

ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN – ĐIỆN TỬ



BÁO CÁO ĐỒ ÁN THIẾT KẾ II

Đề tài: Bàn phím cơ Bluetooth

Giảng viên hướng dẫn

Nguyễn Minh Đức

Mã lớp

748493

Nhóm sinh viên thực hiện

Phạm Văn An

20213790

Lê Minh Long

20213986

Vũ Tuấn Minh

20214015

Hà Nội, 2025

MỤC LỤC

LỜI NÓI ĐẦU.....	2
PHÂN CÔNG CÔNG VIỆC	3
CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI.....	4
1.1 Tổng quan về đề tài	4
1.1.1 Đặt vấn đề	4
1.1.2 Tổng quan về đề tài	4
1.2 Yêu cầu chức năng và phi chức năng	4
1.3 ESP32	5
1.3.1 Giới thiệu ESP32	5
1.3.2 Thông số kỹ thuật và sơ đồ chân ESP32	6
1.4 LCD20x4	9
1.4.1 Thông số kỹ thuật	9
1.4.2 Các chân kết nối	10
CHƯƠNG 2. THIẾT KẾ SẢN PHẨM.....	11
2.1 Tổng quan ngôn ngữ C	11
2.2 Mô hình hóa hệ thống.....	12
2.2.1 Sơ đồ Use-case	12
2.2.2 Sơ đồ thuật toán	13
2.2.3 Sơ đồ khối của hệ thống	14
2.3 Thiết kế mạch bằng phần mềm Altium	15
CHƯƠNG 3. KẾT QUẢ SẢN PHẨM	17

LỜI NÓI ĐẦU

Trong thời kỳ công nghiệp hóa, hiện đại hóa hiện nay, các thiết bị điện tử và hệ thống tự động hóa đóng vai trò vô cùng quan trọng. Chúng không chỉ góp phần nâng cao năng suất trong các nhà máy, phân xưởng mà còn mang lại nhiều tiện ích cho đời sống gia đình. Đặc biệt, lĩnh vực ứng dụng điện tử số ngày càng phát triển mạnh mẽ nhờ vào tính đa dạng, độ chính xác cao và những ưu điểm vượt trội so với kỹ thuật tương tự.

Nhận thức được tầm quan trọng đó, nhóm chúng em đã lựa chọn nghiên cứu và thực hiện đề tài thiết kế bàn phím Bluetooth, một sản phẩm kết hợp giữa công nghệ điện tử và truyền thông không dây. Bàn phím Bluetooth không chỉ là một giải pháp tiện lợi trong việc nhập liệu mà còn mở ra nhiều ứng dụng hữu ích trong các thiết bị di động và hệ thống thông minh, đáp ứng nhu cầu của thời đại công nghệ 4.0.

Trong suốt quá trình nghiên cứu và thực hiện đề tài, chúng em đã nhận được sự hướng dẫn tận tình từ thầy Nguyễn Minh Đức. Thầy đã quan tâm, chỉ dẫn chi tiết và cung cấp nhiều tài liệu quý báu, giúp nhóm em tiếp thu kiến thức nền tảng về thiết kế mạch điện tử, lập trình và sử dụng các phần mềm như Proteus, Altium để triển khai đồ án.

Mặc dù đã nỗ lực hoàn thành đề tài, chúng em nhận thấy đồ án vẫn còn một số thiếu sót. Chúng em rất mong nhận được ý kiến đóng góp từ thầy để có thể cải thiện và phát triển sản phẩm hoàn thiện hơn trong tương lai.

Chúng em xin chân thành cảm ơn thầy!

PHÂN CÔNG CÔNG VIỆC

Thành viên	Công việc	Kết quả
Vũ Tuấn Minh	Lập trình cho sản phẩm, làm báo cáo	Hoàn thành
Phạm Văn An	Thiết kế PCB, làm mạch in thủ công cho sản phẩm, làm báo cáo	Hoàn thành
Lê Minh Long	Thiết kế vỏ sản phẩm, làm báo cáo	Hoàn thành

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1 Tổng quan về đề tài

1.1.1 Đặt vấn đề

Trong thời đại công nghệ số, nhu cầu sử dụng các thiết bị kết nối không dây ngày càng tăng cao nhờ vào đặc tính tiện lợi và linh hoạt của chúng. Trong đó, bàn phím Bluetooth đã trở thành một trong những thiết bị được ưa chuộng nhất với đa dạng ứng dụng từ cá nhân đến chuyên nghiệp. Tuy nhiên, phần lớn các bàn phím Bluetooth hiện có trên thị trường thường không cho phép tùy chỉnh cấu hình hoặc chức năng, dẫn đến việc chưa thực sự tối ưu hóa cho nhu cầu sử dụng cụ thể.

Việc nghiên cứu và thực hiện bàn phím Bluetooth tùy chỉnh sử dụng module ESP32 và màn hình LCD 20x4 nhằm cung cấp một giải pháp tiện lợi, đầy đủ chức năng và linh hoạt cho người dùng. Bên cạnh đó, đề tài hướng tới việc đem lại những kinh nghiệm thực tiễn trong việc tích hợp phần cứng và lập trình nhúng.

1.1.2 Tổng quan về đề tài

Bàn phím Bluetooth tùy chỉnh sử dụng ESP32 là một dự án kết hợp giữa công nghệ vi điều khiển và kết nối không dây. Module ESP32 được lựa chọn với nhiều đặc điểm nổi bật như khả năng kết nối Bluetooth và Wi-Fi tích hợp, độ tin cậy cao, và khả năng xử lý nhanh chóng. Bên cạnh đó, màn hình LCD 20x4 được sử dụng như một giao diện hiển thị trực quan, cho phép người dùng theo dõi các trạng thái và thao tác một cách dễ dàng.

Trên thị trường, các bàn phím Bluetooth thương mại đa phần hạn chế trong việc cung cấp các tùy chỉnh phục vụ nhu cầu đặc thù. Dự án này nhằm khai thác khả năng cá nhân hoá thiết bị thông qua lập trình nhúng, cung cấp khả năng tùy biến chức năng của bàn phím như gán macro, thay đổi layout phím và hiển thị thông tin theo nhu cầu.

Dự án còn bao gồm việc khai thác tính năng tiên tiến như giao tiếp I2C và UART để giao tiếp với các thiết bị ngoại vi. Việc tích hợp này không chỉ đưa ra một bàn phím tùy chỉnh mà còn cung cấp cơ hội để mở rộng nó sang nhiều ứng dụng khác.

1.2 Yêu cầu chức năng và phi chức năng

Yêu cầu chức năng	Yêu cầu phi chức năng
Hỗ trợ kết nối Bluetooth với nhiều thiết bị khác nhau	Thiết kế giao diện người dùng trực quan và dễ sử dụng
Cho phép tùy chỉnh layout phím theo nhu cầu người dùng	Đảm bảo độ tin cậy cao khi kết nối Bluetooth

Hiển thị thông tin trên màn hình LCD 20x4 bao gồm trạng thái kết nối và các phím chức năng	Thêm các tùy chọn cài đặt linh hoạt như chọn tần số kết nối
Gán macro cho các phím để tự động hóa các tác vụ phức tạp	Tiết kiệm năng lượng, đáp ứng nhu cầu sử dụng trong thời gian dài
Tích hợp giao tiếp I2C và UART để giao tiếp với các thiết bị ngoại vi	Khả năng dễ dàng tích hợp và phát triển thêm chức năng trong tương lai

1.3 ESP32

1.3.1 Giới thiệu ESP32

ESP32 là một dòng vi điều khiển tích hợp Wi-Fi và Bluetooth (BLE) được phát triển bởi Espressif Systems. Đây là một trong những vi điều khiển phổ biến nhất trong các dự án IoT (Internet of Things) nhờ vào khả năng kết nối mạnh mẽ, hiệu năng cao, và giá thành hợp lý.

Đặc điểm chính của ESP32:

Tích hợp Wi-Fi và Bluetooth:

- Wi-Fi: Hỗ trợ chuẩn IEEE 802.11 b/g/n, giúp kết nối internet không dây.
- Bluetooth: Hỗ trợ Bluetooth 4.2 (BLE - Bluetooth Low Energy), cho phép giao tiếp với các thiết bị khác.

Hiệu năng cao:

- Bộ vi xử lý lõi kép (dual-core) hoặc lõi đơn, dựa trên kiến trúc Xtensa LX6, với tốc độ xung nhịp lên tới 240 MHz.
- Bộ nhớ RAM (SRAM) từ 520 KB đến 4 MB (tùy phiên bản).
- Bộ nhớ Flash từ 4 MB trở lên.

Đa chức năng:

- GPIO (General Purpose Input Output): Có hơn 30 chân I/O hỗ trợ ADC, DAC, SPI, I2C, UART, và PWM.
- Hỗ trợ các giao thức như MQTT, HTTP, WebSocket phục vụ lập trình IoT.
- Tích hợp cảm biến nhiệt độ và cảm biến hall.

Tiêu thụ năng lượng thấp:

- Chế độ Deep Sleep giúp tiết kiệm năng lượng, phù hợp cho các ứng dụng chạy pin.

Khả năng lập trình linh hoạt:

- Hỗ trợ lập trình bằng nhiều ngôn ngữ như C, C++ qua Arduino IDE, ESP-IDF (Espressif IoT Development Framework), MicroPython.

1.3.2 Thông số kỹ thuật và sơ đồ chân ESP32

1.3.2.1. Thông số kỹ thuật

CPU:

- Bộ vi xử lý: Xtensa® Dual-Core 32-bit LX6.
- Tần số hoạt động: Lên đến 240 MHz.
- Hiệu suất: 600 DMIPS.

RAM và bộ nhớ:

- SRAM: 520 KB (dùng cho dữ liệu và hướng dẫn).
- Bộ nhớ Flash: Thường từ 4 MB (có thể mở rộng tùy module).
- Bộ nhớ ROM: 448 KB (chứa firmware và phần khởi động).

Wi-Fi và Bluetooth:

- Wi-Fi: 802.11 b/g/n.
 - Hỗ trợ cả AP (Access Point), STA (Station), và chế độ kết hợp AP+STA.
- Bluetooth:
 - BLE (Bluetooth Low Energy).
 - Bluetooth v4.2 Classic.

Công suất tiêu thụ:

- Tiêu thụ năng lượng thấp với nhiều chế độ (Active, Light Sleep, Deep Sleep).
- Dòng tiêu thụ ở chế độ Deep Sleep: ~10 μ A.

GPIO và I/O:

- Số chân GPIO: Tối đa 36 chân (tùy module cụ thể).
- Hỗ trợ các chức năng: ADC, DAC, SPI, I2C, I2S, UART, PWM.
- ADC: 12-bit (18 kênh).
- DAC: 8-bit (2 kênh).
- Giao tiếp CAN (Controller Area Network).

Mô-đun tích hợp:

- Timer: 4 bộ đếm 64-bit có thể định cấu hình.
- RTC: Đồng hồ thời gian thực tích hợp (khi chạy ở chế độ Deep Sleep).
- Touch Sensor: Lên đến 10 kênh cảm ứng điện dung.
- SD Card: Hỗ trợ SDIO và SDMMC.

Tính năng bảo mật:

- Hỗ trợ mã hóa AES, SHA-2, RSA.
- Mã hóa Flash (Secure Boot).
- Mã hóa dựa trên phần cứng.

Nguồn điện:

- GPIO12 (MTDI): Phụ thuộc vào mức điện áp khởi động. Phải cẩn thận khi dùng làm đầu vào.

ADC và DAC:

- ADC (Analog to Digital Converter):
 - ESP32 có 18 kênh ADC, từ ADC1_CH0 đến ADC1_CH9 và ADC2_CH0 đến ADC2_CH9.
 - Độ phân giải 12-bit (4096 mức giá trị).
- DAC (Digital to Analog Converter):
 - Có 2 kênh DAC: DAC1 (GPIO25) và DAC2 (GPIO26).
 - Dùng để xuất tín hiệu analog.

Touch Sensor:

- Có 10 cảm biến điện dung (touch pads):
- T0 đến T9 tương ứng với các chân từ GPIO4 đến GPIO15.
- Dùng để nhận diện chạm.

SPI (Serial Peripheral Interface):

- Hỗ trợ 4 giao tiếp SPI: VSPI, HSPI,...
- Các chân phổ biến:
 - SCK (Clock): GPIO18.
 - MOSI (Master Out Slave In): GPIO23.
 - MISO (Master In Slave Out): GPIO19.
 - CS (Chip Select): GPIO5.

I2C (Inter-Integrated Circuit):

- Có 2 chân mặc định:
- SDA (Data): GPIO21.
- SCL (Clock): GPIO22.
- Các chân này có thể định nghĩa lại trong phần mềm.

UART (Universal Asynchronous Receiver Transmitter):

- Hỗ trợ 3 giao tiếp UART:
 - UART0: TX (GPIO1), RX (GPIO3).
 - UART1: TX (GPIO9), RX (GPIO10).
 - UART2: TX (GPIO17), RX (GPIO16).
- PWM (Pulse Width Modulation):
 - Tất cả chân GPIO đều hỗ trợ PWM.
 - Dùng để điều khiển LED, động cơ hoặc tạo tín hiệu dao động.

RTC GPIO:

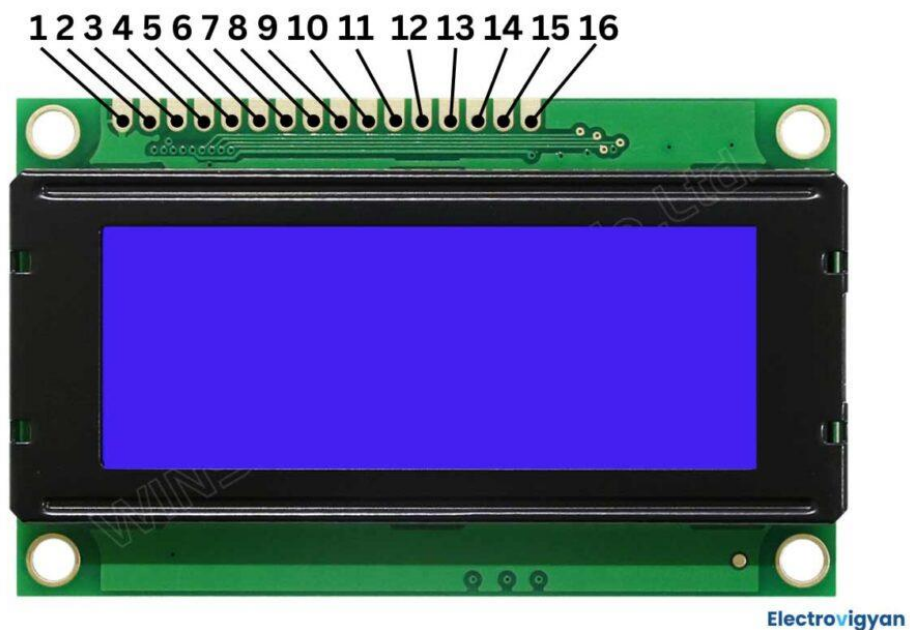
- Một số chân có chức năng RTC GPIO, dùng cho chế độ tiết kiệm năng lượng (Deep Sleep):

- GPIO12, GPIO13, GPIO14, GPIO15, GPIO25, GPIO26, GPIO27, GPIO32, GPIO33.

Chân đặc biệt:

- BOOT (GPIO0): Chân cần kéo xuống GND để đưa module vào chế độ flash firmware.
- SDIO: Các chân GPIO6 đến GPIO11 thường được dùng cho giao tiếp với bộ nhớ Flash SPI.
- LOG (GPIO7, GPIO8): Một số module dùng làm debug

1.4 LCD20x4



Hình 1.2: LCD 20x4

1.4.1 Thông số kỹ thuật

Kích thước:

- Kích thước module: 98mm x 60mm x 14mm (dài x rộng x cao).
- Kích thước khu vực hiển thị: ~76mm x 25mm.

Hiển thị:

- Số ký tự: 20 ký tự x 4 dòng.
- Kích thước mỗi ký tự: 5x8 ma trận pixel.

Nguồn điện:

- Điện áp hoạt động: 4.7V - 5.3V DC.
- Dòng tiêu thụ: ~1mA (không bao gồm đèn nền).

Giao tiếp:

- 4-bit hoặc 8-bit song song (Parallel Interface).
- Tương thích với vi điều khiển như Arduino, STM32, ESP32.

Đèn nền (Backlight):

- Màu: Vàng-xanh dương (phổ biến), hoặc xanh dương-trắng.
- Tùy chọn bật/tắt qua chân K (đất).

1.4.2 Các chân kết nối

LCD 20x4 thường bao gồm 16 chân:

Bảng 1.1: Bảng thông số chân LCD 20x4

Chân	Tên	Mô tả
1	VSS	Nối đất (GND)
2	VDD	Nối với 5V
3	VO	Điều chỉnh độ tương phản
4	RS	Chọn thanh ghi (Command/Data)
5	RW	Chọn đọc/ghi (Read/Write)
6	E	Chân Enable (để kích hoạt)
7-14	D0-D7	Dữ liệu (song song 4 bit hoặc 8 bit)
15	LED+	Đèn nền (Backlight +)
16	LED-	Đèn nền (Backlight -)

CHƯƠNG 2. THIẾT KẾ SẢN PHẨM

2.1 Tổng quan ngôn ngữ C

C là một ngôn ngữ lập trình tương đối nhỏ gọn vận hành gần với phần cứng và nó giống với ngôn ngữ Assembly (ASM) hơn hầu hết các ngôn ngữ bậc cao. Hơn thế, C đôi khi được đánh giá như là "có khả năng di động", cho thấy sự khác nhau quan trọng giữa nó với ngôn ngữ bậc thấp như là Assembly, đó là việc mã C có thể được dịch và thi hành trong hầu hết các máy tính, hơn hẳn các ngôn ngữ hiện tại trong khi đó thì Assembly chỉ có thể chạy trong một số máy tính đặc biệt. Vì lý do này C được xem là ngôn ngữ bậc trung.

C đã được tạo ra với một mục tiêu là làm cho nó thuận tiện để viết các chương trình lớn với số lỗi ít hơn trong mẫu hình lập trình thủ tục mà lại không đặt gánh nặng lên vai người viết ra trình dịch C, là những người bẽ bộn với các đặc tả phức tạp của ngôn ngữ. Cuối cùng C có thêm những chức năng sau:

- Một ngôn ngữ cốt lõi đơn giản, với các chức năng quan trọng chẳng hạn như là những hàm hay việc xử lý tập tin sẽ được cung cấp bởi các bộ thư viện các thủ tục.
- Tập trung trên mẫu hình lập trình thủ tục, với các phương tiện lập trình theo kiểu cấu trúc.
- Một hệ thống kiểu đơn giản nhằm loại bỏ nhiều phép toán không có ý nghĩa thực dụng.
- Dùng ngôn ngữ tiền xử lý, tức là các câu lệnh tiền xử lý C, cho các nhiệm vụ như là định nghĩa các macro và hàm chứa nhiều tập tin mã nguồn (bằng cách dùng câu lệnh tiền xử lý dạng `#include` chẳng hạn)
- Mức thấp của ngôn ngữ cho phép dùng tới bộ nhớ máy tính qua việc sử dụng kiểu dữ liệu pointer.
- Số lượng từ khóa rất nhỏ gọn.
- Các tham số được đưa vào các hàm bằng giá trị, không bằng địa chỉ.
- Hàm các con trỏ cho phép hình thành một nền tảng ban đầu cho tính đóng và tính đa hình.
- Hỗ trợ các bản ghi hay các kiểu dữ liệu kết hợp do người dùng từ khóa định nghĩa struct cho phép các dữ liệu liên hệ nhau có thể được tập hợp lại và được điều chỉnh như là toàn bộ.

Một số chức năng khác mà C không có (hay còn thiếu) nhưng có thể tìm thấy ở các ngôn ngữ khác bao gồm:

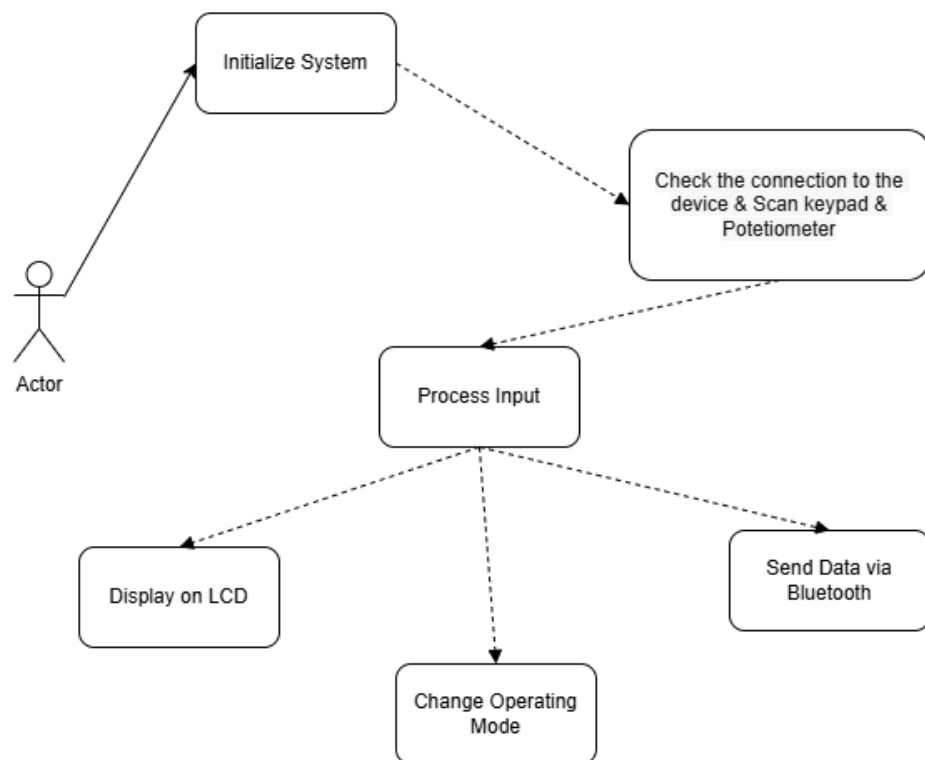
- An toàn kiểu
- Tự động Thu dọn rác
- Các lớp hay các đối tượng cùng với các ứng xử của chúng (xem thêm OOP)

- Các hàm lồng nhau
- Lập trình tiêu bản hay Lập trình phổ dụng
- Quá tải và Quá tải toán tử
- Các hỗ trợ cho đa luồng, đa nhiệm và mạng

Mặc dù C còn thiếu nhiều chức năng hữu ích nhưng lý do quan trọng để C được chấp nhận vì nó cho phép các trình dịch mới được tạo ra một cách nhanh chóng trên các nền tảng mới và vì nó cho phép người lập trình dễ kiểm soát được những gì mà chương trình (do họ viết) thực thi. Đây là điểm thường làm cho mã C chạy hiệu quả hơn các ngôn ngữ khác. Thường thì chỉ có ngôn ngữ ASM chinh bằng tay chạy nhanh hơn (ngôn ngữ C), bởi vì ASM kiểm soát được toàn bộ máy. Mặc dù vậy, với sự phát triển các trình dịch C, và với sự phức tạp của các CPU hiện đại có tốc độ cao, C đã dần thu nhỏ khác biệt về tốc độ này.

2.2 Mô hình hóa hệ thống

2.2.1 Sơ đồ Use-case



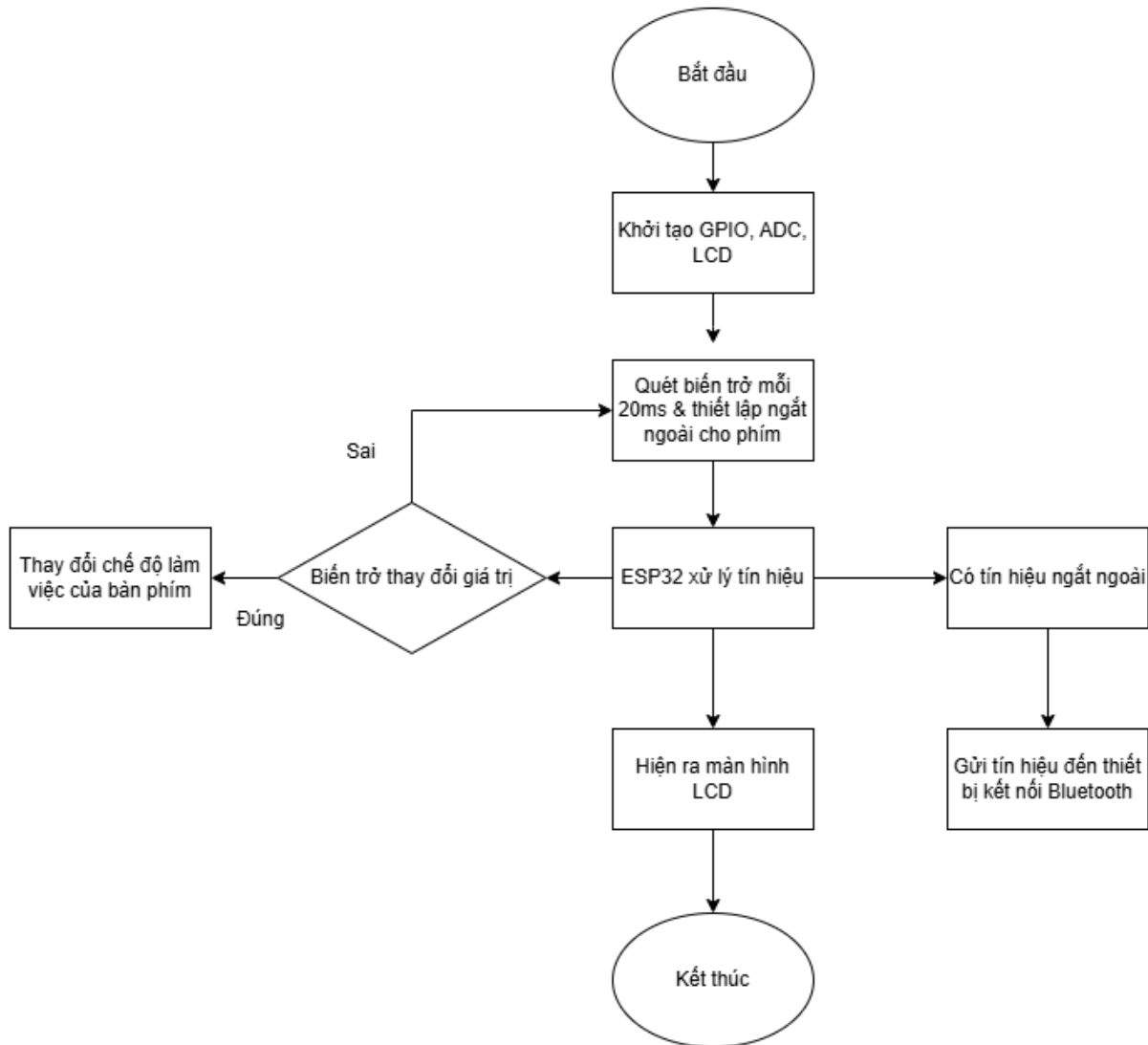
Hình 2.1: Hình ảnh sơ đồ Use-case hệ thống

Các chức năng chính của hệ thống được xây dựng theo ý tưởng thiết kế, bao gồm:

- **Khởi tạo hệ thống:** Bắt đầu quá trình vận hành hệ thống.
- **Kiểm tra kết nối thiết bị và quét bàn phím & biến áp:** Đảm bảo các thiết bị được kết nối đúng cách và kiểm tra các tín hiệu từ bàn phím và biến áp.
- **Xử lý đầu vào:** Nhận và xử lý thông tin đầu vào từ người dùng hoặc thiết bị.

- **Hiển thị trên màn hình LCD:** Xuất thông tin ra màn hình hiển thị.
- **Gửi dữ liệu qua Bluetooth:** Truyền dữ liệu đến thiết bị khác qua kết nối Bluetooth.
- **Thay đổi chế độ hoạt động:** Chuyển đổi giữa các chế độ làm việc của hệ thống.

2.2.2 Sơ đồ thuật toán



Hình 2.2: Hình ảnh sơ đồ thuật toán hệ thống

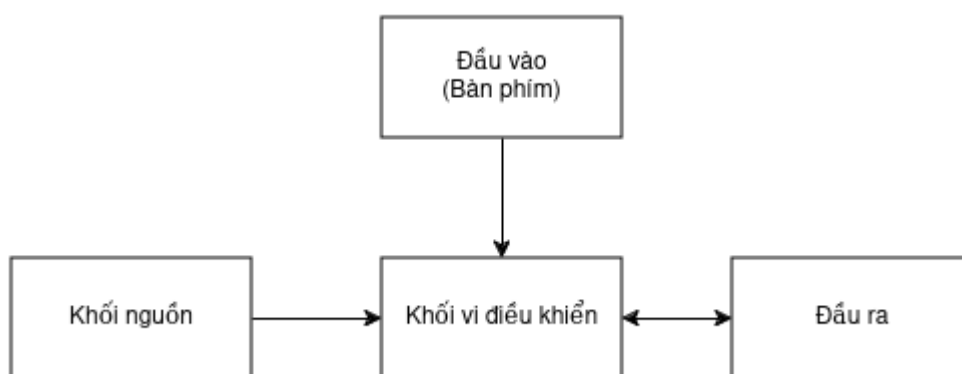
Sơ đồ thuật toán của hệ thống mô tả quy trình hoạt động chính của thiết bị, bao gồm các bước tuần tự từ khi khởi động cho đến khi kết thúc. Quy trình cụ thể như sau:

1. **Bắt đầu:** Hệ thống được khởi động và thực hiện các bước khởi tạo.
2. **Khởi tạo:** Thiết lập các thành phần cơ bản như GPIO, ADC và LCD để chuẩn bị cho quá trình vận hành.

3. **Quét biến trở và thiết lập ngắt:** Mỗi 20ms, hệ thống sẽ quét biến trở để kiểm tra sự thay đổi giá trị và đồng thời thiết lập cơ chế ngắt ngoài cho bàn phím.
4. **Kiểm tra giá trị biến trở:** Nếu giá trị của biến trở thay đổi, hệ thống sẽ thực hiện thay đổi chế độ làm việc của bàn phím.
5. **Xử lý tín hiệu:** Các tín hiệu được gửi đến ESP32 để xử lý.
6. **Phát hiện tín hiệu ngắt ngoài:** Nếu có tín hiệu ngắt ngoài, hệ thống sẽ gửi dữ liệu đến thiết bị kết nối Bluetooth.
7. **Hiển thị thông tin:** Dữ liệu sau khi xử lý sẽ được hiển thị lên màn hình LCD để cung cấp thông tin cho người dùng.
8. **Kết thúc:** Kết thúc quy trình vận hành.

Sơ đồ thuật toán đảm bảo tính tuần tự, rõ ràng và dễ dàng theo dõi, giúp hệ thống hoạt động hiệu quả và đáp ứng các yêu cầu chức năng đã đề ra.

2.2.3 Sơ đồ khối của hệ thống



Hình 2.3: Hình ảnh sơ đồ khối của hệ thống

1. Khối nguồn

Chức năng: Cung cấp nguồn điện cho toàn bộ hệ thống, bao gồm vi điều khiển, thiết bị đầu vào, thiết bị đầu ra. Khối nguồn sử dụng 2 viên pin 18650 mắc song song để tăng dung lượng, sau đó 3.7V sẽ qua mạch boost để tăng áp thành 5V cấp nguồn cho các linh kiện trong mạch.

2. Vi điều khiển (ESP32)

Chức năng: Là trung tâm xử lý của hệ thống. Vi điều khiển nhận và xử lý tín hiệu từ các phím bấm và biến trở, điều khiển các thiết bị đầu ra như màn hình LCD. Tất cả các logic, tính toán và kiểm soát thời gian của hệ thống đều được xử lý tại đây.

3. Thiết bị đầu vào (Bàn phím)

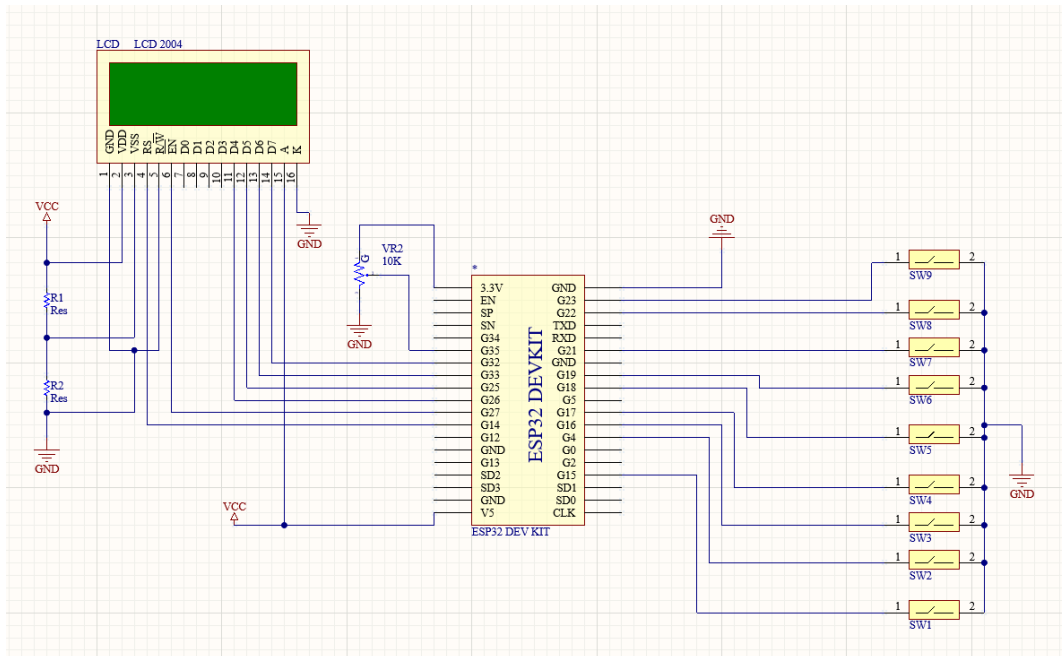
Chức năng: Nhận tín hiệu từ người dùng, như nhấn phím hoặc thay đổi chế độ thông qua biến trở. Các tín hiệu đầu vào này được gửi tới vi điều khiển để xử lý và thực hiện các chức năng tương ứng.

4. Khối đầu ra (LCD 20x4)

Chức năng: Màn hình LCD hiển thị trạng thái thời gian thực như phím đang bấm, chế độ hoạt động hoặc thông tin debug.

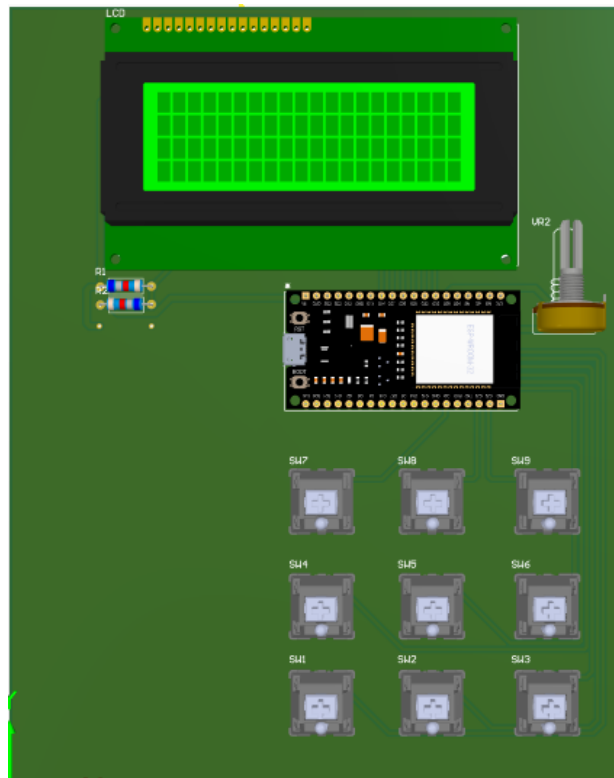
2.3 Thiết kế mạch bằng phần mềm Altium

Sơ đồ nguyên lý trên phần mềm Altium.



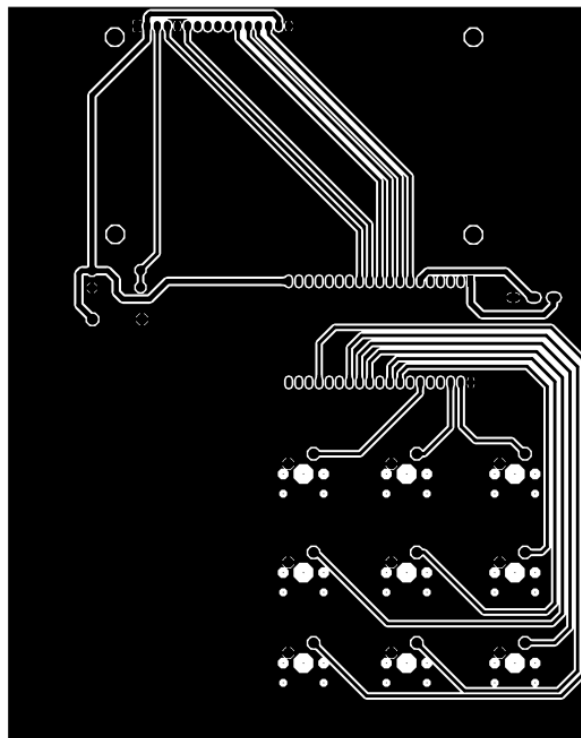
Hình 2.4 Sơ đồ nguyên lý hệ thống

Mô hình 3D trên Altium



Hình 2.5: Hình ảnh 3D hệ thống mô phỏng bằng Altium

Đồ đồng mạch in thủ công



Hình 2.6: Hình ảnh mạch in hệ thống

CHƯƠNG 3. KẾT QUẢ SẢN PHẨM

