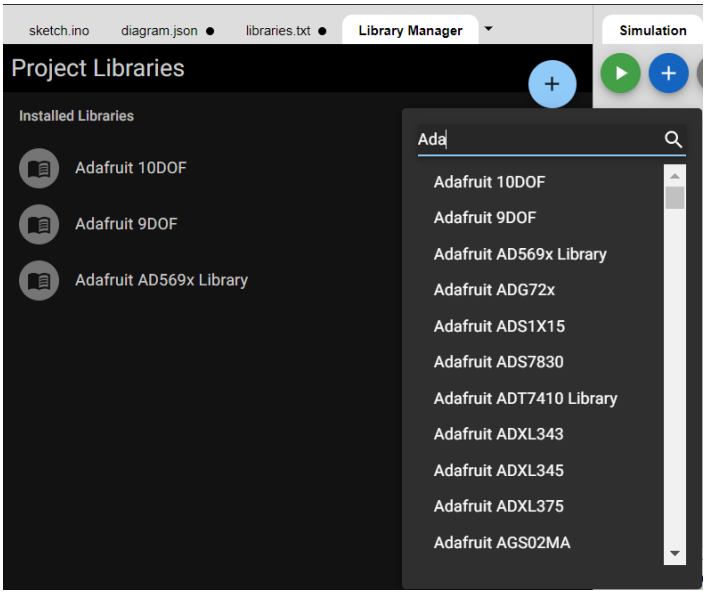


Hình 4.3: *Trình biên dịch được cung cấp bởi Wokwi*

- *Tích hợp thư viện:* Wokwi tích hợp sẵn nhiều thư viện hữu ích, giúp người dùng dễ dàng thêm các tính năng vào dự án của mình mà không cần cài đặt thủ công.

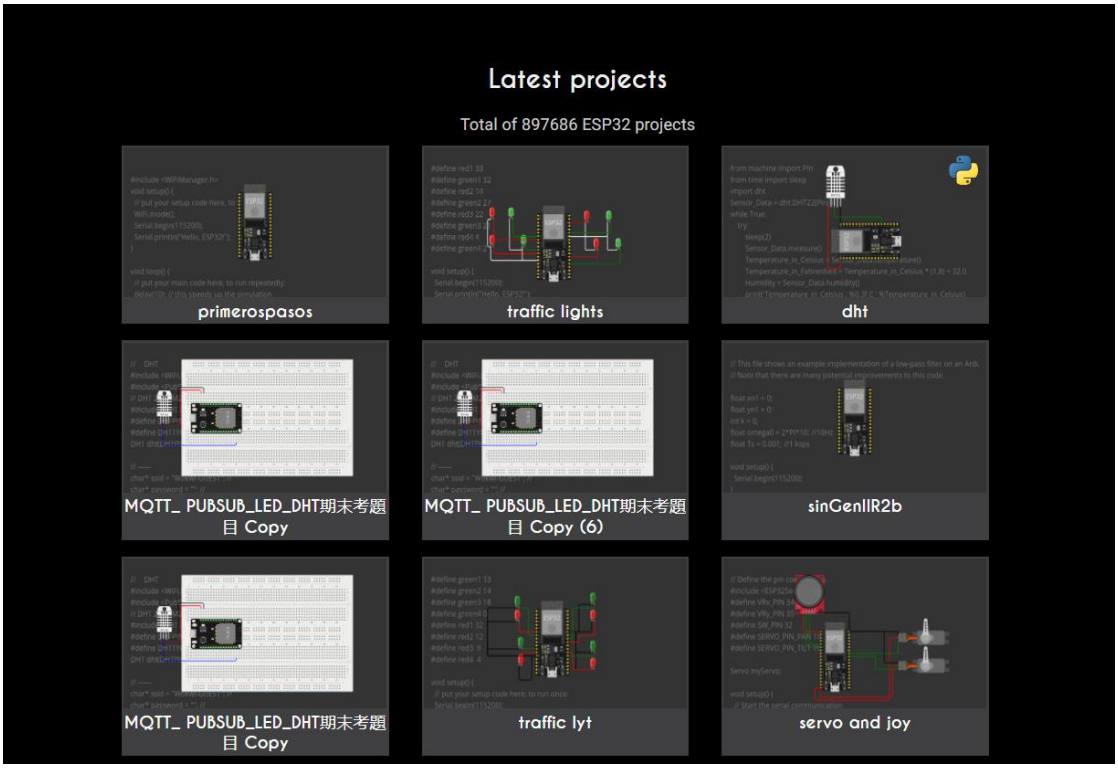


Hình 4.4: *Quản lý thư viện được tích hợp sẵn ở Wokwi*



Hình 4.5: *Thêm các thư viện được tích hợp sẵn ở Wokwi vào dự án*

- *Giao diện đồ họa:* Người dùng có thể kéo và thả các linh kiện vào một bảng mạch ảo, kết nối chúng và lập trình để thực hiện các chức năng mong muốn. Giao diện này thân thiện và trực quan, phù hợp cho cả người mới bắt đầu và các nhà phát triển chuyên nghiệp.
- *Chia sẻ dự án:* Wokwi cho phép người dùng chia sẻ các dự án của mình với cộng đồng. Mọi người có thể xem, sao chép và cải tiến dự án từ người khác, tạo nên một môi trường học tập và phát triển phong phú.



Hình 4.6: *Các dự án được chia sẻ trên Wokwi*

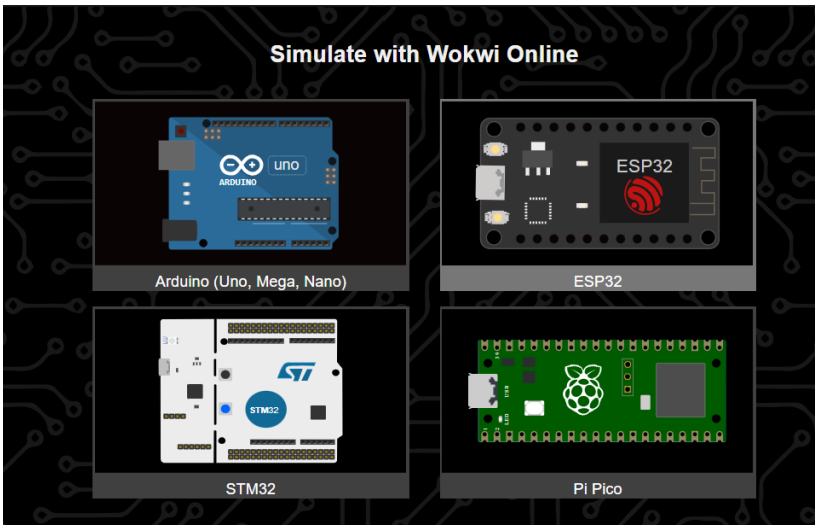
4.1.3. Lợi ích của việc sử dụng Wokwi

- *Tiết kiệm chi phí và thời gian:* Người dùng không cần mua các linh kiện phần cứng để thử nghiệm các ý tưởng của mình. Điều này giúp tiết kiệm chi phí và thời gian đáng kể, đặc biệt là trong giai đoạn nghiên cứu và phát triển ban đầu.
- *Học tập và giảng dạy:* Wokwi là công cụ hữu ích cho việc học tập và giảng dạy về điện tử và lập trình vi điều khiển. Sinh viên và giảng viên có thể sử dụng nền tảng này để thực hành và minh họa các khái niệm lý thuyết một cách sinh động và trực quan.
- *Phát triển nhanh chóng:* Với môi trường mô phỏng trực quan và tích hợp sẵn các thư viện, người dùng có thể nhanh chóng phát triển và kiểm tra các dự án của mình, rút ngắn thời gian phát triển.
- *Cộng đồng hỗ trợ:* Wokwi có một cộng đồng người dùng tích cực, nơi mọi người có thể chia sẻ kinh nghiệm, giải đáp thắc mắc và học hỏi lẫn nhau. Điều này tạo ra một môi trường học tập và phát triển liên tục.

4.2. Sơ đồ kết nối

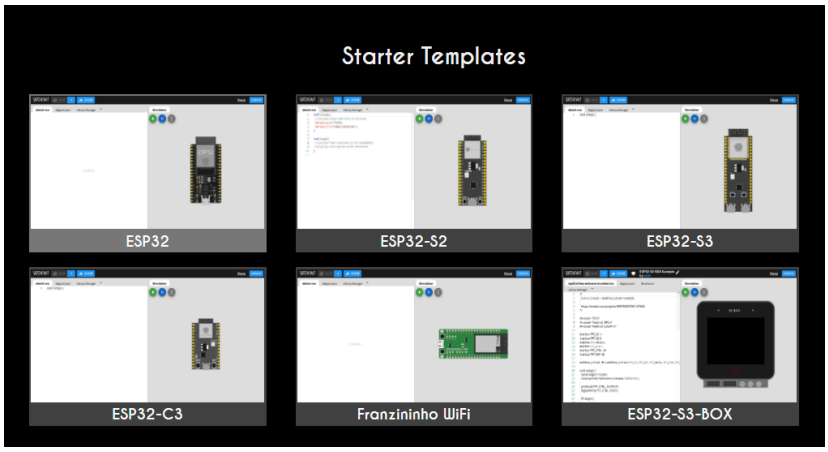
4.2.1. Tạo sơ đồ kết nối trên Wokwi

- Bước 1: Vào trang Wokwi.com, chọn ***ESP32*** ở mục ***Simulate with Wokwi Online***.



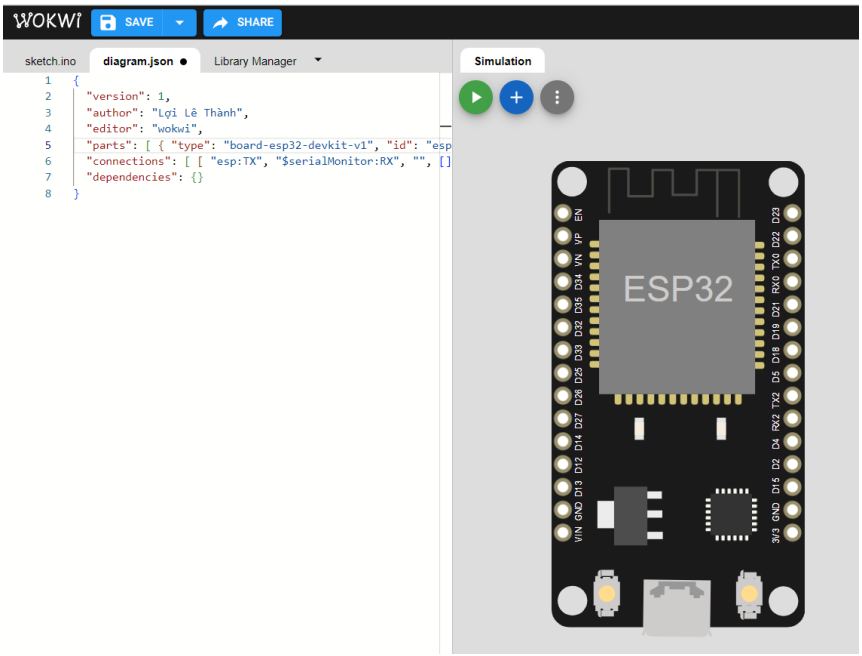
Hình 4.7: *Mục Simulate with Wokwi Online*

- Bước 2: Xuống mục ***Starter Templates*** chọn ***ESP32*** để mở giao diện mô phỏng dự án.



Hình 4.8: *Mục Starter Templates*

- Bước 3: Sang phần ***diagram.json***, sửa phần ***type*** từ *"board-esp32-devkit-c-v4"* sang *"board-esp32-devkit-v1"* để chuyển từ vi điều khiển Esp32-devkit-c-v4 sang vi điều khiển Esp32-devkit-v1.

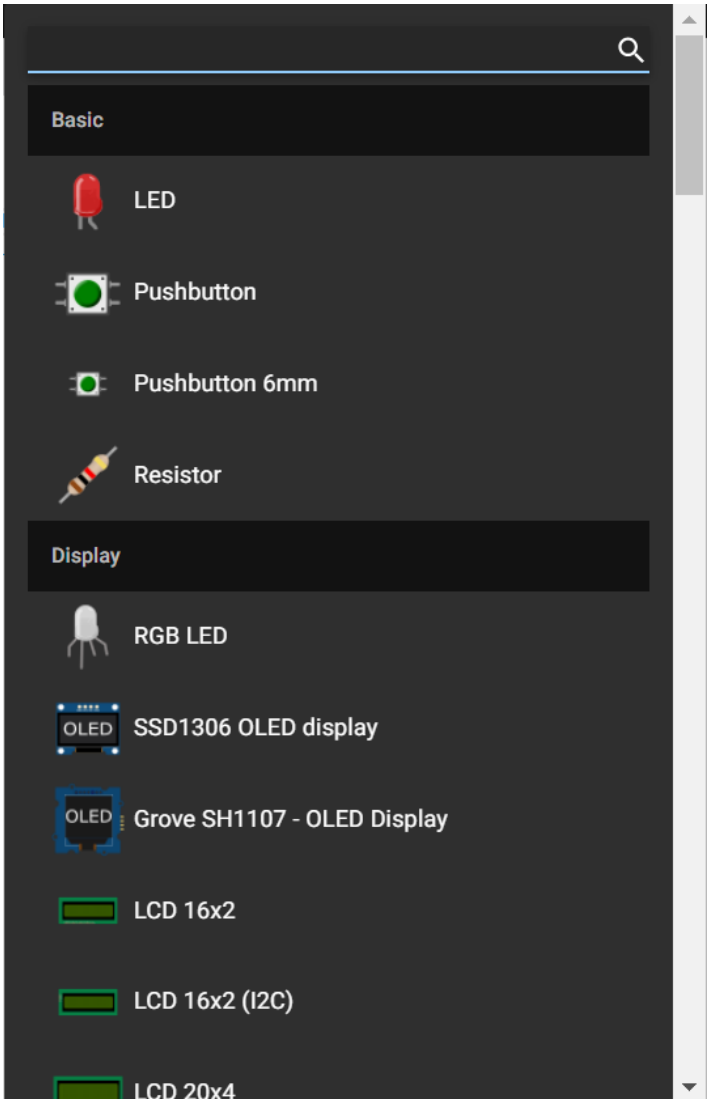


Hình 4.9: *Sửa type của Esp32 trong phần diagram.json*

- Bước 4: Thêm các linh kiện vào mô phỏng bằng cách nhấn ***Add a new part***(Nút cộng màn xanh).

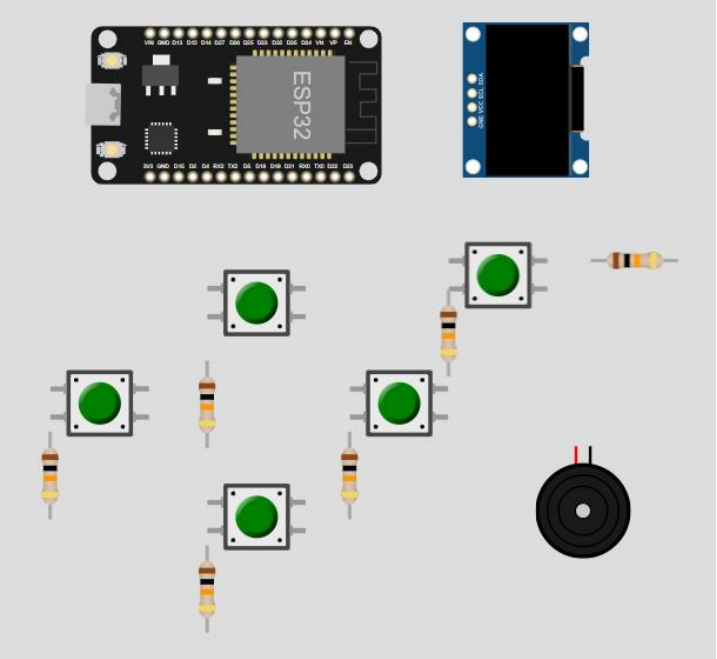
Hình 4.10*: Add a new part*

- Bước 5: Tìm kiếm các linh kiện cần thiết cho dự án như push button, oled ssd1306, buzzer, điện trở,.. rồi nhấn vào linh kiện đó với số lượng cần thiết để đưa vào mô phỏng.



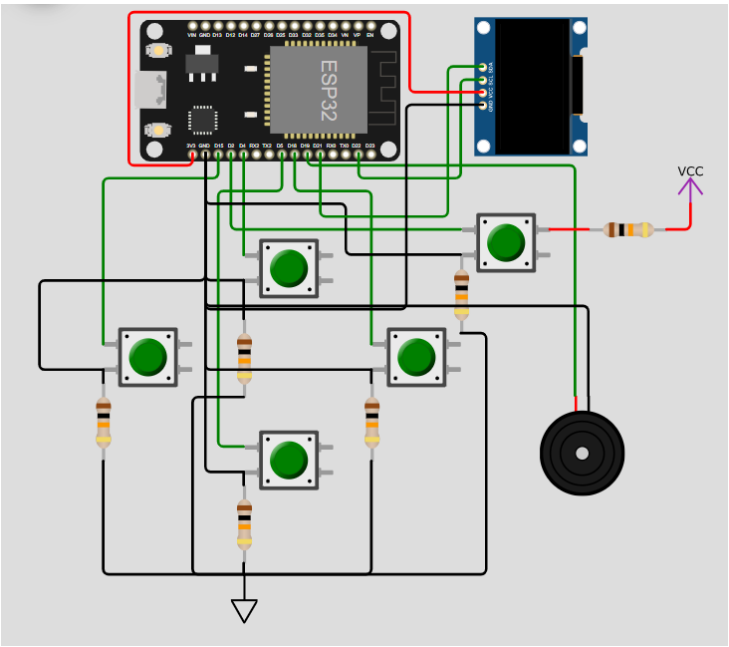
Hình 4.11: *Giao diện tìm kiếm linh kiện*

- Bước 6: Sắp xếp lại vi điều khiển và linh kiện trên mô phỏng.



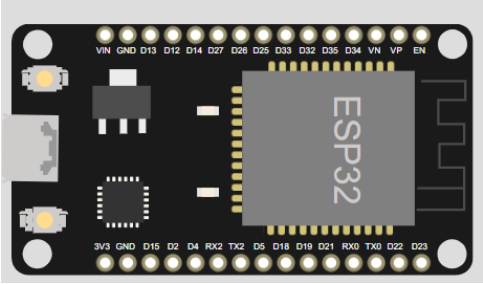
Hình 4.12: *Vi điều khiển và các linh kiện*

- Bước 7: Nối dây các linh kiện với vi điều khiển, nối các nút với điện trở. 5 nút được nói với các chân 4, 5, 15, 18, 2 của vi điều khiển Esp32 thực hiện lần lượt các chức năng của nút lên, xuống, trái, phải và reset. Nối đất các nút nhấn kèm thêm điện trở. Nối chân SCL của Oled với chân 22 của Esp32 và chân SDA của Oled với chân 21 của Esp32, nhằm thực hiện giao tiếp I2C với Oled SSD1306 của vi điều khiển Esp32. Nối chân VCC của Oled với chân 3V3 của vi điều khiển Esp32, nối chân GND của Oled với chân GND của vi điều khiển Esp32. Nối chân dương của Buzzer với chân 19 của vi điều khiển Esp32, nối chân âm của Buzzer với chân GND của vi điều khiển Esp32.

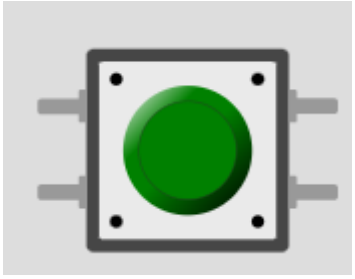


Hình 4.13: *Sơ đồ kết nối trên Wokwi*

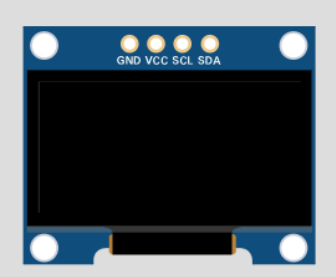
4.2.2. Các linh kiện mô phỏng trên Wokwi



Hình 4.14: *Esp32-devkit-v1 trên Wokwi*

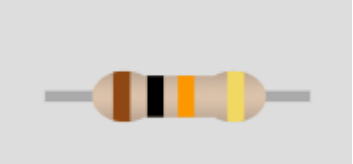


Hình 4.15*: Push Button trên Wokwi*



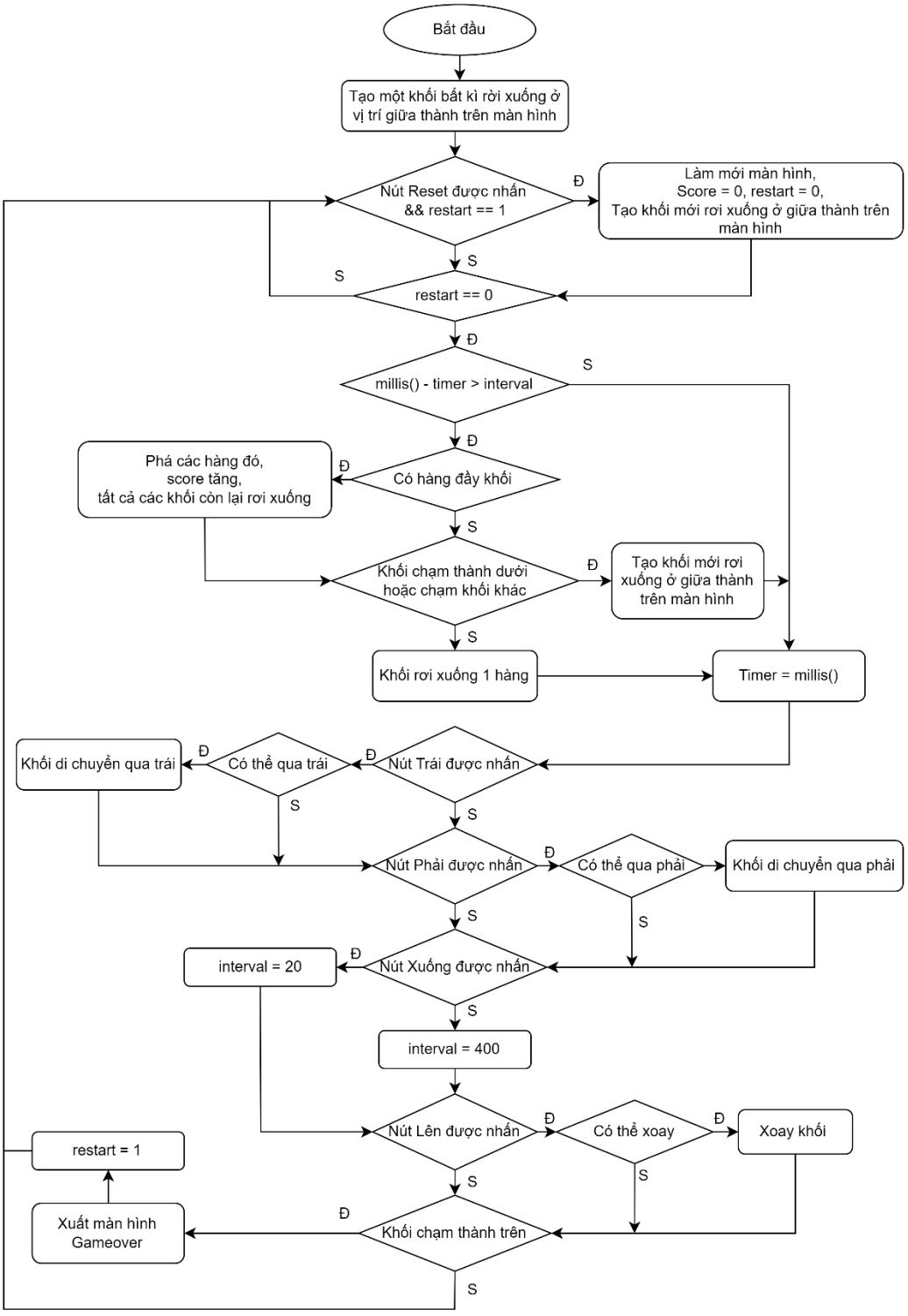
Hình 4.16: *Oled SSD1306 trên Wokwi*

Hình 4.17*: Buzzer trên Wokwi*



Hình 4.18*: Điện trở 10K Ohm trên Wokwi*

4.3. Lưu đồ giải thuật

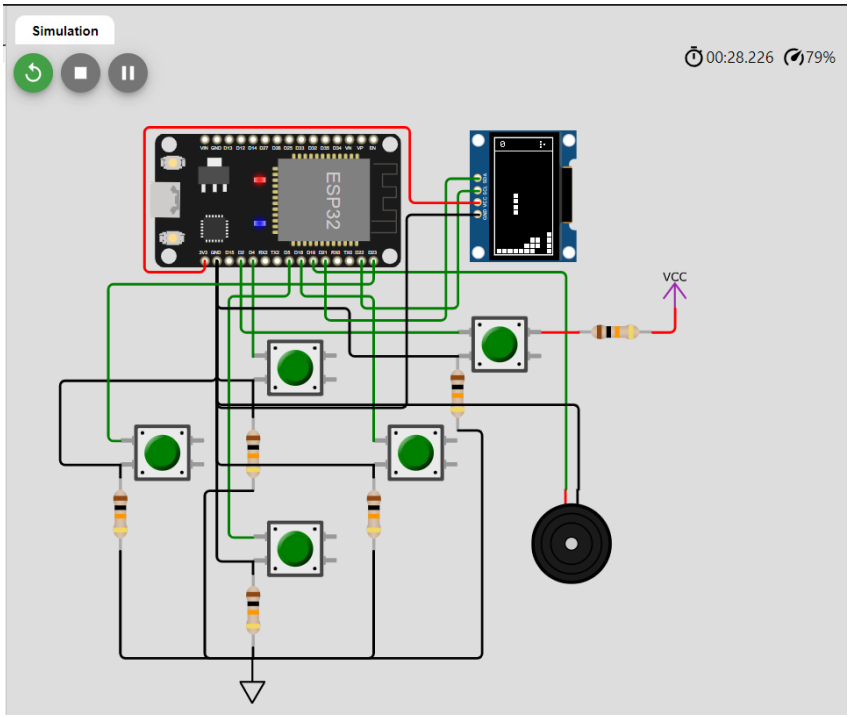


Hình 4.19: *Lưu đồ thuật toán của game Tetris*

4.4. Video mô phỏng

Link video mô phỏng:

https://drive.google.com/file/d/1Rgu8a_Sdpu7W6tQkrxG4uuKly5cpsXuo/view?usp=drive_link



Hình 4.20: *Chạy mô phỏng trên Wokwi*

Chương 5.HIỆN THỰC

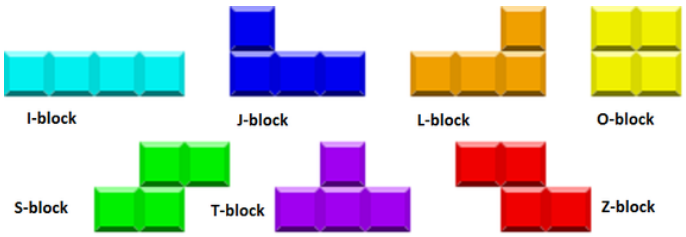
5.1. Lập trình

5.1.1. Ý tưởng game

Game Tetris được phát triển trên vi điều khiển Esp32 là một phiên bản game Tetris cổ điển với những chức năng cơ bản. Người sử dụng các nút trên, xuống, trái, phải để điều khiển các khối với các hình dạng khác nhau được rơi từ trên xuống tạo thành các hàng ngang để ghi càng điểm càng tốt. Trò chơi sẽ kết thúc khi có khối chạm vào thành trên của màn hình.

5.1.2. Cấu trúc game

Thiết lập các hình dạng của khối: Có 7 hình dạng khác nhau của khối: J, L, S, Z, I(thẳng đứng), O(vuông), T và được tạo nên từ 4 khối vuông nhỏ xếp lại với nhau. Các hình dạng khác được tạo nên khi quay các khối trên với các gốc 90^o, 180^o, 270^o.



***Hình 5.1**: Các loại khối trong Tetris*

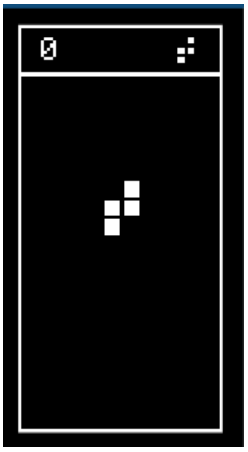
Kiểm tra trạng thái của khối: Luôn kiểm tra trạng thái của khối(đang rơi, chạm thành dưới, chạm khối khác, chạm thành trên) và vị trí của khối trong lúc đang chơi.

Kiểm tra các điều kiện: Khi nhấn nút trên, trái, phải luôn phải kiểm tra khả năng có thể xoay, qua trái, qua phải trước khi thực hiện các thoa tác đó.

Thiết lập âm thanh: Buzzer phát ra tiếng khi người chơi thực hiện các thao tác đặc biệt.

Thiết lặp tăng tốc rơi của khối: Khi người chơi nhấn nút xuống, hệ thống thay đổi thông số tốc độ làm khối rơi nhanh hơn và tốc độ sẽ trở lại bình thường khi nút xuống được thả ra.

Thiết lập màn hình game: Màn hình game bao gồm thanh điểm và khung game. Khung hình game có chiều rộng là 10 ô và chiều dài là 18 ô. Thanh điểm thể hiện điểm đã đạt được(ở gốc trên bên trái) và loại khối sẽ rơi tiếp theo(ở gốc trên bên phải).



Hình 5.2: *Màn hình game*

5.1.3. Quy luật của game

Quy luật tính điểm: Khi tạo được một hay nhiều hàng, điểm của người chơi sẽ tăng với quy ước: 1 hàng là 10 điểm, 2 hàng là 30 điểm, 3 hàng là 60 điểm, 4 hàng là 100 điểm.

Quy luật kết thúc game: Khi chiều cao tối đa mà các khối tạo nên lớn hơn 18 ô (hay nói các khác là khối chạm thành trên của khung game) thì game sẽ kết thúc và hiện ra màn hình kết thúc của game.

Hình 5.3: *Màn hình kết thúc game*

Quy luật của các khối: Khi khối được tạo ra, nó sẽ rơi ở chính giữa màn hình bắt đầu từ thành trên rồi rơi xuống đến khi chạm thành dưới hoặc khối khác. Sau đó sẽ tạo ra khối tiếp theo được quy định bởi khối tiếp theo ở trên thanh điểm. Loại khối được tạo ra đầu tiên và sau đó là ngẫu nhiên.

Quy luật về điều khiển: Nút qua trái, qua phải điều khiển sự di chuyển qua trái hay qua phải của khối đang được rơi xuống. Nút xuống làm tăng tốc độ rơi của khối

và nút lên thực hiện việc xoay khối. Khi màn hình kết thúc game được hiện ra, người chơi có thể nhấn nút reset để bắt đầu game lại từ đầu sau khi đoạn nhạc kết thúc game được phát xong.

Quy luật về âm thanh: Buzzer phát ra tiếng khi người chơi thực hiện nhấn nút trên, trái, phải; khi hoàn thành được một hay nhiều hàng và phát một đoạn nhạc khi kết thúc game.

5.1.4. Quá trình lập trình

Thiết Lập Phần Cứng

Chọn các chân vi điều khiển để sử dụng 4 nút bấm để điều khiển các khối rơi xuống (xoay, tăng tốc, qua trái, qua phải) và 1 nút bấm để reset lại game khi kết thúc game. Các nút này sẽ được kết nối với các chân GPIO sau:

- *Nút xoay:* GPIO4
- *Nút tăng tốc:* GPIO5
- *Nút trái:* GPIO15
- *Nút phải:* GPIO18
- *Nút reset:* GPIO2

Các nút nhấn nối chân còn lại của nút bấm với GND và nối một điện trở 10kΩ

- *Chân nối với Buzzer*: GPIO19

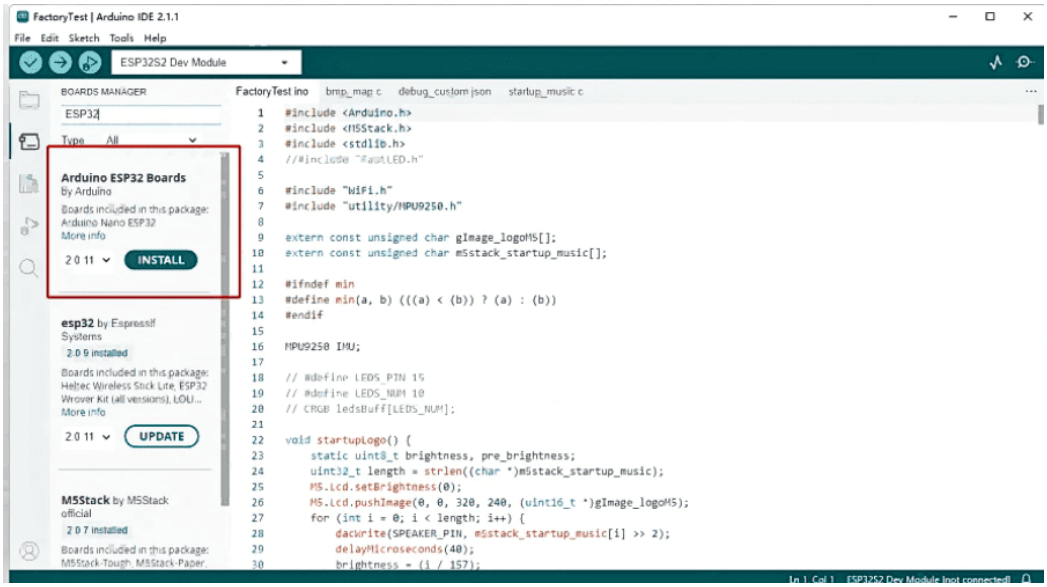
Các chân giao tiếp I2C của Oled với vi điều khiển:

- *Chân nối với SDA của Oled:* GPIO21
- *Chân nối với SCL của Oled:* GPIO22

Cài Đặt Môi Trường Phát Triển

Để phát triển và lập trình game trên vi điều khiển ESP32, chúng em sử dụng môi trường phát triển chính là Arduino IDE [6].

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.



Hình 5.4: *Giao diện của Arduino IDE*

Ưu điểm của Arduino IDE:

- Phần mềm lập trình mã nguồn mở miễn phí.
- Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên.
- Hỗ trợ lập trình tốt cho bo mạch.
- Thư viện hỗ trợ phong phú.
- Giao diện đơn giản, dễ sử dụng.
- Hỗ trợ đa nền tảng như Windows, MacOS, Linux
- Cộng đồng chia sẽ kiên thức lớn.

Cài đặt các thư viện

Để điều khiển hiện thị trên màn hình Oled SSD1306 trên vi điều khiển Esp32 sử dụng Arduino IDE cần cài đặt các thư viên của Adafruit, cụ thể là ***Adafruit_GFX.h*** và ***Adafruit_SSD1306.h***, bên cạnh đó là thư viện ***Wire.h***. Ba thư viện này có nhiều chức năng cần thiết để giao tiếp I2C với màn hình Oled SSD1306.

Định nghĩa chân

Để lập trình điều khiển game qua màn hình,cần định nghĩa các chân GPIO của vi điều khiển ESP32 trong phần code theo sơ đồ đi dây. Việc định nghĩa và liên kết các chân GPIO sẽ giúp chúng ta dễ dàng thử, kiểm tra và kiểm soát các thành phần của trò chơi.

Hiển thị đồ họa và logic game

Lập trình quét LED để hiển thị hình ảnh của các khối đã kết thúc rơi và khối đang rơi trên màn hình OLED và sau mỗi lần quét màn hình sẽ được làm mới để hiển thị các thay đổi về vị trí của khối đang rơi và trạng thái xoay của khối đó, vị trí của các khối đã kết thúc rơi khi có một hàng nhiều hàng được xếp đầy và điểm mới sau khi có hàng được xếp đầy bị xóa. Game sẽ kết thúc khi khối chạm thành trên. Điều kiện này được kiểm tra sau mỗi lần quét led.

Kiểm Tra và sửa lỗi

Để đảm bảo game hoạt động chính xác và mượt mà, thực hiện các bước kiểm tra và sửa lỗi sau lặp lại nhiều lần:

- *Kiểm Tra Các Trường Hợp:* Thử nghiệm game với nhiều trường hợp khác nhau để đảm bảo rằng tất cả các tính năng hoạt động như mong đợi. Bao gồm việc kiểm tra di chuyển của khối rơi, sự ngẫu nhiên của loại khối rơi xuống, cơ chế tính điểm, điều kiện kết thúc game và khởi động lại sau khi kết thúc game.
- *Phát Hiện Lỗi:* Sau nhiều lần kiểm tra cần xác định các lỗi xảy ra như loại khối được rơi không hề ngẫu nhiêu, game không kết thúc khi có khối chạm thành trên và việc khởi động lại game không hoạt động đúng khi nhấn nút reset.
- *Tối Ưu Hóa Code:* Dựa trên các lỗi phát hiện được, tiến hành tối ưu hóa và sửa lỗi trong mã nguồn để cải thiện hiệu suất và độ tin cậy của game.

5.2. Thiết kết PCB

5.2.1. Layout mạch

EasyEDA

EasyEDA [7] là viết tắt của Easy Electronics Design Automation hay Tự động hóa thiết kế điện tử dễ dàng. Đây là gói công cụ dựa trên web cho phép các kỹ sư nhúng, phần cứng và kỹ sư điện... thiết kế mạch, mô phỏng, chia sẻ và xem xét sơ đồ mạch, bảng mạch in và mô phỏng thông qua công cụ toàn diện này.

Đây là phần mềm miễn phí mà bất kỳ ai cũng có thể sử dụng mà không cần mua. Bạn chỉ cần thực hiện quá trình đăng ký và đăng nhập là đã sẵn sàng tạo mạch riêng của bạn. Các hệ điều hành được hỗ trợ là Windows, Mac và Linux. Trình duyệt được hỗ trợ là Chrome, Internet Explorer, Safari và Firefox. Sau khi thiết kế dự án, bạn sẽ không bị mất dự án vì nó sẽ được lưu trên chính trang web EasyEDA.

***Hình 5.5**: Giao diện trang web EasyEDA*

Công dụng của EasyEDA:

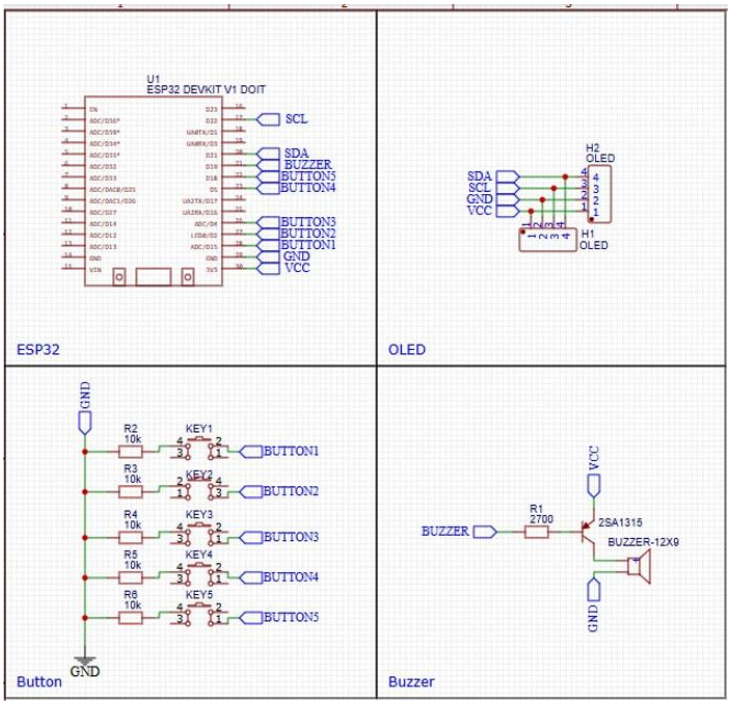
- *Thiết kế sơ đồ mạch (Schematic Capture):* Cho phép người dùng tạo ra các sơ đồ mạch điện tử chi tiết và dễ dàng.
- *Thiết kế PCB (PCB Layout):* Hỗ trợ tạo và chỉnh sửa các bản vẽ PCB với các công cụ như auto-routing, DRC (Design Rule Check), và nhiều công cụ khác.
- *Mô phỏng mạch (Circuit Simulation):* Cung cấp khả năng mô phỏng các mạch điện tử để kiểm tra và xác nhận hoạt động của chúng trước khi sản xuất.
- *Thư viện linh kiện phong phú:* Cung cấp hàng ngàn linh kiện điện tử từ các nhà sản xuất lớn, giúp dễ dàng thêm vào thiết kế của bạn.
- *Cộng đồng và chia sẻ:* Người dùng có thể chia sẻ thiết kế của họ với cộng đồng hoặc tìm kiếm các thiết kế từ người dùng khác.

Quá trình thực hiện trên EasyEDA:

1) Chọn vi điều khiển esp32-devkit V1 DOIT, 5 nút nhấn, màn hình Oled, buzzer, điện trở từ thư viện linh kiện.
2) Kết nối nút "xoay" với chân GPIO4 của vi điều khiển.
3) Kết nối nút "tăng tốc" với chân GPIO5 của vi điều khiển.
4) Kết nối nút "trái" với chân GPIO15 của vi điều khiển.
5) Kết nối nút "phải" với chân GPIO18 của vi điều khiển.
6) Kết nối nút "reset" với chân GPIO2 của vi điều khiển.
7) Kết nối một đầu của mỗi điện trở với chân 3.3V của vi điều khiển.
8) Kết nối chân còn lại của mỗi điện trở với chân GPIO tương ứng của các nút nhấn (song song với các nút nhấn).
9) Kết nối chân còn lại của mỗi nút nhấn với GND của vi điều khiển.
10) Sau đó bấm nút chạy để tạo ra mẫu PCB
11) Chỉnh lại vị trí linh kiện và dây

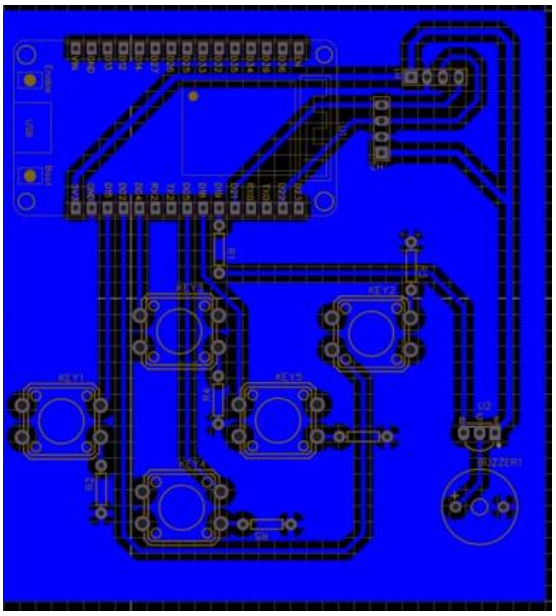
12)Kiểm tra và in mạch

Sau khi thực hiện quá trình trên ta sẽ được một sơ đồ nguyên lí như sau:



Hình 5.6: *Sơ đồ nguyên lí*

Kèm theo đó là một thiết kế mạch PCB sau khi chỉnh lại vị trí linh kiện và dây



Hình 5.7: *Thiết kế mạch PCB*

5.2.2. Gia công

Xuất thiết kế thành file pdf

Sau khi hoàn thành thiết kế, bản vẽ mạch được xuất ra file PDF. File PDF này phải bao gồm tất cả các lớp cần thiết của mạch in, chẳng hạn như lớp mạch dẫn, lớp linh kiện, và lớp phủ hàn. Việc xuất ra file PDF đảm bảo rằng thiết kế sẽ được in chính xác và rõ ràng.

In mạch ra giấy in mạch

Chuẩn bị giấy in mạch chuyên dụng có bề mặt đặc biệt giúp chuyển mạch lên phíp đồng dễ dàng. Đảm bảo giấy in mạch không bị nhăn hoặc có bụi bẩn. File PDF được in ra trên giấy in mạch. Máy in phải có độ phân giải cao để đảm bảo các chi tiết nhỏ của mạch không bị mờ hoặc biến dạng. Quá trình in phải được thực hiện cẩn thận để đảm bảo tính chính xác của thiết kế.

Ủi mạch lên phíp đồng

Chuẩn Bị Phíp đồng phải được làm sạch để loại bỏ bụi bẩn. Sử dụng dung dịch làm sạch chuyên dụng để đảm bảo bề mặt phíp đồng hoàn toàn sạch sẽ. Đặt giấy in mạch lên bề mặt phíp đồng sao cho mặt in tiếp xúc với đồng. Sử dụng bàn ủi nóng (điều chỉnh ở nhiệt độ phù hợp, thường là khoảng 150-180°C) ủi đều lên mặt giấy. Quá trình này làm cho mực in chuyển từ giấy sang bề mặt phíp đồng. Ủi đều tay và cẩn thận để mực in chuyển hết và không bị lem.

Ngâm dung dịch ăn mòn

Chuẩn Bị dung dịch ăn mòn như muối ăn mòn. Đảm bảo an toàn khi sử dụng dung dịch ăn mòn bằng cách đeo găng tay, kính bảo hộ và làm việc trong không gian thông thoáng. Ngâm phíp đồng đã ủi vào dung dịch ăn mòn. Dung dịch sẽ ăn mòn phần đồng không được phủ mực in, để lại các đường dẫn điện theo thiết kế. Quá trình ngâm cần được theo dõi chặt chẽ để đảm bảo không ăn mòn quá mức, thường kéo dài từ 10 đến 30 phút tùy thuộc vào nồng độ dung dịch và nhiệt độ.

Khoan lỗ để hàn linh kiện lên

Sau khi ăn mòn hoàn tất, loại bỏ phần bảo vệ và tiến hành khoan các lỗ trên mạch để đặt chân linh kiện. Sử dụng máy khoan nhỏ với mũi khoan có đường kính phù hợp với kích thước chân linh kiện. Quá trình khoan cần độ chính xác cao để đảm bảo các lỗ khoan đúng vị trí thiết kế.

Hàn linh kiện

Đặt các linh kiện vào vị trí đã khoan lỗ và hàn chúng lên mạch bằng mỏ hàn và thiếc hàn. Đảm bảo nhiệt độ mỏ hàn phù hợp (200^{o}C – 350^{o}C), kích thước thiếc hàn vừa đủ, làm sạch vùng cần hàn và cố định các linh kiện.

Vệ sinh và bảo vệ mạch

Sau khi hàn linh kiện, mạch còn lưu lại một số chất bẩn. Sử dụng axetol hoặc các dung dịch làm sạch chuyên dụng để vệ sinh mạch. Sau đó cho một lớp dung dịch bảo vệ mạch điện. Việc vệ sinh và bảo vệ giúp loại bỏ các chất cặn bẩn, tăng tính thẩm mĩ, tăng thời gian sử dụng của mạch.

Kiểm tra lại mạch

Sau khi vệ sinh, kiểm tra lại toàn bộ mạch để đảm bảo không có các lỗi hàn. Sử dụng đồng hồ đo vạn năng để kiểm tra những đường mạch. Đảm bảo rằng các đường mạch không bị đứt và không bị ngắn mạch và tất cả các linh kiện đều hoạt động đúng cách

5.2.3. Kiểm thử và điều chỉnh lại mạch

Chạy thử chương trình trên mạch

Nạp chương trình kiểm thử vào ESP32 DevKit V1 và chạy chương trình. Chương trình kiểm thử nên bao gồm các bài kiểm tra cho tất cả các chức năng của mạch như đọc dữ liệu của nút bấm, giao tiếp với các thiết bị ngoại vi, và kiểm tra tín hiệu trên các chân I/O.

Phát hiện lỗi phát sinh

Trong quá trình kiểm tra, phát hiện rằng chân GPIO2 không hoạt động đúng do điện áp khi có điện trở kéo lên chỉ đạt 0,7V thay vì mức logic cao (VCC), đây là nguyên nhân khiến cho vi điều khiển không thể giao tiếp được với nút reset(nút được nối với chân GPIO2 của vi điều khiển).

Khắc phục lỗi phát sinh

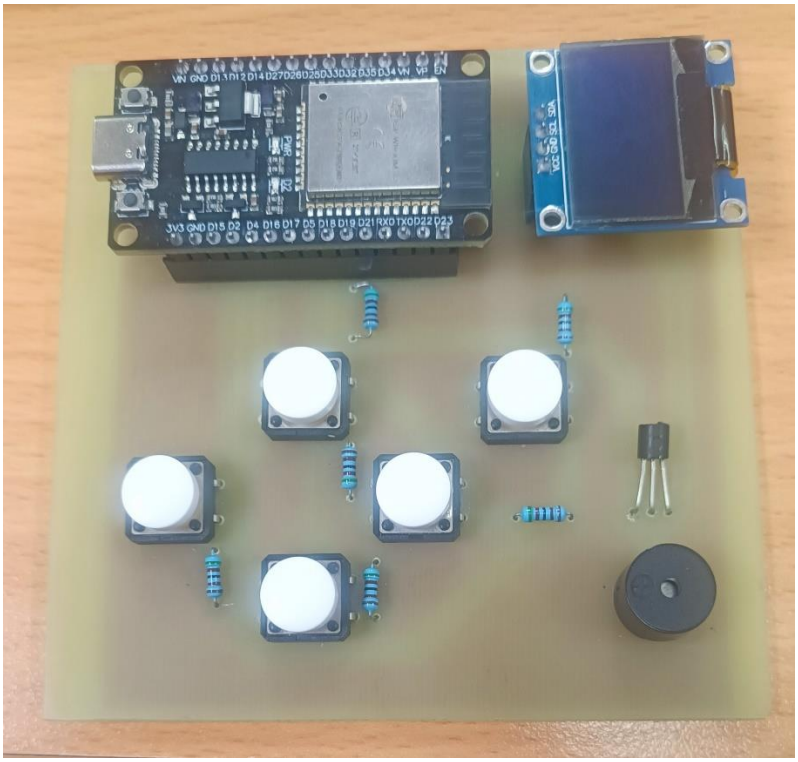
Để khắc phục vấn đề trên chân GPIO2, cần hàn thêm một điện trở 10k Ohm kéo lên nguồn VCC. Điều này sẽ đảm bảo rằng chân GPIO2 có mức điện áp đúng và hoạt động ổn định với nguồn điện 3V3 và điện trở giúp đảm bảo an toàn.

5.2.4. Sản phẩm thu được

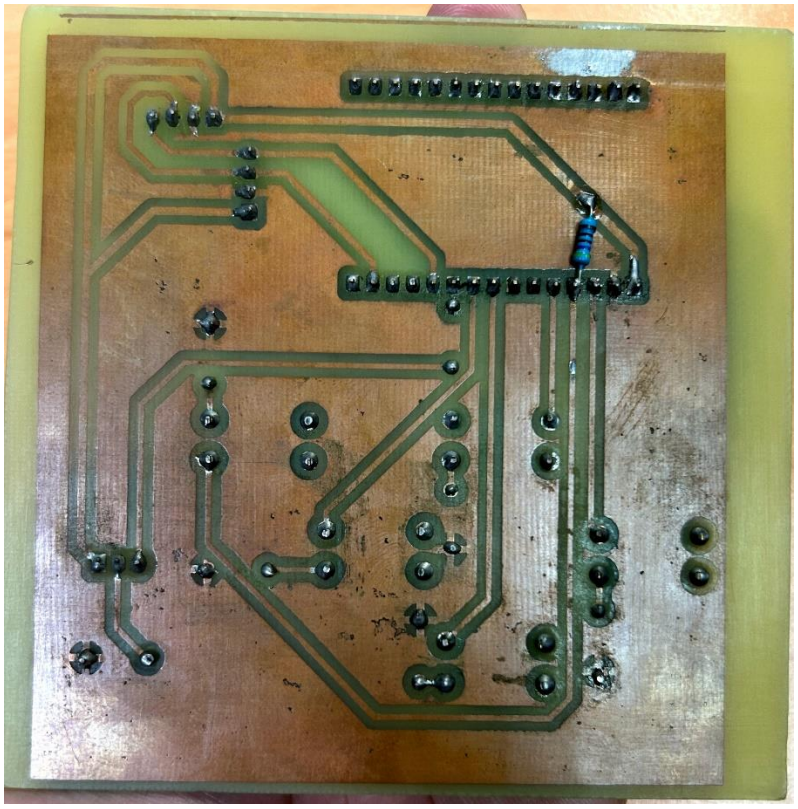
Kết quả sau quá trình thiết kế, in mạch và kiểm tra mạch PCB là một mạch PCB có kích thước 10 cm x 10 cm, gồm:

- 1 vi điều khiển Esp32-DevKit-V1
- 1 màn hình Oled SSD1306
- 5 nút nhấn tactile
- 6 điện trở 10k Ohm
- 1 transistor PNP
- 1 buzzer thụ động

Dưới đây là hình ảnh của mạch PCB:



Hình 5.8: *Mặt trước của mạch PCB*



Hình 5.9: *Mặt sau của mạch PCB*

Chương 6.KẾT QUẢ

Mạch PCB đã được thiết kế và gia công thành công với các tiêu chí nghiêm ngặt về độ chính xác và chất lượng hàn, đảm bảo mọi khía cạnh kỹ thuật đều đạt chuẩn cao nhất. Quá trình thiết kế và sản xuất đã trải qua nhiều bước kiểm tra và tối ưu hóa để đảm bảo rằng mạch không chỉ hoạt động ổn định mà còn đáp ứng đầy đủ mọi yêu cầu khắt khe của thiết kế ban đầu.

Mạch đã trải qua nhiều bài kiểm tra chức năng và độ bền một cách kỹ lưỡng, tất cả đều cho thấy mạch hoạt động đúng như mong đợi. Các chức năng chính bao gồm đọc/ghi dữ liệu, giao tiếp với các thiết bị ngoại vi, và điều khiển tín hiệu I/O đều vận hành ổn định, không gặp bất kỳ trục trặc nào trong suốt quá trình thử nghiệm.

Ứng dụng thực tiễn của mạch PCB này là trò chơi Tetris được hiện thị trên màn hình OLED, một dự án đã được phát triển thành công và mang lại nhiều trải nghiệm thú vị cho người dùng. Màn hình OLED hiển thị hình ảnh rõ ràng và sắc nét, loa phát ra âm thanh chất lượng cao, và các nút bấm được thiết kế nhạy bén, hoạt động mượt mà, cho phép người dùng điều khiển trò chơi một cách dễ dàng và chính xác. Đặc biệt, sự phối hợp giữa các thành phần điện tử đã tạo nên một hệ thống đồng bộ, đảm bảo trải nghiệm chơi game liên tục và không bị gián đoạn.

Link video chạy sản phẩm:

https://drive.google.com/file/d/15uRVZRVBOO9l7fUVRhPDq4kgawy4Ubsz/view?usp=drive_link

Chương 7.KẾT LUẬN

Đồ án "Máy Game Tetris trên ESP32" đã mang lại cho nhóm một hành trình học tập và áp dụng kiến thức vô cùng giá trị. Được thực hiện trong khuôn khổ môn học vi xử lý-vi điều khiển tại Đại học Công nghệ thông tin - ĐHQG.TPHCM, đề tài được chọn dựa trên mong muốn xây dựng một trò chơi thú vị và phổ biến trên nền tảng ESP32. Việc nghiên cứu và thực hành trò chơi này đã giúp củng cố và mở rộng các kỹ năng quan trọng trong lĩnh vực vi điều khiển và thiết kế mạch PCB.

Bắt đầu từ việc tìm hiểu sâu về vi điều khiển ESP32 là một trong những bước quan trọng nhất của dự án. Nhóm đã nghiên cứu về cấu trúc của ESP32, các tính năng nổi bật như WiFi, Bluetooth, và cách lập trình cho vi điều khiển này bằng Arduino IDE. Qua đó, nhóm đã nắm được cách thức hoạt động của vi điều khiển, cũng như những điều cần lưu ý khi sử dụng và triển khai các ứng dụng IoT.

Việc tích hợp màn hình OLED, buzzer và các nút bấm là một phần không thể thiếu trong dự án. Nhóm đã tìm hiểu sâu về cách kết nối và điều khiển các linh kiện này từ ESP32. Quá trình này không chỉ giúp nâng cao kỹ năng kỹ thuật mà còn cho thấy sự quan trọng của sự chính xác và chi tiết trong thiết kế mạch điện tử.

Thiết kế và vẽ mạch PCB không chỉ đơn thuần là một bước trong quá trình sản xuất, mà còn là cơ hội để áp dụng các kiến thức về routing, layout và chọn linh kiện. Nhóm đã học được cách phối hợp giữa các linh kiện trên một mặt PCB, đảm bảo điều kiện tối ưu cho hoạt động của mạch.

Trong suốt quá trình thực hiện, nhóm đã đối mặt với nhiều thử thách và vấn đề phát sinh. Tuy nhiên, nhờ vào sự cộng tác chặt chẽ và quyết tâm vượt qua, nhóm đã từng bước giải quyết, rút kinh nghiệm và học hỏi từ mỗi sai lầm. Điều này không chỉ giúp nâng cao khả năng giải quyết vấn đề mà còn tạo đà cho sự tiến bộ của mỗi cá nhân và cả nhóm.

Chương 8.PHỤ LỤC

8.1. Quá trình làm việc

Tìm hiểu về các dòng vi điều khiển

Ban đầu, nhóm đã tiến hành nghiên cứu và so sánh nhiều dòng vi điều khiển khác nhau như Arduino Nano, STM32 và ESP32. Sau khi xem xét kỹ lưỡng các tính năng, khả năng mở rộng và mục đích sử dụng trong tương lai, quyết định chọn ESP32. ESP32 là một vi điều khiển mạnh mẽ, tích hợp Wi-Fi và Bluetooth, phù hợp với các dự án yêu cầu kết nối không dây và xử lý tốc độ cao trong tương lai.

Tìm mua linh kiện

Sau khi quyết định sử dụng ESP32, đã lập danh sách các linh kiện cần thiết cho dự án bao gồm ESP32 DevKit V1, màn hình OLED, nút bấm, buzzer, điện trở và các linh kiện phụ trợ khác. Việc mua sắm linh kiện được thực hiện thông qua các nhà cung cấp trực tuyến và cửa hàng điện tử tại địa phương.

Code và mô phỏng trên web mô phỏng wokwi

Trước khi tiến hành làm mạch thực tế, đã sử dụng website mô phỏng wokwi.com để thử nghiệm và kiểm tra code. Đây là bước quan trọng giúp phát hiện và sửa chữa lỗi logic trong mã trước khi nạp vào vi điều khiển thực tế, đồng thời đảm bảo các kết nối giữa các linh kiện là chính xác.

Cài đặt môi trường phát triển

Sử dụng Arduino IDE để viết và nạp code cho ESP32 do giao diện thân thiện và dễ sử dụng.

Test code và linh kiện trên breadboard

Sau khi hoàn thiện code và mô phỏng thành công, tiến hành kiểm tra mã và các linh kiện trên breadboard. Việc này giúp xác nhận rằng tất cả các linh kiện hoạt động đúng như mong đợi và phát hiện sớm các vấn đề tiềm ẩn trước khi thực hiện bước tiếp theo.

Tuy nhiên, trong quá trình test trên breadboard thì nhóm đã gặp phải sự cố khi làm tổn thất 1 con ESP32 do lỗi không xác định. Sự cố này đã ảnh hưởng đến thời gian và chi phí của nhóm.

Thiết kế và vẽ mạch PCB

Khi đã chắc chắn về các kết nối và code, bắt đầu thiết kế và vẽ mạch PCB. Sử dụng phần mềm thiết kế mạch, tạo ra layout PCB chi tiết, đảm bảo rằng tất cả các linh kiện được bố trí hợp lý và các đường dẫn tín hiệu được tối ưu hóa để giảm thiểu nhiễu và đảm bảo hiệu suất mạch.

Sửa lỗi và hoàn thiện sản phẩm

Sau khi hoàn thành thiết kế, mạch PCB được in và gia công. Quá trình này bao gồm kiểm tra và sửa lỗi phát sinh. Một ví dụ điển hình là vấn đề với chân D2 của ESP32 DevKit V1, nơi mà điện trở kéo lên chỉ đạt 0,7V thay vì mức cần thiết. đã khắc phục bằng cách thêm điện trở 10k ohm kéo lên nguồn VCC.

Viết báo cáo

Cuối cùng, tổng hợp tất cả các thông tin, tài liệu hóa quá trình làm việc và viết báo cáo. Báo cáo chi tiết từ việc nghiên cứu, thiết kế, thử nghiệm đến hoàn thiện sản phẩm. Nó không chỉ là một tài liệu học thuật mà còn là cơ sở để tham khảo cho các dự án tương lai.

8.2. Chi phí thực hiện

Linh kiện	Đơn giá	Số lượng	Thành tiền
ESP32	112.000	1	112.000
Buzzer 1208TS	2.000	1	2.000
B3F-4055 12x12 nút nhấn	2.500	5	12.500
OLED SSD1306 I2C	50.000	1	50.000
A1015 Transistor PNP	500	10	5.000
Nắp nút nhấn 12x12	300	10	3.000
Tấm đồng FR4 2 lớp 10x10	7.500	1	7.500
Trở 10k ohm	56	50	2.800
Bột ăn mòn	25.000	1	25.000
Hàng rào cái đơn 40 chân	2.000	2	4.000
Tổng cộng			223.800

***Bảng 1**: Bảng chi phí thực hiện*

TÀI LIỆU THAM KHẢO

[1] A. Pajitnov, "Tetris game," [Online]. Available: https://vi.wikipedia.org/wiki/Tetris. [Accessed 1 March 2024].

[2] "Pulse-width modulation," [Online]. Available: https://en.wikipedia.org/wiki/Pulse-width_modulation. [Accessed 15 March 2024].

[3] "I2C," [Online]. Available: https://en.wikipedia.org/wiki/I%C2%B2C. [Accessed 15 March 2024].

[4] "ESP32," ESPRESSIF. [Online]. [Accessed 28 February 2024].

[5] "Wokwi," [Online]. Available: https://wokwi.com/. [Accessed 10 March 2024].

[6] "Arduino IDE," [Online]. Available: https://www.arduino.cc/en/software. [Accessed 15 March 2024].

[7] "EasyEDA," [Online]. Available: https://easyeda.com/. [Accessed 6 May 2024].