TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ



BÁO CÁO MÔN HỌC LẬP TRÌNH PYTHON CHO HỆ THỐNG NHÚNG

ĐỀ TÀI: **NHÀ THÔNG MINH**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | DOÃN CAO DANH - 21CE074 |
| Lớp: | 21CE2 |
| Giảng viên hướng dẫn: | TS. Nguyễn Vũ Anh Quang |

Đà Nẵng, ngày 25 tháng 12, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

KHOA KỸ THUẬT MÁY TÍNH VÀ ĐIỆN TỬ



BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH

ĐỀ TÀI: **NHÀ THÔNG MINH**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | DOÃN CAO DANH - 21CE074 |
|  |
|  |  |
| Lớp: | 21CE2 |
| Giảng viên hướng dẫn: | TS. Nguyễn Vũ Anh Quang |

Đà Nẵng, ngày 25 tháng 12, năm 2024

**NHẬN XÉT**

(Nhận xét của giảng viên hướng dẫn)

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

**Chữ Ký**

Nguyễn Vũ Anh Quang

LỜI CẢM ƠN

Để thực hiện và hoàn thành tốt đề tài này, chúng em đã nhận được sự giúp đỡ và hướng dẫn rất tận tình của các thầy cô thuộc Khoa Kỹ thuật máy tính và điện tử–trường đại học Công Nghệ Thông Tin và Truyền Thông Việt Hàn. Em xin cảm ơn các thầy cô thuộc bộ môn chuyên ngành đã cung cấp cho chúng em các thông tin, kiến thức vô cùng quý báu và cần thiết trong suốt thời gian quá để em có thể thực hiện và hoàn thành dự án của mình. Đặc biệt em xin chân thành cảm ơn thầy Nguyễn Vũ Anh Quang đã trực tiếp hướng dẫn chúng em trong thời gian thực hiện đồ án này.

Cuối cùng, xin chân thành cảm ơn các bạn trong ngành công nghệ thông tin đã ủng hộ, giúp đỡ, chia sẻ kiến thức, kinh nghiệm và tài liệu có được giúp chúng tôi trong quá trình nghiên cứu và thực hiện đề tài. Do giới hạn về mặt thời gian và kiến thức cũng như kinh nghiệm thực tiễn nên đề tài không tránh khỏi những sai xót. Em rất mong nhận được sự thông cảm của quý thầy cô và mong đón nhận những góp ý của thầy cô và các bạn.

Chúng em xin chân thành cảm ơn!

`

**LỜI MỞ ĐẦU**

Đề tài " **Nhà Thông Minh**" là một hệ thống nhà ở được tích hợp các công nghệ hiện đại, cho phép tự động hóa và điều khiển từ xa các thiết bị trong ngôi nhà, như chiếu sáng, điều khiển nhiệt độ và các thiết bị thông qua internet. Mục đích của nhà thông minh không chỉ là mang lại sự tiện nghi, mà còn giúp nâng cao an toàn và tối ưu hóa quản lý không gian sống. Nhờ vào sự phát triển của Internet of Things (IoT) và các nền tảng kết nối, nhà thông minh đang dần trở thành xu hướng trong đời sống hiện đại, đáp ứng nhu cầu ngày càng cao của người dùng về sự thoải mái, hiệu quả và bảo mật.

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc186165103)

[DANH MỤC HÌNH ẢNH 9](#_Toc186165104)

[CHƯƠNG I: GIỚI THIỆU 1](#_Toc186165105)

[1.1 Tổng quan 1](#_Toc186165106)

[1.1.1 Giới thiệu 1](#_Toc186165107)

[CHƯƠNG II: TÌM HIỂU CÔNG CỤ VÀ LINH KIỆN 3](#_Toc186165108)

[2.1 Công cụ 3](#_Toc186165109)

[2.1.1 Ngôn ngữ C++ 3](#_Toc186165110)

[2.1.2 Ngôn ngữ Python 6](#_Toc186165111)

[2.1.3 Telegram 22](#_Toc186165112)

[2.1.4 Chatbot Telegram 24](#_Toc186165113)

[2.1 Linh kiện 26](#_Toc186165114)

[2.2.1 ESP32 26](#_Toc186165115)

[2.2.2 Module Relay 28](#_Toc186165116)

[2.2.3 Khóa cửa chốt điện tử 12v 28](#_Toc186165117)

[2.2.4 Cảm Biến 29](#_Toc186165118)

[2.2.5 Màn hình hiển thị 31](#_Toc186165119)

[2.2.6 Buzzer 32](#_Toc186165120)

[2.2.7 Đèn 32](#_Toc186165121)

[2.2.8 Điện trở 32](#_Toc186165122)

[2.2.9 Quạt 5v 33](#_Toc186165123)

[CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ SẢN PHẨM 34](#_Toc186165124)

[3.1 Phân tích hệ thống 34](#_Toc186165125)

[3.1.1 Mục đích của hệ thống 34](#_Toc186165126)

[3.1.2 Yêu cầu của hệ thống 34](#_Toc186165127)

[3.2 Thiết kế và hoàn thiện sản phẩm 34](#_Toc186165128)

[3.2.1 Thiết kế 34](#_Toc186165129)

[3.2.2 Hoàn thiện sản phẩm 50](#_Toc186165132)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 55](#_Toc186165134)

[1 Kết luận 55](#_Toc186165135)

[2 Ứng dụng thực tế của mạch: 55](#_Toc186165136)

[3 Hướng phát triển 55](#_Toc186165137)

[TÀI LIỆU THAM KHẢO 56](#_Toc186165138)

# CHƯƠNG I: GIỚI THIỆU

1.1 Tổng quan



Hình 1. Nhà thông minh

1.1.1 Giới thiệu

Nhà thông minh là một khái niệm đang trở thành xu hướng nổi bật trong thế giới công nghệ hiện đại. Đây là một ngôi nhà được trang bị các thiết bị và hệ thống tự động hóa có khả năng kết nối với nhau qua Internet, cho phép người dùng điều khiển và giám sát các thiết bị trong nhà từ xa, thông qua ứng dụng di động hoặc các thiết bị trợ lý ảo. Các thiết bị trong nhà thông minh có thể tự động điều chỉnh theo các tác động từ môi trường hoặc theo thói quen và sở thích của người sử dụng, mang đến sự tiện lợi, tiết kiệm năng lượng và cải thiện chất lượng cuộc sống.

1.1.2 Mục tiêu

Mục tiêu chính khi làm hệ thống nhà thông minh là:

- **Thiết bị thông minh**: Gồm các thiết bị như cửa ra vào, đèn, quạt và các thiết bị khác có thể được kết nối internet, điều khiển từ xa và có thể xem thông tin trên màn hình.

- **Cảm biến**: Các thiết bị cảm biến khí gas, cảm biến độ ẩm giúp quản lý sự cố, giảm thiểu thiệt hại cho ngôi nhà.

- **Hệ thống tự động hóa**: Hệ thống cho phép các thiết bị hoạt động tự động hoặc hoạt động theo lịch trình được cài đặt sẵn.

- **WebServer**: Cho phép người sử dụng có thể điều khiển từ xa.

- **Độ Tin Cậy Cao**: Hệ thống cảnh báo phải hoạt động đúng cách và có độ tin cậy cao để người dùng có thể tin tưởng vào nó trong mọi tình huống.

Kế hoạch thực hiện

**- Nghiên cứu và xác định yêu cầu**

**Nghiên cứu thiết bị**: Tìm hiểu các thiết bị thông minh như đèn, cảm biến, quạt, ESP-32CAM

**Xác định yêu cầu**: Lên danh sách nhu cầu về tự động hóa, tiết kiệm năng lượng.

**- Thiết kế hệ thống**

**Thiết kế hệ thống**: Lập sơ đồ kết nối thiết bị và lựa chọn phần mềm điều khiển.

**Mua sắm thiết bị**: Chọn thiết bị từ các nhà cung cấp uy tín.

**Phát triển WebSever.**

**- Cài đặt và triển khai**

**Lắp đặt thiết bị**: Cài đặt các thiết bị như đèn, cảm biến, quạt, espcam.

**Cấu hình hệ thống**: Thiết lập các thiết bị và nối dây.

**- Kiểm tra và đánh giá**

**Kiểm tra hoạt động**: Đảm bảo hệ thống hoạt động ổn định, kiểm tra tính năng.

**Đánh giá hiệu quả**: Lấy phản hồi từ người dùng về tính tiện lợi.

# CHƯƠNG II: TÌM HIỂU CÔNG CỤ VÀ LINH KIỆN

2.1 Công cụ

2.1.1 Ngôn ngữ C++

2.1.1.1 Khái niệm

C++ là một ngôn ngữ lập trình bậc cao được phát triển bởi **Bjarne Stroustrup** vào đầu những năm 1980 tại Bell Labs. C++ được thiết kế như một sự mở rộng của ngôn ngữ **C**, bổ sung các tính năng lập trình hướng đối tượng và những cải tiến khác để hỗ trợ phát triển phần mềm hiện đại.

C++ là một ngôn ngữ **đa năng**, được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển hệ thống, phần mềm nhúng, trò chơi điện tử, trình biên dịch, và ứng dụng doanh nghiệp.



Hình 2. Ngôn ngữ C++

2.1.1.2 Lịch sữ hình thành

Bjarne Stroustrup là người sáng lập ngôn ngữ C++. Ông bắt đầu phát triển C++ vào năm 1979 tại Bell Labs với mục tiêu mở rộng ngôn ngữ C để hỗ trợ lập trình hướng đối tượng. C++ được thiết kế để kết hợp những ưu điểm của C (như khả năng truy cập phần cứng trực tiếp và hiệu suất cao) với các tính năng mới như lớp (class) và kế thừa (inheritance) để hỗ trợ lập trình hướng đối tượng.

1983 - Ngôn ngữ C++ ra đời với các tính năng cơ bản như class, constructor, destructor, và inheritance.

1985 - Phiên bản chính thức đầu tiên của C++ (C++ 1.0) được phát hành.

1989 - C++ 2.0 ra mắt với tính năng multiple inheritance (kế thừa đa hình) và abstract classes (lớp trừu tượng).

1990 - C++ 2.1 giới thiệu Standard Template Library (STL), một thư viện chuẩn cung cấp các cấu trúc dữ liệu và thuật toán.

1998 - C++98 chính thức được tiêu chuẩn hóa bởi ISO với một số cải tiến và chỉnh sửa.

2003 - C++03 là bản cập nhật sửa lỗi của C++98 mà không thay đổi nhiều tính năng.

2011 - C++11 giới thiệu nhiều tính năng mới như auto, nullptr, lambda expressions, và concurrency (đồng thời hóa).

2014 - C++14 ra mắt với các cải tiến nhỏ, sửa lỗi và cải thiện tính năng của C++11.

2017 - C++17 cải thiện ngôn ngữ với tính năng như structured bindings, optional, và if constexpr.

2020 - C++20 giới thiệu các tính năng lớn như concepts, ranges, coroutines và các cải tiến cho thư viện chuẩn.

2023 - C++23 ra mắt với các tính năng mới như reflection (phản chiếu) và các cải tiến nhỏ giúp tăng hiệu suất và khả năng tương thích.

2.1.1.3 Mục đích sử dụng

**Quản lý bộ nhớ thủ công:**

C++ cung cấp quyền kiểm soát bộ nhớ trực tiếp cho lập trình viên, cho phép họ cấp phát và giải phóng bộ nhớ khi cần thiết.

Các từ khóa new và delete giúp cấp phát và giải phóng bộ nhớ động, điều này giúp tối ưu hóa hiệu suất và sử dụng tài nguyên hệ thống một cách hiệu quả.

Lập trình hướng đối tượng (OOP):

C++ hỗ trợ lập trình hướng đối tượng, giúp mô hình hóa các đối tượng và hành vi trong phần mềm.

C++ hỗ trợ các nguyên lý của OOP như đóng gói, kế thừa, đa hình, và trừu tượng, giúp xây dựng phần mềm dễ bảo trì và mở rộng.

Lớp và đối tượng là các khái niệm cơ bản trong C++, giúp tổ chức mã nguồn và tái sử dụng mã hiệu quả.

Tính đa hình (Polymorphism):

C++ hỗ trợ tính đa hình, cho phép các hàm hoặc phương thức có thể hoạt động với nhiều kiểu dữ liệu khác nhau.

C++ sử dụng hàm ảo (virtual functions) và chỉ thị kiểu động (dynamic typing) để đạt được tính đa hình, giúp chương trình linh hoạt hơn khi thực thi.

**Tính kế thừa (Inheritance):**

Tính kế thừa cho phép tạo ra các lớp con kế thừa các thuộc tính và phương thức của lớp cha. Điều này giúp tăng tính tái sử dụng mã và giảm sự trùng lặp trong chương trình.

C++ hỗ trợ kế thừa đơn và kế thừa đa (kế thừa từ nhiều lớp cha).Đóng gói (Encapsulation):

Đóng gói giúp bảo vệ dữ liệu của đối tượng khỏi việc truy cập trái phép và giảm sự phụ thuộc giữa các thành phần của phần mềm.

C++ sử dụng thành phần public, private, và protected để kiểm soát quyền truy cập vào các thành viên dữ liệu và phương thức của lớp.

**Xử lý lỗi (Error Handling):**

C++ sử dụng cơ chế ngoại lệ (exceptions) để xử lý lỗi. Khi một lỗi xảy ra, chương trình có thể ném ra một ngoại lệ, và các phần của chương trình có thể bắt và xử lý ngoại lệ này.

Các từ khóa try, catch, và throw được sử dụng để làm việc với ngoại lệ.

**Tính tổng quát (Generic Programming):**

C++ hỗ trợ lập trình tổng quát thông qua các template, cho phép xây dựng các hàm và lớp có thể làm việc với bất kỳ kiểu dữ liệu nào mà không cần thay đổi mã nguồn.

Templates giúp tăng tính tái sử dụng mã và giảm sự trùng lặp trong phần mềm.

Lập trình theo kiểu hàm (Function Overloading và Operator Overloading):

C++ hỗ trợ overloading (nạp chồng) các hàm và toán tử, cho phép sử dụng tên hàm hoặc toán tử giống nhau nhưng với các tham số hoặc kiểu dữ liệu khác nhau. Điều này giúp làm mã dễ đọc và dễ hiểu hơn.

**Tối ưu hóa hiệu suất:**

C++ được thiết kế để có thể xử lý hiệu quả các tác vụ có yêu cầu về hiệu suất cao, chẳng hạn như xử lý bộ nhớ, tính toán và xử lý dữ liệu. Ngôn ngữ này không sử dụng nhiều tự động hóa như các ngôn ngữ khác, giúp lập trình viên tối ưu hóa mã một cách chi tiết.

C++ có các công cụ và thư viện mạnh mẽ giúp tối ưu hóa bộ biên dịch, chẳng hạn như tối ưu hóa vòng lặp, các phép toán bộ nhớ, và các thuật toán xử lý dữ liệu.

**Tính năng đa luồng (Multithreading):**

C++ hỗ trợ đa luồng, cho phép thực thi các tác vụ song song. Điều này giúp cải thiện hiệu suất của các chương trình phức tạp hoặc ứng dụng yêu cầu xử lý thời gian thực.

**Tính tương thích ngược (Backward Compatibility):**

C++ được xây dựng để tương thích với C, điều này có nghĩa là các mã nguồn C có thể được biên dịch và thực thi trong môi trường C++ mà không gặp phải vấn đề tương thích. Điều này giúp C++ trở thành một sự lựa chọn tuyệt vời cho các dự án cần tích hợp với mã nguồn C. Thư viện chuẩn (Standard Library):

C++ đi kèm với một thư viện chuẩn phong phú, bao gồm các cấu trúc dữ liệu như vector, list, map, và các hàm thao tác với tệp, xử lý chuỗi, toán học và các chức năng khác. Thư viện chuẩn giúp tăng tốc phát triển phần mềm và giảm thiểu lỗi

2.1.1.4 Đặc điểm của ngôn ngữ c++

Phát triển: Bắt đầu từ C++, ngôn ngữ này nhanh chóng phát triển với các tính năng mới như lớp (class), kế thừa (inheritance), và đa hình (polymorphism), giúp hỗ trợ lập trình hướng đối tượng (OOP).

Mục đích sử dụng: C++ được sử dụng trong phát triển phần mềm hệ thống, ứng dụng máy tính, game, mô phỏng, tài chính, và các hệ thống thời gian thực.

**Đặc điểm:**

Lập trình hướng đối tượng: Hỗ trợ các tính năng như lớp, đối tượng, kế thừa, đa hình và đóng gói.

Quản lý bộ nhớ: Cung cấp khả năng quản lý bộ nhớ bằng tay, cho phép tối ưu hóa hiệu suất.

Hiệu suất cao: C++ cung cấp khả năng tối ưu hóa và kiểm soát tài nguyên hệ thống, thích hợp cho các ứng dụng yêu cầu tốc độ xử lý cao.

Ứng dụng: C++ được dùng trong nhiều lĩnh vực, từ phát triển hệ điều hành, game, phần mềm tài chính, khoa học, mô phỏng, đến các ứng dụng mạng và trí tuệ nhân tạo.

2.1.2 Ngôn ngữ Python

2.1.2.1 Khái niệm

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.



Hình 2. Ngôn ngữ Python

2.1.2.2 Lịch sử phát triển

Guido Van Rossum, một lập trình viên máy tính ở Hà Lan, đã tạo ra Python. Ông bắt đầu phát triển ngôn ngữ này vào năm 1989 tại Centrum Wiskunde & Informatica (CWI), ban đầu chỉ là một dự án tiêu khiển để giúp ông giết thời gian trong dịp Giáng sinh. Tên gọi của ngôn ngữ này được lấy cảm hứng từ chương trình truyền hình Monty Python's Flying Circus của đài BBC vì Guido Van Rossum là một “fan cứng” của chương trình này.

* Guido Van Rossum cho ra mắt phiên bản đầu tiên của ngôn ngữ Python (phiên bản 0.9.0) vào năm 1991. Ngôn ngữ này đã bao gồm các tính năng hữu ích như một số kiểu dữ liệu và hàm để xử lý lỗi.
* Python 1.0 đã được ra mắt vào năm 1994 với các hàm mới để dễ dàng xử lý danh sách dữ liệu, chẳng hạn như ánh xạ, lọc và lược bỏ.
* Python 2.0 đã được ra mắt vào ngày 16 tháng 10 năm 2000, với các tính năng hữu ích mới cho lập trình viên, chẳng hạn như hỗ trợ ký tự Unicode và cách xử lý chi tiết một danh sách nhanh chóng hơn.
* Python 3.0 đã được ra mắt vào ngày 3 tháng 12 năm 2008. Phiên bản này bao gồm các tính năng như hàm in và hỗ trợ nhiều hơn cho việc phân chia số và xử lý lỗi.

2.1.2.3 Mục đích sử dụng của ngôn ngữ Python

Ngôn ngữ Python được sử dụng nhiều trong lĩnh vực phát triển ứng dụng, bao gồm những ví dụ sau:

**Phát triển web phía máy chủ:**

Phát triển web phía máy chủ bao gồm những hàm backend phức tạp mà các trang web thực hiện để hiển thị thông tin cho người dùng. Ví dụ: các trang web phải tương tác với cơ sở dữ liệu, giao tiếp với các trang web khác và bảo vệ dữ liệu khi truyền qua mạng.

Python hữu ích trong việc lập trình mã phía máy chủ bởi vì ngôn ngữ này cung cấp nhiều thư viện bao gồm mã viết sẵn cho các hàm backend phức tạp. Các nhà phát triển cũng sử dụng một loạt các khung Python cung cấp tất cả những công cụ cần thiết để xây dựng ứng dụng web một cách nhanh chóng và dễ dàng hơn. Ví dụ: các nhà phát triển có thể tạo ứng dụng web khung trong nháy mắt bởi vì họ không cần phải lập trình nó từ đầu. Sau đó, họ có thể kiểm tra ứng dụng web này bằng cách sử dụng các công cụ kiểm thử của khung, mà không cần phụ thuộc vào những công cụ kiểm thử bên ngoài.

**Tự động hóa bằng các tập lệnh Python:**

Ngôn ngữ tập lệnh là một ngôn ngữ lập trình tự động hóa các tác vụ mà thường được con người thực hiện. Các lập trình viên thường xuyên sử dụng các tập lệnh Python để tự động hóa nhiều tác vụ hàng ngày như:

* Đổi tên một số lượng lớn tệp cùng lúc
* Chuyển đổi một tệp sang một loại tệp khác
* Loại bỏ các từ trùng lặp trong tệp văn bản
* Thực hiện các phép tính toán cơ bản
* Gửi email
* Tải xuống nội dung
* Thực hiện phân tích nhật ký cơ bản
* Tìm kiếm lỗi trong nhiều tệp

**Khoa học dữ liệu và máy học:**

Khoa học dữ liệu trích xuất thông tin quý giá từ dữ liệu và máy học (ML) dạy máy tính tự động học hỏi từ dữ liệu và đưa ra các dự đoán chính xác. Các nhà khoa học dữ liệu sử dụng Python cho các tác vụ khoa học dữ liệu sau:

* Sửa và loại bỏ dữ liệu không chính xác, hay còn được gọi là làm sạch dữ liệu
* Trích xuất và chọn lọc các đặc điểm đa dạng của dữ liệu
* Ghi nhãn dữ liệu gán tên có ý nghĩa cho dữ liệu
* Tìm các số liệu thống kê khác nhau từ dữ liệu
* Trực quan hóa dữ liệu bằng cách sử dụng các biểu đồ và đồ thị, chẳng hạn như biểu đồ đường, biểu đồ cột, biểu đồ tần suất và biểu đồ tròn

Các nhà khoa học dữ liệu sử dụng những thư viện ML của Python để đào tạo các mô hình ML và xây dựng các công cụ phân loại giúp phân loại dữ liệu một cách chính xác. Các chuyên gia từ nhiều lĩnh vực sử dụng những công cụ phân loại dựa trên Python để thực hiện các tác vụ phân loại, chẳng hạn như phân loại hình ảnh, văn bản cũng như lưu lượng truy cập mạng, nhận dạng giọng nói và nhận diện khuôn mặt. Các nhà khoa học dữ liệu cũng sử dụng Python cho học sâu, một kỹ thuật ML nâng cao.

**Phát triển phần mềm:**

Các nhà phát triển phần mềm thường sử dụng Python cho những tác vụ phát triển và ứng dụng phần mềm khác nhau, chẳng hạn như:

Theo dõi lỗi trong mã của phần mềm

Tự động xây dựng phần mềm

Đảm nhận quản lý dự án phần mềm

Phát triển nguyên mẫu phần mềm

Phát triển các ứng dụng máy tính bằng cách sử dụng những thư viện Giao diện đồ họa người dùng (GUI)

Phát triển từ các trò chơi văn bản đơn giản cho đến những trò chơi điện tử phức tạp

Tự động hóa kiểm thử phần mềm

Kiểm thử phần mềm là quy trình kiểm tra xem kết quả thực tế từ phần mềm có khớp với kết quả mong đợi không để đảm bảo rằng phần mềm không có lỗi.

Các nhà phát triển sử dụng khung kiểm thử đơn vị Python, chẳng hạn như Unittest, Robot và PyUnit, để kiểm thử các hàm do họ viết.

Các kỹ sư kiểm thử phần mềm sử dụng Python để viết các trường hợp kiểm thử cho nhiều tình huống khác nhau. Ví dụ: họ sử dụng ngôn ngữ này để kiểm thử giao diện người dùng của một ứng dụng web, nhiều thành phần của phần mềm và những tính năng mới.

Các nhà phát triển có thể sử dụng một số công cụ để tự động chạy tập lệnh kiểm thử. Những công cụ này có tên gọi là công cụ Tích hợp liên tục/Triển khai liên tục (CI/CD). Các kỹ sư kiểm thử phần mềm cũng như những nhà phát triển sử dụng các công cụ CI/CD như Travis CI và Jenkins để tự động hóa quy trình kiểm thử. Công cụ CI/CD tự động chạy các tập lệnh kiểm thử Python và báo cáo kết quả kiểm thử bất kỳ khi nào nhà phát triển thêm vào những dòng mã mới.

2.1.2.4 Đặc điểm của ngôn ngữ Python

Các đặc điểm sau tạo nên sự độc đáo của ngôn ngữ lập trình Python:

*Python là một ngôn ngữ thông dịch:*

Python là một ngôn ngữ thông dịch, điều này nghĩa là ngôn ngữ này trực tiếp chạy từng dòng mã. Nếu có lỗi trong mã chương trình, nó sẽ ngừng chạy. Do đó, lập trình viên có thể nhanh chóng tìm ra lỗi trong đoạn mã.

*Python là một ngôn ngữ dễ sử dụng:*

Python sử dụng từ ngữ giống trong tiếng Anh. Không giống như các ngôn ngữ lập trình khác, Python không sử dụng dấu ngoặc ôm. Thay vào đó, ngôn ngữ này sử dụng thụt đầu dòng.

*Python là một ngôn ngữ linh hoạt:*

Các lập trình viên không cần phải khai báo loại biến khi viết mã bởi vì Python sẽ xác định chúng vào thời điểm chạy. Vì vậy, bạn có thể viết các chương trình Python một cách nhanh chóng hơn.

*Python là một ngôn ngữ cấp cao:*

Python gần gũi với ngôn ngữ con người hơn các ngôn ngữ lập trình khác. Do đó, các lập trình viên không cần phải lo lắng về những chức năng cơ bản của nó như kiến trúc và quản lý bộ nhớ.

*Python là một ngôn ngữ lập trình hướng đối tượng:*

Python coi mọi thứ đều là đối tượng, nhưng ngôn ngữ này cũng hỗ trợ các phương thức lập trình khác như lập trình hàm và lập trình cấu trúc.

2.1.2.5 Những lợi ích ngôn ngữ Python đem lại

Những lợi ích của Python bao gồm:

Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.

Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.

Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.

Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.

Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.

Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.

Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

2.1.2.6 Thư viện Python

Thư viện là một tập hợp các mã thường xuyên được sử dụng mà các nhà phát triển có thể bao gồm trong những chương trình Python của họ để không phải lập trình từ đầu. Theo mặc định, Python đi kèm với Thư viện chuẩn, chứa rất nhiều các hàm có thể tái sử dụng. Ngoài ra, hơn 137.000 thư viện Python có sẵn cho các ứng dụng khác nhau, bao gồm phát triển web, khoa học dữ liệu và máy học (ML).

*Matplotlib:*

Các nhà phát triển sử dụng Matplotlib để hiển thị dữ liệu dưới dạng đồ họa hai và ba chiều (2D và 3D) chất lượng cao. Thư viện này thường được sử dụng trong các ứng dụng khoa học. Với Matplotlib, bạn có thể trực quan hóa dữ liệu bằng cách hiển thị dữ liệu dưới dạng các biểu đồ khác nhau, chẳng hạn như biểu đồ cột và biểu đồ đường. Bạn cũng có thể hiển thị nhiều biểu đồ cùng lúc và các chi tiết đồ họa có thể được di chuyển qua mọi nền tảng.

*Pandas:*

Pandas cung cấp cấu trúc dữ liệu được tối ưu hóa và linh hoạt mà bạn có thể sử dụng để thao tác với dữ liệu chuỗi thời gian và dữ liệu có cấu trúc, chẳng hạn như bảng và nhóm. Ví dụ, bạn có thể sử dụng Pandas để đọc, ghi, hợp nhất, lọc và nhóm dữ liệu. Thư viện này được nhiều người sử dụng cho các tác vụ khoa học dữ liệu, phân tích dữ liệu và ML.

*NumPy:*

NumPy là một thư viện phổ biến mà các nhà phát triển sử dụng để dễ dàng tạo và quản lý nhóm, thao tác với các hình dạng logic và thực hiện các phép toán đại số tuyến tính. NumPy hỗ trợ tích hợp với nhiều ngôn ngữ như C và C ++.

Requests

Thư viện Requests cung cấp nhiều hàm hữu ích cần thiết để phát triển web. Bạn có thể sử dụng thư viện này để gửi các yêu cầu HTTP, bổ sung tiêu đề, thêm tham số URL, bổ sung dữ liệu và thực hiện nhiều tác vụ khác khi giao tiếp với các ứng dụng web.

*OpenCV-Python:*

OpenCV-Python là một thư viện mà các nhà phát triển sử dụng để xử lý hình ảnh cho các ứng dụng thị giác máy tính. Thư viện này cung cấp nhiều hàm cho các tác vụ xử lý hình ảnh như đọc và ghi hình ảnh cùng lúc, xây dựng môi trường 3D từ môi trường 2D cũng như chụp và phân tích hình ảnh từ video.

*Keras:*

Keras là thư viện mạng nơ-ron chuyên sâu của Python với khả năng hỗ trợ tuyệt vời cho việc xử lý dữ liệu, trực quan hóa và hơn thế nữa. Keras hỗ trợ nhiều mạng nơ-ron. Thư viện này có cấu trúc mô-đun mang lại sự linh hoạt cho việc lập trình các ứng dụng sáng tạo.

*Khung Python:*

Khung Python là tập hợp của các gói và mô-đun. Mô-đun là tập hợp của các mã liên quan và gói là tập hợp của các mô-đun. Các nhà phát triển có thể sử dụng khung Python để xây dựng các ứng dụng Python nhanh hơn vì họ không phải lo lắng về các chi tiết cấp thấp như cách giao tiếp trong ứng dụng web hoặc cách Python sẽ làm cho chương trình chạy nhanh hơn. Python có hai loại khung:

* Khung full-stack bao gồm hầu hết mọi thứ cần thiết để xây dựng một ứng dụng lớn.
* Khung micro là một khung cơ bản cung cấp các chức năng tối thiểu để xây dựng những ứng dụng Python đơn giản. Nó cũng cung cấp các tiện ích mở rộng nếu ứng dụng cần các hàm phức tạp hơn.

*Khung Python nào phổ biến nhất:*

Các nhà phát triển có thể sử dụng nhiều khung Python để giúp hoạt động phát triển của họ hiệu quả hơn, bao gồm các khung sau:

*Django:*

Django là một trong những khung web Python full-stack được sử dụng rộng rãi nhất để phát triển các ứng dụng web trên quy mô lớn. Khung này cung cấp một số tính năng hữu ích, bao gồm một máy chủ web để phát triển và kiểm thử, một công cụ mẫu để xây dựng giao diện người dùng của trang web và nhiều cơ chế bảo mật khác.

*Flask:*

Flask là một khung micro được sử dụng để phát triển các ứng dụng web nhỏ. Các đặc điểm của khung này bao gồm sự hỗ trợ nhiệt tình từ cộng đồng, tài liệu hữu ích, công cụ mẫu, tính năng kiểm thử đơn vị và máy chủ web được tích hợp sẵn. Nó cũng cung cấp các tiện ích mở rộng để hỗ trợ xác thực, các lớp ánh xạ cơ sở dữ liệu và bảo mật web.

*TurboGears:*

TurboGears là một khung được thiết kế để xây dựng các ứng dụng web một cách nhanh chóng và dễ dàng hơn. Sau đây là một số đặc điểm nổi bật của khung này:

* Cấu trúc bảng cơ sở dữ liệu cụ thể
* Các công cụ tạo và quản lý dự án
* Một công cụ mẫu để xây dựng cơ sở dữ liệu
* Một công cụ mẫu để tạo giao diện người dùng
* Các cơ chế bảo mật web

*Apache MXNet:*

Apache MXNet là một khung học sâu nhanh, linh hoạt và có thể điều chỉnh quy mô được các nhà phát triển sử dụng để xây dựng những nguyên mẫu nghiên cứu và ứng dụng học sâu. Khung này hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Java, C++, R và Perl. Nó cung cấp một bộ công cụ cũng như thư viện phong phú để hỗ trợ phát triển. Ví dụ: bạn có thể tìm thấy một cuốn sách tương tác về máy học (ML), bộ công cụ thị giác máy tính và các mô hình học sâu dành cho Xử lý ngôn ngữ tự nhiên (NLP) để xử lý ngôn ngữ tự nhiên, chẳng hạn như văn bản và lời nói.

*PyTorch:*

PyTorch là một khung cho ML được xây dựng dựa trên thư viện Torch, một thư viện ML nguồn mở khác. Các nhà phát triển sử dụng khung này cho những ứng dụng như NLP, robot và thị giác máy tính, tìm kiếm thông tin có ý nghĩa từ hình ảnh và video. Họ cũng sử dụng PyTorch để chạy những ứng dụng đó bằng CPU và GPU.

2.1.2.7 Môi trường phát triển ngôn ngữ Python

Python là một trong những ngôn ngữ lập trình phổ biến nhất thế giới nhờ cú pháp đơn giản, dễ học và khả năng ứng dụng đa dạng. Để phát triển các ứng dụng Python hiệu quả, việc lựa chọn môi trường phát triển (IDE – Integrated Development Environment) phù hợp đóng vai trò quan trọng. Dưới đây là phân tích chi tiết về các IDE phổ biến dành cho Python:

*PyCharm:*

* Nhà phát triển: JetBrains
* Loại: IDE mạnh mẽ cho Python
* Giới thiệu:

PyCharm là một trong những IDE phổ biến nhất và mạnh mẽ nhất dành cho Python. Được phát triển bởi JetBrains, PyCharm cung cấp nhiều tính năng cao cấp hỗ trợ phát triển ứng dụng Python, bao gồm auto-completion, debugging, và refactoring. Nó thích hợp cho các dự án lớn và phát triển phần mềm phức tạp.

* Tính năng nổi bật:

Tự động hoàn thành mã (Code Completion): Giúp tăng tốc quá trình lập trình bằng cách tự động gợi ý các phương thức, biến và lớp.

* Debugger mạnh mẽ: Hỗ trợ gỡ lỗi thông qua giao diện người dùng dễ sử dụng.

Hỗ trợ cho các framework và thư viện: PyCharm hỗ trợ rất tốt cho các framework như Django, Flask, và nhiều thư viện Python khác.

* Quản lý phiên bản Git: Tích hợp sẵn công cụ quản lý phiên bản Git, giúp theo dõi sự thay đổi của mã nguồn.
* Ưu điểm:

Tính năng mạnh mẽ, phù hợp cho các dự án lớn.

Tích hợp đầy đủ công cụ phát triển và gỡ lỗi.

Hỗ trợ phát triển web với Django và Flask.

* Nhược điểm:

Tốn tài nguyên hệ thống, có thể chậm trên máy tính cấu hình thấp.

Quá phức tạp cho người mới bắt đầu.

*Thonny:*

* Nhà phát triển: University of Tartu
* Loại: IDE đơn giản, phù hợp cho người mới bắt đầu
* Giới thiệu:

Thonny là một IDE nhẹ và đơn giản, phù hợp cho những người mới bắt đầu học Python. Nó cung cấp giao diện trực quan và dễ sử dụng, giúp lập trình viên dễ dàng viết mã, gỡ lỗi và thực thi mã Python.

* Tính năng nổi bật:

Giao diện đơn giản: Giao diện thân thiện và dễ sử dụng cho người mới bắt đầu.

REPL tích hợp: Cho phép thử nghiệm và kiểm tra mã trực tiếp trong môi trường phát triển.

Debugger cơ bản: Hỗ trợ gỡ lỗi, theo dõi từng bước mã.

Cài đặt dễ dàng: Cài đặt đơn giản và không yêu cầu nhiều tài nguyên hệ thống.

* Ưu điểm:

Thân thiện với người mới bắt đầu.

Giao diện dễ sử dụng, không rối rắm.

Nhẹ và hoạt động mượt mà trên các máy cấu hình thấp.

* Nhược điểm:

Thiếu tính năng nâng cao so với các IDE chuyên nghiệp.

Không hỗ trợ nhiều cho các dự án phức tạp.

*uPyCraft IDE:*

* Nhà phát triển: DFRobot
* Loại: IDE chuyên dụng cho MicroPython trên ESP32/ESP8266
* Giới thiệu:

uPyCraft IDE là một IDE được phát triển đặc biệt cho MicroPython, phù hợp cho lập trình trên các vi điều khiển như ESP32 và ESP8266. Đây là công cụ lý tưởng cho các dự án IoT sử dụng các thiết bị nhúng này, giúp lập trình viên dễ dàng tải mã nguồn lên thiết bị và giao tiếp với hệ thống.

* Tính năng nổi bật:

Hỗ trợ các vi điều khiển ESP32/ESP8266: IDE tối ưu hóa cho lập trình trên ESP32 và ESP8266, hai vi điều khiển phổ biến trong các dự án IoT.

Tích hợp REPL: Cho phép lập trình viên gửi mã trực tiếp đến vi điều khiển và nhận phản hồi tức thì.

Quản lý tệp hệ thống: Hỗ trợ tải lên, xóa và chỉnh sửa các tệp Python trực tiếp trên bộ nhớ của vi điều khiển.

Hỗ trợ firmware: Cho phép nạp firmware MicroPython vào thiết bị.

* Ưu điểm:

Giao diện đơn giản và dễ sử dụng.

Quản lý tệp và giao tiếp trực tiếp với thiết bị nhúng.

Hỗ trợ MicroPython và các cảm biến/thiết bị IoT.

* Nhược điểm:

Chỉ hỗ trợ các vi điều khiển cụ thể như ESP32 và ESP8266.

Tính năng chưa đầy đủ như các IDE chuyên nghiệp khác.

*Visual Studio Code (VS Code):*

* Nhà phát triển: Microsoft
* Loại: IDE đa năng, hỗ trợ nhiều ngôn ngữ lập trình
* Giới thiệu:

Visual Studio Code là một IDE mã nguồn mở nổi tiếng, hỗ trợ đa ngôn ngữ lập trình, trong đó có Python. Với sự hỗ trợ mạnh mẽ từ các plugin và tiện ích mở rộng, VS Code là lựa chọn phổ biến cho các lập trình viên Python, đặc biệt là khi phát triển các ứng dụng phức tạp.

* Tính năng nổi bật:

Hỗ trợ đa ngôn ngữ lập trình: Ngoài Python, VS Code còn hỗ trợ nhiều ngôn ngữ khác như JavaScript, C++, và Java.

Plugin và tiện ích mở rộng: Với kho plugin phong phú, lập trình viên có thể tùy chỉnh IDE để phục vụ nhu cầu của mình.

Gỡ lỗi và kiểm tra mã: Hỗ trợ gỡ lỗi và kiểm tra mã với tính năng debug mạnh mẽ.

Hỗ trợ Git: Tích hợp công cụ Git để quản lý mã nguồn dễ dàng.

* Ưu điểm:

Mạnh mẽ và linh hoạt, phù hợp cho các dự án lớn.

Hỗ trợ rất tốt cho các lập trình viên Python.

Tính năng mở rộng mạnh mẽ qua các plugin.

* Nhược điểm:

Yêu cầu cấu hình hệ thống tốt để hoạt động mượt mà.

Không chuyên dụng cho Python như PyCharm, nhưng với các plugin, có thể tùy chỉnh để trở thành một IDE hoàn hảo cho Python.

*Mu Editor:*

* Nhà phát triển: Adafruit Industries
* Loại: IDE đơn giản cho Python, phù hợp cho các dự án nhỏ và học tập
* Giới thiệu:

Mu là một IDE nhẹ và đơn giản dành cho Python, đặc biệt hỗ trợ cho MicroPython và các vi điều khiển như ESP8266, ESP32, và BBC micro:bit. IDE này được thiết kế dành cho người mới bắt đầu và giáo dục, giúp học sinh, sinh viên dễ dàng tiếp cận lập trình Python.

* Tính năng nổi bật:

Giao diện người dùng dễ sử dụng: Giao diện đơn giản, không có nhiều phức tạp, dễ dàng cho người mới bắt đầu.

Hỗ trợ MicroPython: Phát triển ứng dụng cho vi điều khiển sử dụng MicroPython một cách đơn giản.

Chạy mã trực tiếp: Cho phép chạy mã Python trực tiếp trên vi điều khiển.

Mã nguồn mở: Là phần mềm mã nguồn mở và miễn phí.

* Ưu điểm:

Dễ sử dụng cho người mới bắt đầu.

Giao diện sạch sẽ và dễ dàng tiếp cận.

Phù hợp cho giáo dục và các dự án nhỏ.

* Nhược điểm:

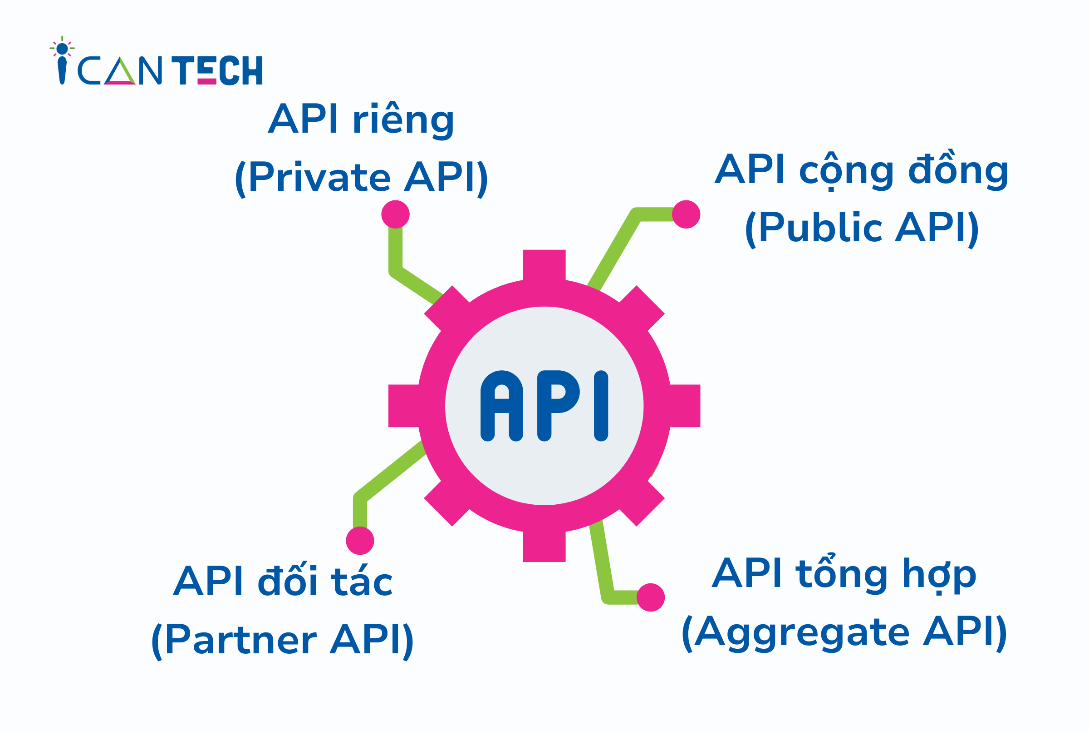
Tính năng hạn chế, không phù hợp cho các dự án lớn hoặc phức tạp.

Thiếu hỗ trợ các tính năng nâng cao so với IDE chuyên nghiệp.

2.1.2.1 Khái niệm API

API là giao thức giúp 2 hệ thống phần mềm tương tác với nhau thông qua một loạt những khái niệm.

Ví dụ: Phần mềm của tổng cục thời tiết cung cấp thông tin dự báo thời tiết mỗi ngày. Ứng dụng (App) thời tiết trên điện thoại di động của bạn sẽ lấy dữ liệu từ phần mềm của tổng cục thời tiết bằng API và cung cấp tin tức chính xác dự báo về nhiệt độ, hiện tượng thời tiết mỗi ngày tới điện thoại di động của bạn.



Hình 2. API

2.1.2.2 Các hình thức hoạt động trong API

Soap API:

Soap API (Simple Object Access Protocol API) là API hỗ trợ Khả năng truy vấn đối tượng đơn giản giữa máy chủ (server) và máy khách (client) chia sẻ thông tin tệp qua XML.

RPC API:

RPC API (Remote Procedure Call API) là API thực thi hàm lệnh từ xa. Máy khách (client) thực hiện một hàm (hoặc lệnh) trên máy chủ và máy chủ gửi phản hồi đến các máy khách (client).

REST API:

REST API (Representational State Transfer API) là dạng API thông dụng và phổ biến nhất trên web ngày nay. Máy khách (client) gửi yêu cầu đến máy chủ (server) dưới dạng dữ liệu. Máy chủ (server) dùng dữ liệu nhập bởi máy khách (client) này để tạo ra hàm mới và gửi các dữ liệu mới tới máy khách (client).

Websocket API:

Websocket API là một phiên bản của API web hiện đại hơn dùng các hàm JSON trong truyền dữ liệu. WebSocket API cung cấp sự kết nối hai chiều giữa phần mềm máy khách (client) và máy chủ (server). Máy chủ (server) sẽ gửi yêu cầu gọi lại tới những máy khách (client) được liên kết, việc này làm cho API này hữu ích và hiệu quả hơn API REST.

2.1.2.3 Lợi ích khi sử dụng API

Những thành phần, phần mềm sẽ được đồng bộ hoá dữ liệu giữa ứng dụng và máy chủ (server) khi tích hợp API.

Ví dụ nếu bạn dùng iphone, ảnh của bạn sẽ được tự động đồng bộ hóa với dữ liệu đám mây trong hoặc trong một trường hợp khác, máy tính xách tay của bạn tự động đồng bộ hoá ngày giờ khi chúng ta ở một quốc gia có múi giờ khác. Đa số các doanh nghiệp có thể sử dụng API để thực hiện nhiều tác vụ của hệ thống (server) hiệu quả hơn.

2.1.2.4 Các loại API

API riêng (Private API):

Đây là các API nội bộ của một doanh nghiệp. Chúng được sử dụng nhằm kết nối toàn bộ hệ thống cũng như mọi dữ liệu trong doanh nghiệp đó đang có.

API cộng đồng (Public API):

Những API này dành cho cộng đồng và không được sử dụng riêng bởi bất cứ ai, bất kì cá nhân hay tổ chức. Những loại API này không yêu cầu bất kỳ giấy phép và người dùng không cần trả phí khi sử dụng.

API đối tác (Partner API):

Những API này sẽ dành cho những nhà phát triển hoặc những nhà điều hành bên ngoài được uỷ quyền nhằm phát triển các mối quan hệ đối tác giữa các doanh nghiệp với doanh nghiệp.

API tổng hợp (Aggregate API):

Những API có thể kết hợp giữa hai API khác nhau trở lên nhằm đáp ứng các nhu cầu hoặc các hành động phức tạp của hệ thống.

2.1.2.5 Điểm cuối của API

Điểm cuối của API là điểm tiếp xúc cuối cùng bên trong hệ thống giao tiếp của API. Những điểm cuối này bao gồm các URL máy chủ (server), dịch vụ (service) và các địa chỉ số quan trọng khác, tại đây thông tin được truyền đi và nhận thông qua các hệ thống. Điểm cuối API đặc biệt quan trọng với doanh nghiệp bởi 2 lý do chính:

Bảo mật (Security):

Điểm cuối API rất dễ làm cho hệ thống bị hack. Việc giám sát và đảm bảo độ bảo mật cho API nhằm ngăn chặn việc hệ thống bị hack là vô cùng quan trọng.

Hiệu năng (Performance):

Điểm cuối của API có lưu lượng truy cập cao, có thể gây ra hiện tượng tắc nghẽn cục bộ và tác động lên hiệu suất hệ thống.

2.1.2.6 Các bước tạo API

Dưới đây là 5 bước cần làm để tạo ra API:

Lập kế hoạch về API

Lập kế hoạch về thông số chi tiết kỹ thuật của API, ví dụ như sử dụng OpenAPI để cung cấp bản phác thảo về API của bạn. Bạn nên dự tính trước những trường hợp sử dụng khác nhau và chắc chắn rằng API đáp ứng những yêu cầu thiết kế API hiện tại.

Lập tài liệu cho API

Mặc dù không cần giải thích nhiều về API, tài liệu về API có vai trò là chỉ dẫn nhằm nâng cao tính khả dụng. Các API được xây dựng tài liệu hoàn chỉnh, với những chức năng và trường hợp để dễ dàng hơn trong thiết kế hướng đến khách hàng.

Xây dựng API

Lập trình viên có thể sử dụng các API nguyên mẫu để tùy biến thành API có thông số kỹ thuật theo yêu cầu của dự án.

Kiểm thử API

Kiểm thử API tương tự với kiểm thử phần mềm để nhằm phát hiện lỗ hổng và khiếm khuyết.

Các chiến lược kiểm thử API tương tự với các chiến lược kiểm thử ứng dụng khác. Trọng tâm chủ yếu là xác minh phản hồi của server. Các bước kiểm thử API bao gồm:

Đưa ra nhiều tiêu chí về điểm cuối API trong kiểm thử hiệu năng.

Viết các test case chức năng nhằm đánh giá logic hoạt động và tính chính xác của chức năng.

Kiểm thử lỗ hổng bằng việc mô phỏng những vụ tấn công hệ thống.

Phát hành API ra thị trường

Tương tự với sàn mua sắm thương mại điện tử Amazon, nhà phát triển có thể mua bán các API khác trên thị trường mua bán API.

2.1.2.7 Tại sao bạn nên quan tâm đến API

APIs quan trọng vì nhiều lý do, và đây là lý do bạn nên quan tâm đến chúng:

* *Tích hợp với các hệ thống khác:* APIs cho phép các chương trình phần mềm khác nhau giao tiếp, giúp việc tích hợp và kết nối các hệ thống khác nhau trong một tổ chức trở nên khả thi. Điều này có thể dẫn đến các hoạt động hiệu quả hơn và quản lý dữ liệu tốt hơn.
* *Độ linh hoạt và kiểm soát*: Với APIs, bạn có thể tùy chỉnh trải nghiệm người dùng bằng cách tạo ra các giao diện tùy chỉnh hoặc tích hợp các tính năng từ các dịch vụ khác vào ứng dụng của bạn.
* *Đổi mới và phát triển:* APIs cho phép các nhà phát triển xây dựng dựa trên các nền tảng hiện có, thúc đẩy đổi mới và cho phép các doanh nghiệp tận dụng các chức năng mới mà không cần phải phát minh lại bánh xe.
* *Sự tương tác với khách hàng và nhà cung cấp:* Bằng cách công khai một số APIs, bạn có thể cho phép khách hàng hoặc nhà cung cấp tương tác trực tiếp với các hệ thống của bạn, điều này có thể đơn giản hóa quy trình và cải thiện việc cung cấp dịch vụ.
* *Đưa ra doanh thu:* APIs có thể được sử dụng để tạo ra các nguồn doanh thu mới bằng cách cho phép các bên thứ ba truy cập vào dịch vụ hoặc dữ liệu của bạn một cách có kiểm soát.

2.1.2.8 Các phản hồi API bằng Python

Mã trạng thái: Phản hồi từ một yêu cầu API bao gồm một mã trạng thái chỉ ra kết quả của yêu cầu.

Các mã trạng thái phổ biến là:

* 200 OK: Yêu cầu đã thành công.
* 400 Bad Request: Yêu cầu không được hiểu bởi máy chủ.
* 401 Unauthorized: Cần xác thực và đã không thành công hoặc không được cung cấp.
* 403 Forbidden: Yêu cầu là hợp lệ, nhưng máy chủ từ chối hành động.
* 404 Not Found: Tài nguyên yêu cầu không thể được tìm thấy.
* 500 Internal Server Error: Một điều kiện bất ngờ đã xảy ra bởi máy chủ.

Nội dung phản hồi: Nội dung của phản hồi thường chứa dữ liệu bạn đã yêu cầu, thường ở định dạng JSON. Bạn có thể phân tích dữ liệu này bằng cách sử dụng thư viện json của Python.

Tiêu đề: Các tiêu đề phản hồi có thể cung cấp thông tin hữu ích như loại nội dung, máy chủ, ngày tháng, và nhiều hơn nữa.

2.1.3 Telegram

2.1.3.1 Giới thiệu về Teleram

Telegram là một ứng dụng trực tuyến cho phép người dùng nhắn tin, gọi điện, video call miễn phí và chia sẻ các dữ liệu đa nền tảng thông qua kết nối mạng internet.

Ứng dụng này sở hữu hệ thống server mạnh mẽ với quy mô rộng lớn trên phạm vi toàn cầu, vậy nên sẽ đảm bảo duy trình dịch vụ nhanh chóng, ổn định. Telegram hiện được hỗ trợ trên hầu hết các nền tảng như Linux, Windows, MacOS, iOS và Android. Qua đó, bạn có thể dễ dàng tải ứng dụng về và kết nối với người khác trên nền tảng này.

Với những ưu điểm vượt trội cùng hàng loạt tính năng nổi bật, Telegram đã và đang trở thành một trong những nền tảng sở hữu số lượng người dùng đông đảo hàng đầu hiện nay. Chỉ tính riêng đến tháng 1/2021, đã có hơn 500 triệu người dùng hoạt động thường xuyên trên Telegram và con số này cũng đưa ứng dụng lên vị trí top đầu trong bảng xếp hạng app có lượt tải về nhiều nhất trên thế giới của tháng.



Hình 2. Telegram

2.1.3.2 Nguồn gốc và lịch sử phát triển của Telegram

Telegram chính thức có mặt trên các nền tảng lần đầu tiên vào tháng 8/2013 bởi Pavel Durov. Ngay sau khi ra mắt, ứng dụng này đã nhanh chóng đạt được nhiều thành tựu nổi bật và trở thành một trong những nền tảng trò chuyện được sử dụng nhiều nhất toàn cầu.

Pavel Durov – nhà sáng lập của Telegram đã tạo nên ứng dụng này với mong muốn phát triển một nền tảng trò chuyện an toàn và bảo mật tối đa. Được biết, doanh nhân người Nga này đã “rót vốn” hàng triệu đô la từ tiền của chính mình để đầu tư cho Telegram. Mục tiêu của ông là không tập trung vào quảng cáo rầm rộ hay thu thập thông tin, dữ liệu của người dùng, mà sẽ hướng đến nguyên tắc cốt lõi là “tôn trọng quyền tự do và sự riêng tư trong ngôn luận”.

Từ ý tưởng ban đầu đó, Telegram đã hình thành và trở thành nền tảng kết nối giữa hàng trăm triệu người dùng đến từ mọi nơi trên thế giới. Tuy nhiên, theo thời gian, nhiều kẻ gian đã bắt đầu nhắm tới và lợi dụng nền tảng này để thực hiện các hành vi lừa đảo. Điều này gây ra hệ lụy đáng kể, thậm chí khiến người mất hàng chục, hàng trăm triệu đồng từ đó.

Tuy nhiên, về bản chất thì Telegram không xấu, chỉ người lợi dụng nó để trục lợi bất chính mới xấu. Vậy nên, bạn cần nắm rõ các tính năng nổi bật của Telegram và những hành vi lừa đảo thường gặp trên Telegram là gì để tránh trở thành nạn nhân của chúng. Nội dung những phần tiếp theo sẽ đề cập rõ hơn về điều này!

2.1.3.3 Các tính năng nổi trội của Telegram

*Cho phép chỉnh sửa nội dung tin nhắn đã gửi*

Đây là một tính năng khá thú vị trên Telegram tiếng Việt. Nếu bạn vô tình gửi nhầm tin nhắn cho đối phương, bạn có thể “chữa cháy” bằng cách nhấn giữ dòng tin nhắn đó, sau đó chọn Edit và tiến hành chỉnh sửa nội dung bên trong. Mặc dù người dùng sẽ nhận được thông báo về hành vi thay đổi, chỉnh sửa của bạn, nhưng họ không thể biết được tin nhắn cũ mà bạn gửi trên Telegram là gì.

*Trò chuyện bí mật*

Đôi khi, bạn sẽ muốn tạo một cuộc trò chuyện bí mật để không ai có thể đọc trộm được nội dung tin nhắn, chẳng hạn như đoạn hội thoại bàn bạc về chiến lược bí mật hoặc chia sẻ thông tin nhạy cảm của cá nhân, doanh nghiệp… Khi đó, bạn có thể sử dụng tính năng Secret Chat trên Telegram để mã hóa đầu cuối, ngăn không ghi lại dữ liệu trên máy chủ và tự động xóa đi nội dung sau một khoảng thời gian được thiết lập từ trước.

*Chia sẻ file với dung lượng lớn*

Sau khi đã nắm rõ Telegram là gì, hẳn bạn sẽ hình dung được những tính năng cơ bản của ứng dụng này, bao gồm nhắn tin, gọi điện, chia sẻ file phương tiện… Tuy nhiên, ứng dụng này sẽ khiến bạn ngạc nhiên hơn khi cho phép người dùng gửi các file đa phương tiện với kích thước lên đến 2GB – một điểm hiếm thấy trên các nền tảng mạng xã hội khác.

*Kiểm soát các file tải xuống*

Người dùng hoàn toàn có thể kiểm soát tính năng tải về tự động hình ảnh, video được gửi đến trong đoạn chat bằng cách: Đầu tiên, bạn vào mục Settings để bắt đầu cài đặt, sau đó tại Message hãy chọn Save to gallery để hoàn tất quá trình này.

*Chatbot Telegram*

Hãy thử khám phá tính năng Chatbot – một tính năng vô cùng thú vị trên Telegram. Hiện nay, Telegram cung cấp rất nhiều chatbot, bạn có thể tương tác với chúng và khám phá hàng loạt nội dung hữu ích, bao gồm tìm kiếm thông tin, kết nối, nhắc nhở lịch, dạy học, chơi game…

*Sử dụng dịch vụ đám mây*

Thông qua dịch vụ đám mây, tất cả dữ liệu bao gồm tin nhắn, video, hình ảnh, file tài liệu… đều được lưu trữ trên nền tảng này và người dùng có thể truy cập vào mọi lúc, mọi nơi từ mọi thiết bị một cách nhanh chóng, dễ dàng.

*Đa nền tảng*

Hiện nay, Telegram đã có mặt trên hầu hết các nền tảng, từ điện thoại, máy tính bảng, máy tính… cho đến các hệ điều hành phổ biến như MacOs, iOS, Android, Windows, Linux,…

2.1.4 Chatbot Telegram

Telegram Bot là một công cụ mạnh mẽ trong ứng dụng Telegram, cho phép tự động hóa các tác vụ và tương tác với người dùng thông qua các cuộc trò chuyện. Chatbot Telegram là một bot (chương trình tự động) được thiết kế để trả lời câu hỏi, thực hiện các tác vụ cụ thể hoặc cung cấp dịch vụ hỗ trợ khách hàng trong các cuộc trò chuyện trên Telegram. Chatbot này có thể giao tiếp với người dùng như một người trợ lý ảo, tự động hóa các tác vụ và mang đến trải nghiệm tương tác tự động, hiệu quả.

2.1.4.1 Các tính năng của Chatbot Telegram:

* Tự động trả lời câu hỏi:

Chatbot Telegram có thể tự động trả lời các câu hỏi thường gặp (FAQs). Ví dụ, bot có thể cung cấp thông tin về sản phẩm, dịch vụ, tình trạng đơn hàng, hoặc các câu hỏi kỹ thuật đơn giản mà người dùng thường xuyên hỏi.

* Gửi thông báo và cập nhật:

Chatbot có thể gửi thông báo tự động tới người dùng về các sự kiện, chương trình khuyến mãi, tin tức, hoặc các thông báo quan trọng. Điều này đặc biệt hữu ích trong việc duy trì sự liên lạc với khách hàng mà không cần sự can thiệp thủ công.

* Quản lý và phân tích dữ liệu:

Chatbot Telegram có thể lưu trữ và xử lý dữ liệu người dùng như tên, email, yêu cầu, lịch sử trò chuyện để phân tích hành vi người dùng, tạo báo cáo hoặc tùy chỉnh trải nghiệm của người dùng. Một số bot còn hỗ trợ tạo các cuộc thăm dò hoặc khảo sát để thu thập thông tin từ người dùng.

* Tích hợp với các dịch vụ bên ngoài:

Một trong những tính năng mạnh mẽ của Telegram bot là khả năng tích hợp với các API của bên thứ ba, như Google Sheets, hệ thống thanh toán, hoặc các nền tảng khác. Ví dụ, bạn có thể tạo một chatbot bán hàng và tích hợp với dịch vụ thanh toán để người dùng có thể mua sản phẩm trực tiếp thông qua Telegram.

* Hỗ trợ nhiều loại hình tệp tin:

Chatbot Telegram có thể gửi và nhận các loại tệp tin khác nhau, bao gồm văn bản, hình ảnh, video, âm thanh, và tệp PDF. Điều này làm cho chatbot trở nên linh hoạt và tiện lợi trong việc chia sẻ tài liệu và thông tin với người dùng.

* Giao diện người dùng trực quan:

Telegram bots có thể sử dụng custom keyboards (bàn phím tùy chỉnh) và inline buttons (nút inline) để cung cấp giao diện người dùng trực quan và dễ sử dụng. Người dùng có thể chọn các tùy chọn trực tiếp mà không cần phải nhập văn bản.

2.1.4.2 Các ứng dụng của Chatbot Telegram:

* Hỗ trợ khách hàng tự động:

Chatbot Telegram có thể cung cấp dịch vụ hỗ trợ khách hàng 24/7 mà không cần sự can thiệp thủ công. Các câu hỏi thường gặp có thể được trả lời tự động, và bot có thể phân loại các yêu cầu của khách hàng để chuyển cho người hỗ trợ khi cần thiết.

* Quản lý đơn hàng và dịch vụ bán hàng:

Chatbot có thể giúp các doanh nghiệp theo dõi và quản lý đơn hàng, cung cấp thông tin về sản phẩm, xử lý yêu cầu thanh toán và gửi thông báo về tình trạng đơn hàng cho khách hàng.

* Cung cấp thông tin và giáo dục:

Bots có thể được sử dụng để cung cấp các khóa học, bài học trực tuyến, thông tin giáo dục, tài liệu học tập hoặc tổ chức các cuộc thi trực tuyến. Ví dụ, chatbot có thể cung cấp các bài kiểm tra kiến thức, giảng dạy ngôn ngữ, hoặc cung cấp thông tin về các khóa học.

* Tạo cuộc thăm dò và khảo sát:

Telegram bot có thể dễ dàng tạo ra các cuộc thăm dò ý kiến, khảo sát và thu thập phản hồi từ người dùng, giúp các công ty hoặc tổ chức hiểu hơn về khách hàng và cải thiện sản phẩm, dịch vụ của mình.

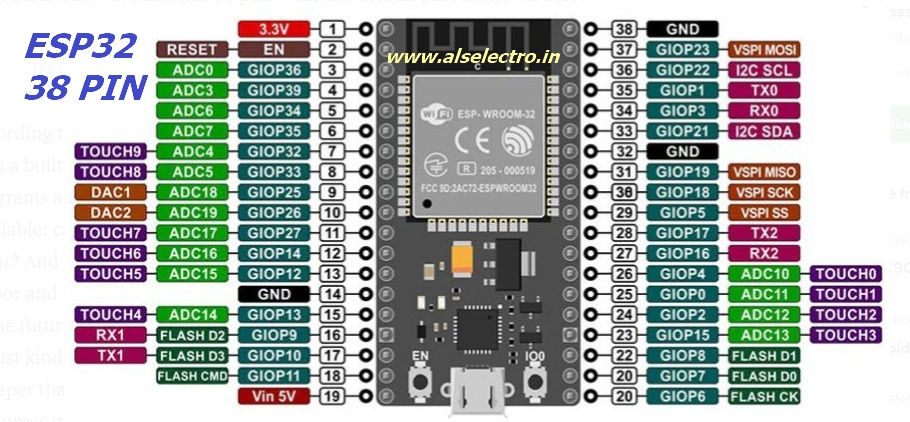
* Tích hợp với các dịch vụ và nền tảng khác.

Telegram Bot có thể tích hợp với các dịch vụ bên ngoài như Google Sheets, CRM, hệ thống thanh toán,....

2.1 Linh kiện

2.2.1 ESP32

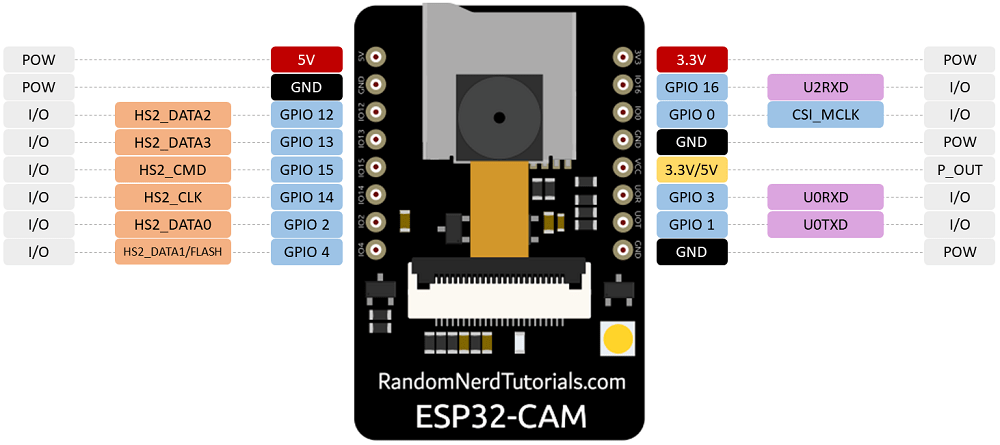
**ESP32** là một dòng vi điều khiển (microcontroller) mạnh mẽ, tích hợp Wi-Fi và Bluetooth (cả Bluetooth Classic và Bluetooth Low Energy - BLE), được phát triển bởi **Espressif Systems**. ESP32 được sử dụng rộng rãi trong các dự án IoT (Internet of Things) nhờ vào khả năng kết nối mạng mạnh mẽ, tính năng linh hoạt và chi phí thấp.



Hình 2. ESP 32

* **CPU: Dual-core Xtensa® LX6, 240 MHz, 600 DMIPS**
* **RAM: 520 KB SRAM**
* **Flash: Lên đến 4 MB**
* **Nguồn: 3.0V – 3.6V**
* **Wi-Fi: IEEE 802.11 b/g/n, 2.4 GHz**
* **Bluetooth: Dual-mode (4.2 BR/EDR, BLE)**
* **GPIO: Lên đến 36 chân**
* **ADC/DAC: ADC 12-bit (18 kênh), DAC 8-bit (2 kênh)**
* **Giao tiếp: UART, SPI, I2C, I2S, CAN (tùy model)**
* **Tiết kiệm năng lượng: Deep Sleep < 10 µA**

1. **ESP32-CAM**



Hình 2. ESP32-CAM

ESP32-CAM có một camera kích thước nhỏ, rất cạnh tranh trong ngành, giống như mô-đun chính, mô-đun này có thể được xử lý công việc độc lập, module có kích thước nhỏ gon chỉ 40 x 27 x12 mm, dòng nghỉ chỉ 6mA.

ESP-32CAM có thể được sử dụng rộng rãi trong các ứng dụng IoT khác nhau, thích hợp cho thiết bị thông minh gia đình, điều khiển không dây công nghiệp, giám sát không dây kiểm soát, nhận dạng không dây QR, tín hiệu hệ thống định vị không dây…Nó là một giải pháp lý tưởng cho các ứng dụng IoT.

**Thông Số kỉ thuật:**

* Vi điều khiển: ESP32 (Dual-core, 240 MHz)
* Bộ nhớ: 520KB SRAM, 4MB Flash
* Kết nối: Wi-Fi 802.11 b/g/n (2.4 GHz), Bluetooth BLE v4.2
* Camera: OV2640, độ phân giải tối đa 1600x1200 (UXGA), nén JPEG
* GPIO: 9 chân, hỗ trợ ADC, PWM, UART, SPI, I2C
* Nguồn: 5V (hoặc 3.3V tùy phiên bản)

2.2.2 Module Relay



Hình 2. Relay

Module Relay là một thành phần điện tử được sử dụng để điều khiển các thiết bị điện tử hoặc tải điện áp cao (như bóng đèn, động cơ, quạt, v.v.) từ một mạch điện tử có điện áp thấp (thường là các vi điều khiển như Arduino, ESP32, Raspberry Pi). Relay hoạt động như một công tắc điện từ, cho phép bật/tắt các thiết bị bên ngoài mà không cần kết nối trực tiếp giữa mạch điện tử và thiết bị đó.

**Thông số kỉ thuật:**

Điện áp: 3.6V - 6V DC.

Giao tiếp: UART (Serial).

Dung lượng bộ nhớ: Lưu trữ đến 1000 mẫu vân tay.

Độ phân giải: 500 DPI.

Tốc độ quét: Nhận diện vân tay nhanh chóng trong 1-2 giây.

2.2.3 Khóa cửa chốt điện tử 12v

**Khóa chốt điện tử** có chức năng hoạt động như một ổ khóa cửa sử dụng Solenoid để kích đóng mở bằng điện, được sử dụng nhiều trong nhà thông minh hoặc các loại tủ, cửa phòng, cửa kho , Khóa chốt điện từ này sử dụng điện áp 12VDC, là loại thường đóng (cửa đóng) với chất lượng tốt, độ bền cao. Khóa có thể sử dụng chung với các mạch chức năng tạo thành một hệ thống thông minh.



Hình 2. Khóa cửa chốt điện từ

**Thông số kỹ thuật:**

* Điện áp điều khiển: 5V DC.
* Công suất tải: Điều khiển thiết bị DC 10A, AC 220V với dòng tối đa 10A.
* Tín hiệu điều khiển: Tín hiệu 5V (HIGH để đóng, LOW để mở).
* Ưu điểm:
* Điều khiển thiết bị cao áp từ mạch vi xử lý.
* An toàn, dễ sử dụng và ứng dụng rộng rãi.
* Nhược điểm:
* Chỉ chịu tải vừa phải.
* Tiếng ồn khi hoạt động.
* Nguy cơ điện khi làm việc với thiết bị AC.

2.2.4 Cảm Biến

2.2.4.1 Cảm biến ánh sáng

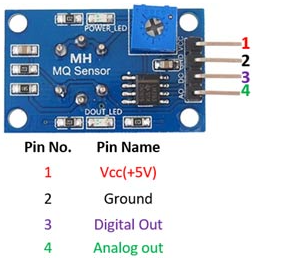
Cảm biến ánh sáng là thiết bị quang điện có khả năng chuyển đổi ánh sáng thành tín hiệu điện. Nguồn sánh sáng mà thiết bị chuyển đổi có thể là ánh sáng nhìn thấy hoặc ánh sáng hồng ngoại.



Hình 2. Cảm biến ánh sáng

2.2.4.2 Cảm biến khí gas MQ-2

Các cảm biến MQ-2 có thể phát hiện và đo khí LPG, Rượu, Propane, Hydrogen, CO, và thậm chí là metan.



Hình 2. MQ-2

* Điện áp hoạt động + 5V.
* Sử dụng để đo hoặc phát hiện khí LPG, Alcohol, Propane, Hydrogen, CO và thậm chí cả methane.
* Điện áp đầu ra analog: từ 0V đến 5V.
* Điện áp đầu ra digital: 0V hoặc 5V (chuẩn logic TTL).
* Thời gian làm nóng trước khi sử dụng 20 giây.
* Có thể sử dụng như một cảm biến digital hoặc analog.
* Độ nhạy của chân DO (digital output) có thể được thay đổi bằng cách chiết áp.

2.2.4.3 Cảm biến DHT-11

DHT-11 là một cảm biến giá rẻ được sử dụng rộng rãi để đo nhiệt độ và độ ẩm trong các dự án điện tử. Cảm biến này cung cấp dữ liệu ở dạng kỹ thuật số, giúp dễ dàng giao tiếp với các vi điều khiển như Arduino, ESP32, Raspberry Pi. DHT11 nổi bật nhờ sự đơn giản trong sử dụng, chi phí thấp và kích thước nhỏ gọn.



Hình 2. DHT-11

**Dải đo nhiệt độ:** 0°C đến 50°C .

* **Dải đo độ ẩm**: 20% đến 90% RH.
* **Nguồn cấp**: 3.3V - 5V DC.
* **Tín hiệu đầu ra**: Kỹ thuật số.
* **Kích thước**: Nhỏ gọn, thường chỉ khoảng 15mm x 12mm x 5mm.
* **Thời gian phản hồi**: 1 giây/lần đọc dữ liệu.
* **Đầu ra dữ liệu**: 1 dây.

2.2.5 Màn hình hiển thị

2.2.5.1 Màn hình LCD 16x2



Hình 2. Màn hình LCD 16x2 I2c

LCD 16×2 được sử dụng để hiển thị trạng thái hoặc các thông số.

* LCD 16×2 có 16 chân trong đó 8 chân dữ liệu (D0 – D7) và 3 chân điều khiển (RS, RW, EN).
* 5 chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16×2.
* Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.
* Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

2.2.5.2 Màn hình OLED

Giao tiếp I2C cho khả năng hiển thị đẹp, sang trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO.



Hình 2. OLED 0.96 I2C

**Thông số kỉ thuật:**

* Điện áp sử dụng: 2.2~5.5VDC.
* Công suất tiêu thụ: 0.04w.
* Góc hiển thị: lớn hơn 160 độ.
* Số điểm hiển thị: 128×64 điểm.
* Độ rộng màn hình: 0.96 inch.
* Màu hiển thị: Trắng.
* Giao tiếp: I2C.
* Driver: SSD1306.

2.2.6 Buzzer

Buzzer là một thiết bị tạo ra tiếng còi hoặc tiếng bíp. Có nhiều loại nhưng cơ bản nhất là buzzer áp điện, là một miếng phẳng của vật liệu áp điện với hai điện cực. Loại buzzer này đòi hỏi phải có các bộ dao động (hoặc vi điều khiển) để điều khiển nó.

****

Hình 2. BUZZER

2.2.7 Đèn

LED (Light Emitting Diode) là một loại diode phát sáng, được sử dụng rộng rãi trong các ứng dụng chiếu sáng và hiển thị.

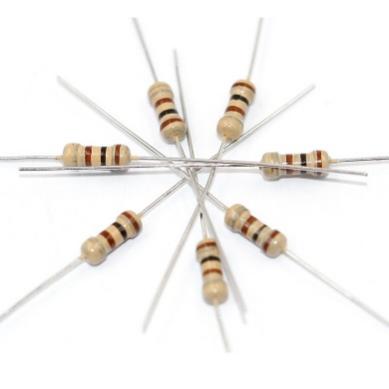


Hình 2. Đèn

2.2.8 Điện trở

Điện trở là linh kiện điện tử có 2 tiếp điểm kết nối. Chức năng của điện trở là điều chỉnh mức độ tín hiệu, hạn chế cường độ dòng điện chạy trong mạch. Bên cạnh đó, điện trở còn có thể chia điện áp, kích hoạt các linh kiện điện tử chủ động như tiếp điểm cuối trong đường truyền điện, transistor và một số ứng dụng khác.

Điện trở công suất có khả năng tiêu tán lượng lớn điện năng chuyển sang nhiệt năng trong các hệ thống phân phối điện, bộ điều khiển động cơ.



Hình 2. Điện trở

2.2.9 Quạt 5v

Trong các ứng dụng IoT, quạt 5V thường được sử dụng để hỗ trợ làm mát và đảm bảo hiệu suất ổn định cho các thiết bị điện tử. Các hệ thống IoT, đặc biệt là những hệ thống hoạt động liên tục hoặc xử lý dữ liệu phức tạp, có thể sinh nhiệt trong quá trình vận hành. Nếu nhiệt độ tăng cao, các linh kiện như vi điều khiển, module cảm biến, hoặc các mạch nguồn có thể bị ảnh hưởng, dẫn đến giảm tuổi thọ hoặc hiệu suất. Quạt 5V với kích thước nhỏ gọn, tiêu thụ điện năng thấp, là lựa chọn lý tưởng để duy trì nhiệt độ ổn định trong các hộp điều khiển hoặc thiết bị IoT có không gian hạn chế. Ngoài ra, quạt 5V còn góp phần giảm thiểu nguy cơ hỏng hóc do nhiệt độ, giúp hệ thống hoạt động lâu dài và đáng tin cậy hơn.



Hình 2. Quạt 5v

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ SẢN PHẨM

3.1 Phân tích hệ thống

3.1.1 Mục đích của hệ thống

Mục đích chính là **tăng cường tiện nghi, an toàn, và hiệu quả năng lượng** cho người dùng thông qua việc sử dụng công nghệ để tự động hóa và điều khiển các thiết bị trong nhà. Hệ thống này giúp cải thiện chất lượng sống bằng cách làm cho ngôi nhà trở nên thông minh, dễ dàng kiểm soát và tối ưu hóa các hoạt động hàng ngày.

3.1.2 Yêu cầu của hệ thống

Yêu cầu của hệ thống nhà thông minh:

- Nhận Diện Mở Cửa: Sử dụng ESP32-CAM để nhận diện khuôn mặt và sử dụng khóa điện tử để mở cửa.

- Phát Hiện Khí Gas: Đảm bảo khả năng phát hiện và giám sát khí gas, đặc biệt là loại khí nguy hiểm như khí cháy, propan, và metan.

- Nhiệt Độ Môi Trường: Tích hợp đầu dò nhiệt độ để giám sát và điều chỉnh hoạt động của cảm biến dựa trên biến động nhiệt độ môi trường.

- Automatic: Dựa trên cảm biến ánh sáng và DHT-11 để tự động bật thiết bị điện được lập trình sẵn.

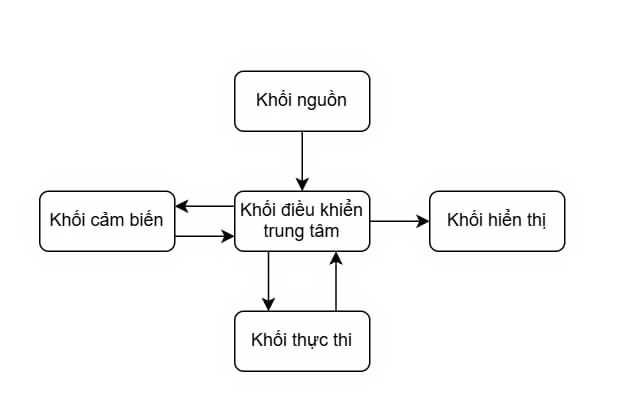
- Hiển thị thông tin: Đảm bảo hiển thị các thông tin lên màn hình.

- WebServer: Đảm bảo mọi thông tin, chức năng đều hiển thị và điều khiển trên Server.

3.2 Thiết kế và hoàn thiện sản phẩm

3.2.1 Thiết kế

3.2.1.1 Sơ đồ khối ESP 32



Hình 3. Sơ đồ khối

### ****Khối nguồn****

Cung cấp điện cho toàn bộ hệ thống, đảm bảo hoạt động của các thành phần trong nhà thông minh và nó được cấp trực tiếp từ nguồn của ESP.

**Khối xử lý trung tâm (ESP32)**

Đây là trung tâm xử lý dữ liệu của hệ thống như:

Nhận dữ liệu từ cảm biến.

Gửi dữ liệu đến WebServer.

Hiển thị thông tin lên màn hình.

Điều khiển khối chấp hành.

**Khối cảm biến**

Cung cấp thông tin môi trường xung quanh cho hệ thống như: nhiệt độ, độ ẩm, nồng độ khí gas, ánh sáng.

**Khối chấp hành**

Thực hiện các hành động dựa trên lệnh của khối xử lý trung tâm như phát tín hiệu âm thanh và mở/khóa cửa.

**WebServer**

Cung cấp giao diện giám sát và điều khiển từ xa.

**Màn hình**

Hiển thị thông tin trực tiếp cho người dùng tại nhà.

**Quy trình hoạt động tổng quát của hệ thống**:

**Thu thập dữ liệu**:

Các cảm biến (DHT11, MQ2, cảm biến ánh sáng) đo các thông số môi trường và gửi dữ liệu đến khối xử lý trung tâm.

**Xử lý dữ liệu**:

Khối xử lý trung tâm nhận và phân tích dữ liệu từ cảm biến.

Kích hoạt các hành động phù hợp (ví dụ, kích hoạt buzzer nếu phát hiện khí gas hoặc điều chỉnh trạng thái khóa cửa).

**Hiển thị và điều khiển**:

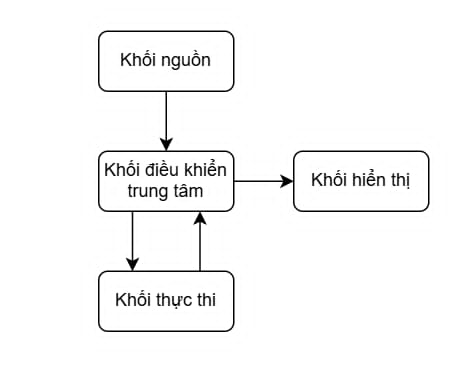
Dữ liệu được hiển thị trên màn hình để người dùng theo dõi.

Gửi dữ liệu lên WebServer để giám sát và sử dụng cùng với Telegram để điều khiển từ xa.

**Thực thi lệnh**:

Khối xử lý trung tâm gửi lệnh đến các thiết bị chấp hành (buzzer,led, quạt) để thực hiện các hành động cần thiết.

3.2.1.2 Sơ đồ khối ESP 32-CAM



**Khối Khởi nguồn:**

Đây là điểm bắt đầu của toàn bộ hệ thống.

Khối này có thể đại diện cho việc cấp nguồn điện, tín hiệu khởi động hoặc bất kỳ sự kiện nào đánh dấu sự bắt đầu hoạt động của hệ thống.

**Khối Khởi động điều khiển trung tâm:**

Sau khi nhận được tín hiệu khởi động, khối này sẽ được kích hoạt.

Đây là bộ não của hệ thống, chịu trách nhiệm điều khiển và phối hợp hoạt động của các khối khác.

Khối này sẽ tiến hành các hoạt động khởi tạo ban đầu như:

Đọc các thông số cấu hình

Khởi động các module phần cứng

Thiết lập các kết nối cần thiết

**Khối Khởi động thực thi:**

Sau khi được điều khiển trung tâm kích hoạt, khối này sẽ thực hiện các tác vụ chính của hệ thống.

Tùy thuộc vào từng hệ thống cụ thể, khối này có thể thực hiện các nhiệm vụ như:

Xử lý dữ liệu và đưa ra quyết định

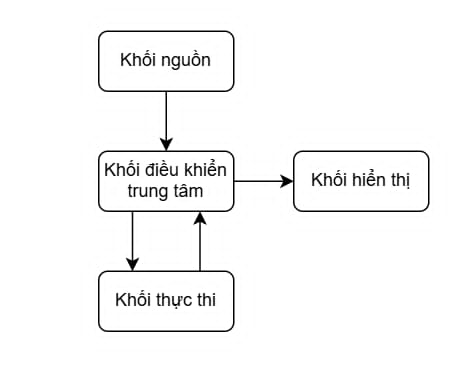
Điều khiển các thiết bị thực thi

**Khối Khởi động hiển thị:**

Khối này chịu trách nhiệm hiển thị thông tin về trạng thái hoạt động của hệ thống.

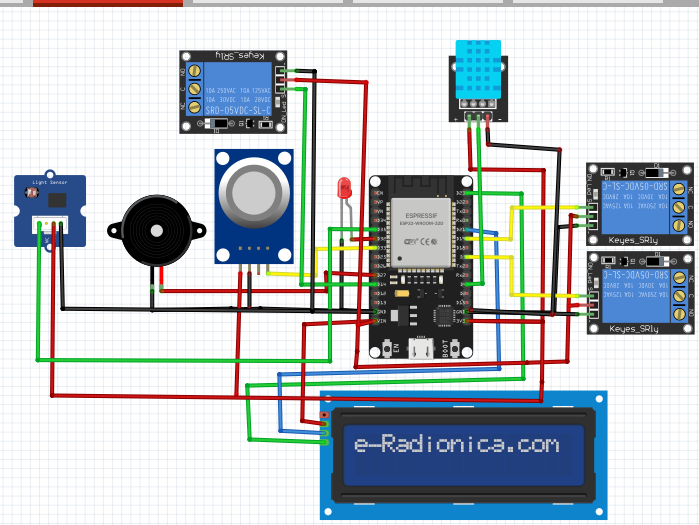
Thông tin hiển thị có thể bao gồm các thông số đo đạc, các thông báo lỗi, hoặc các thông tin điều khiển.

Các thiết bị hiển thị có thể là màn hình LCD, màn hình LED.

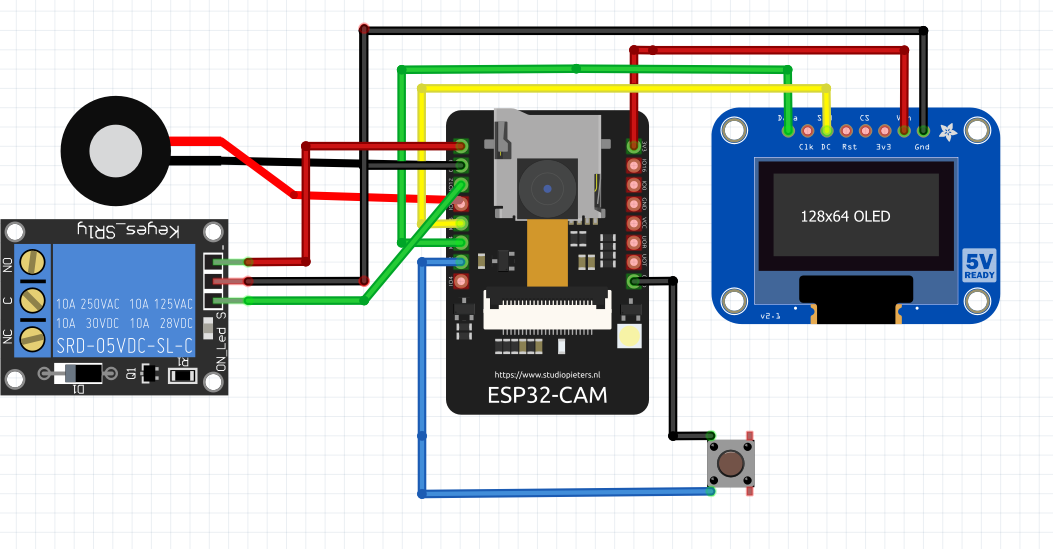


Hình 3. Sơ đồ khối esp32

3.2.1.2 Sơ đồ nối dây sản phẩm

****

Hình 3. Sơ đồ nối dây esp32

****

Hình 3. Sơ đồ nối dây esp32 cam

Bảng nối chân ESP 32-WROOM-32

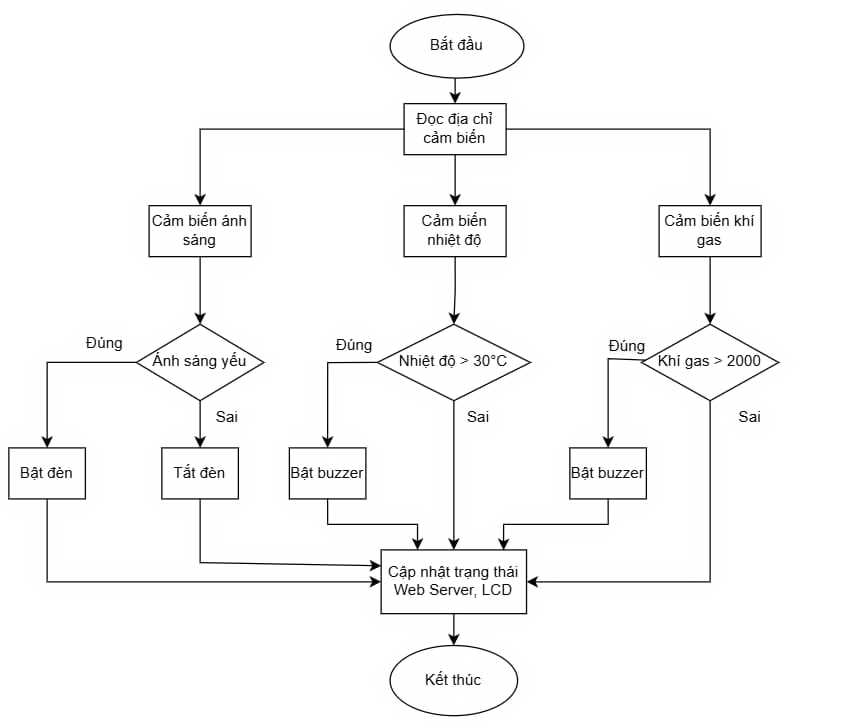
|  |  |  |  |
| --- | --- | --- | --- |
| **Thiết bị** | **Chân ESP32** | **Chân Thiết bị** | **Mô tả** |
| **Relay 1** | D14 | |  | | --- | | IN(1) |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Điều khiển relay 1 | |
| **Relay 2** | D5 | IN(2) | Điều khiển relay 2 |
| **Relay 3** | D22 | IN(3) | Điều khiển relay 3 |
| **Relay nguồn (+)** | |  | | --- | | 5V |  |  | | --- | |  | | |  | | --- | | DC+ |  |  | | --- | |  | | |  | | --- | | Nguồn cho relay |  |  | | --- | |  | |
| **Relay nguồn (-)** | GND | |  | | --- | | DC- |  |  | | --- | |  | | |  | | --- | | GND cho relay |  |  | | --- | |  | |
| **LCD (I2C)** | D23 | |  | | --- | | SCL |  |  | | --- | |  | | Giao tiếp I2C (SCL) |
|  | D21 | SDA | Giao tiếp I2C(SDA) |
|  | GND | GND | Nối đất |
|  | VIN | VCC | Nguồn 5V từ ESP32 |
| |  | | --- | | **Cảm biến ánh sáng** |  |  | | --- | |  | | D27 | A0 | Đọc giá trị cảm biến |
|  | GND | GND | Nối đất |
|  | 3V3 | VCC | Nguồn 3.3V từ ESP32 |
| |  | | --- | | **MQ-2 (Gas)** |  |  | | --- | |  | | D33 | A0 | Đọc giá trị cảm biến |
|  | GND | GND | Nối đất |
|  | 3V3 | VCC | Nguồn 3.3V từ ESP32 |
| |  | | --- | | **DHT11 (Nhiệt độ)** |  |  | | --- | |  | | D4 | DATA | Đọc dữ liệu từ cảm biến |
|  | GND | GND | Nối đất |
|  | 3V3 | VCC | Nguồn 3.3V từ ESP32 |
| **Buzzer** | D27 | |  | | --- | |  |  |  | | --- | | (+) | | Điều khiển buzzer |
|  | GND | (-) | Nối đất |
| |  | | --- | | **Đèn LED** |  |  | | --- | |  | | D32 | (+ qua điện trở) | Điều khiển LED |
|  | GND | (-) | |  | | --- | | Nối đất | |

Hình 3. Bảng nối chân ESP 32

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Thiết bị** | |  | | --- | | **Chân ESP32-CAM** |  |  | | --- | |  | | |  | | --- | | **Chân Thiết bị** |  |  | | --- | |  | | | **Mô tả** | | --- |  |  | | --- | |  | |
| **Relay** | IO13 | |  | | --- | | IN |  |  | | --- | |  | | |  | | --- | | Điều khiển relay |  |  | | --- | |  | |
|  | 5V | VCC | |  | | --- | | Nguồn 5V cho relay |  |  | | --- | |  | |
|  | GND | GND | |  | | --- | | Nối đất |  |  | | --- | |  | |
| **Buzzer** | |  | | --- | | IO12 |  |  | | --- | |  | | (+) | Điều khiển Buzzer |
|  | |  | | --- | | GND |  |  | | --- | |  | | (-) | Nối đất |
| **LCD OLED SSD1306** | IO14 | SDA | |  | | --- | | Giao tiếp SPI Data (SDA) |  |  | | --- | |  | |
|  | IO15 | SCL | |  | | --- | | Giao tiếp SPI Clock (SCL) |  |  | | --- | |  | |
|  | |  | | --- | | GND |  |  | | --- | |  | | |  | | --- | | GND |  |  | | --- | |  | | |  | | --- | | Nối đất |  |  | | --- | |  | |
|  | 5V | VCC | |  | | --- | | Nguồn 5V |  |  | | --- | |  | |
| **Nút nhấn (Button)** | IO2 | |  | | --- | | Một đầu nút nhấn |  |  | | --- | |  | | |  | | --- | | Dùng để reset hoặc điều khiển |  |  | | --- | |  | |
|  | GND | |  | | --- | | Đầu còn lại |  |  | | --- | |  | | |  | | --- | | Nối đất | |

Hình 3. Bản nối chân ESP 32-cam

3.2.1.4 Sơ đồ thuật toán



Hình 3. Sơ đồ thuật toán esp32

**Nguyên lý hoạt động:**

Bắt đầu: Hệ thống khởi động và bắt đầu đọc dữ liệu từ các cảm biến.

Đọc dữ liệu cảm biến: Hệ thống lần lượt đọc giá trị từ các cảm biến ánh sáng, nhiệt độ và khí gas.

Kiểm tra điều kiện:

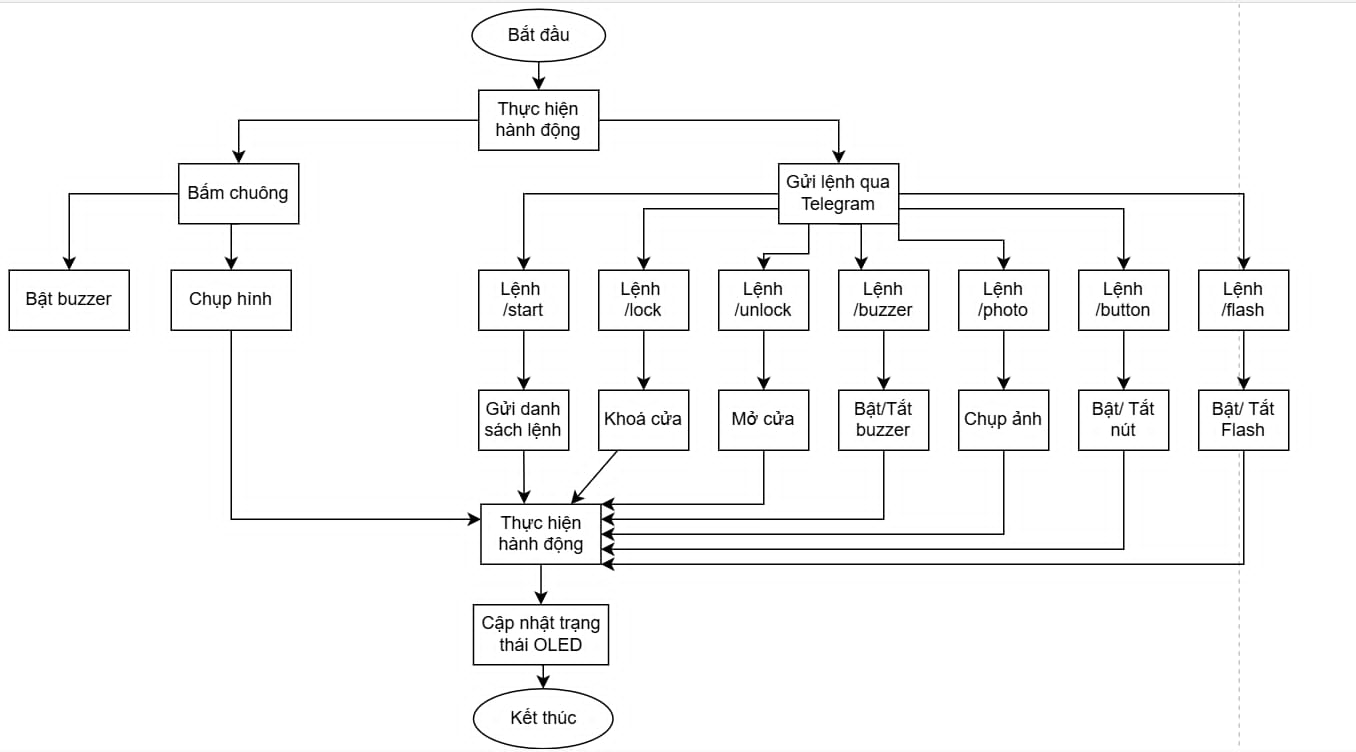
Cảm biến ánh sáng: Nếu ánh sáng yếu hơn một ngưỡng nhất định, hệ thống sẽ bật đèn. Ngược lại, nếu ánh sáng đủ, hệ thống sẽ tắt đèn.

Cảm biến nhiệt độ: Nếu nhiệt độ vượt quá 30 độ C, hệ thống sẽ kích hoạt còi báo động (buzzer).

Cảm biến khí gas: Nếu nồng độ khí gas vượt quá 2000 đơn vị (đơn vị chưa được xác định rõ), hệ thống sẽ kích hoạt còi báo động (buzzer).

Cập nhật trạng thái: Sau khi thực hiện các hành động điều khiển, hệ thống sẽ cập nhật trạng thái mới lên các thiết bị hiển thị như màn hình LCD hoặc gửi dữ liệu lên máy chủ web để theo dõi và quản lý từ xa.

Kết thúc: Hệ thống sẽ quay lại bước 1 để tiếp tục kiểm tra và điều khiển



Hình 3. Sơ đồ thuật toán esp 32 cam

**Nguyên lý hoạt động:**

Bắt đầu: Hệ thống khởi động và sẵn sàng nhận lệnh.

Nhận lệnh: Hệ thống liên tục lắng nghe các lệnh được gửi từ ứng dụng Telegram.

Phân tích lệnh: Khi nhận được lệnh, hệ thống sẽ phân tích nội dung lệnh để xác định hành động cần thực hiện.

Các loại lệnh:

/start: Bắt đầu hoặc khởi động một chức năng nào đó.

/lock: Khóa một thiết bị (ví dụ: cửa).

/unlock: Mở khóa một thiết bị.

/buzzer: Bật/tắt còi báo động.

/photo: Chụp ảnh.

/button: Điều khiển một nút nhấn.

/flash: Điều khiển đèn flash.

Thực hiện hành động: Tùy thuộc vào lệnh nhận được, hệ thống sẽ thực hiện các hành động tương ứng:

Bật/tắt còi báo động: Điều khiển mạch điện để bật hoặc tắt còi báo động.

Chụp ảnh: Kích hoạt camera để chụp ảnh và lưu trữ.

Khóa/mở cửa: Điều khiển động cơ hoặc solenoid để khóa hoặc mở cửa.

Điều khiển nút nhấn: Mô phỏng việc nhấn nút để kích hoạt các chức năng khác.

Điều khiển đèn flash: Bật/tắt đèn flash.

Cập nhật trạng thái: Sau khi thực hiện xong lệnh, hệ thống có thể cập nhật trạng thái lên màn hình OLED để người dùng biết được kết quả.

Kết thúc: Hệ thống quay lại bước 2 để tiếp tục lắng nghe lệnh mới.

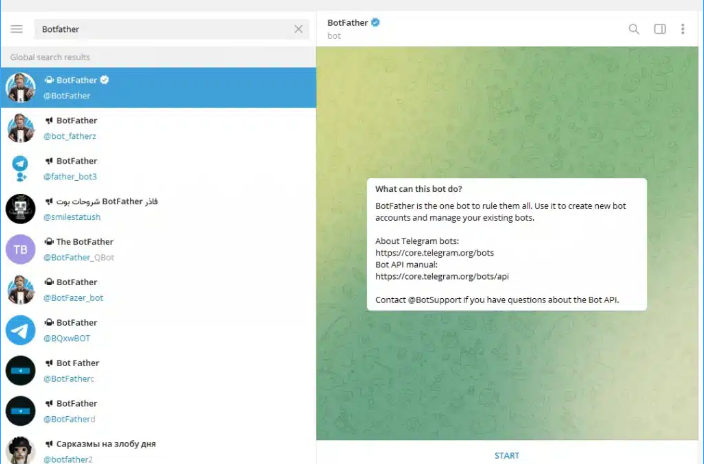
3.2.1.5 Tạo API Botchat Telegram

Bot Telegram là một tính năng giống như robot được cài đặt trên ứng dụng nhắn tin Telegram giúp người dùng tạo lập và quản lý một cách dễ dàng.

Người tạo ra Bot Telegram có thể thêm các chức năng cho Bot như: cập nhật tin tức, tạo GIF, trả lời tin nhắn, Hiện tại, việc tạo Bot trên Telegram là cực kỳ dễ dàng và bất cứ ai cũng có thể làm được cả. Việc khó nhất trong cách tạo lập Bot Telegram chính là thêm các chức năng giúp Bot gửi thông báo cho bạn.

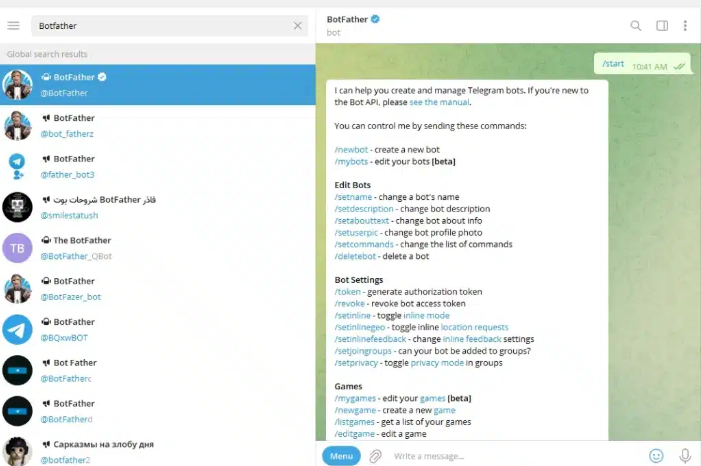
Tạo 1 con Bot trên Telegram gồm các bước như sau:

**Bước 1**: Trên **Telegram**, tìm kiếm **Botfather** và chọn **Botfathe**r có tích xanh.

****

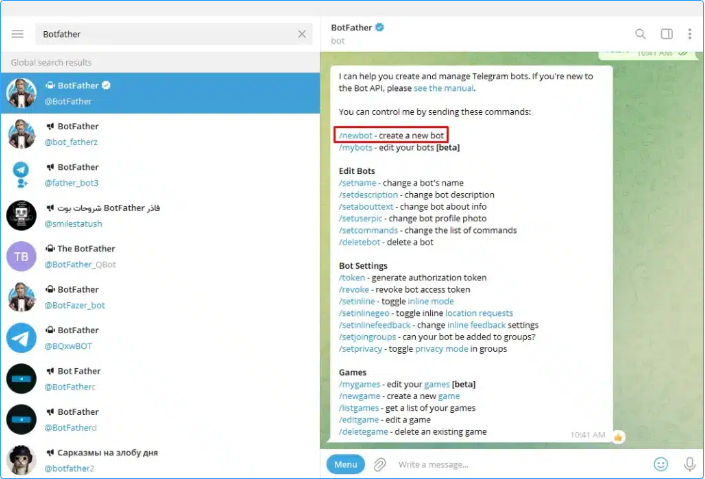
Hình 3. Hướng dẫn cài đặt bot trên telegram

**Bước 2**: Nhập lệnh /**start** và nhấn **Enter** để bắt đầu tạo bot mới.



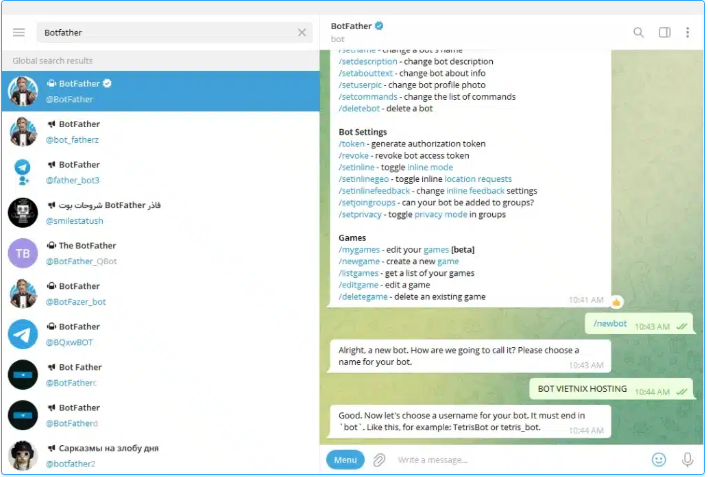
Hình 3. Tạo bot mới

**Bước 3**: Nhấn vào lệnh /**newbot**.



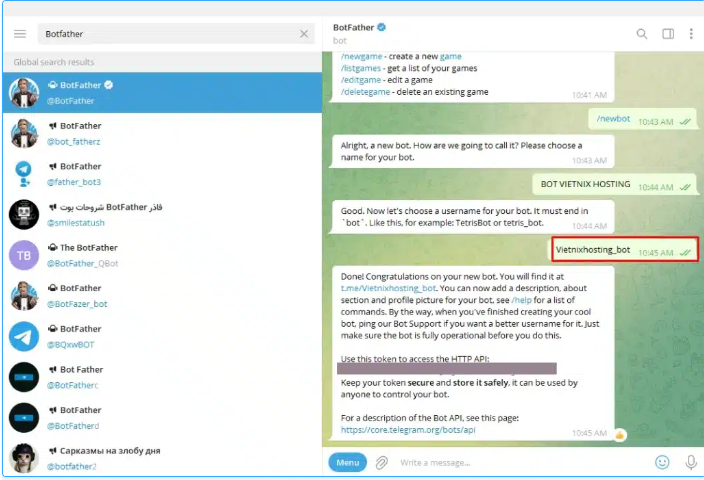
Hình 3. Bắt đầu tạo bot mới

**Bước 4**: Nhập **tên bot** của bạn vào ô tin nhắn và nhấn **Enter.**



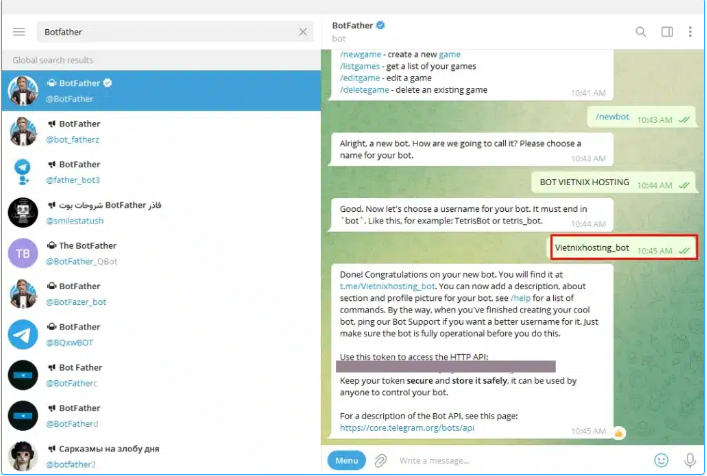
Hình 3. Tạo một bot mới trên Telegram bằng cách sử dụng BotFather

**Bước 5**: Sau đó, nhập tên **user name** cho bot của bạn và nhấn **Enter**



Hình 3. Cài đặt user name cho bot

**Bước 6**: Lúc này hệ thống sẽ gửi tin nhắn báo **tạo bot Telegram** thành công.



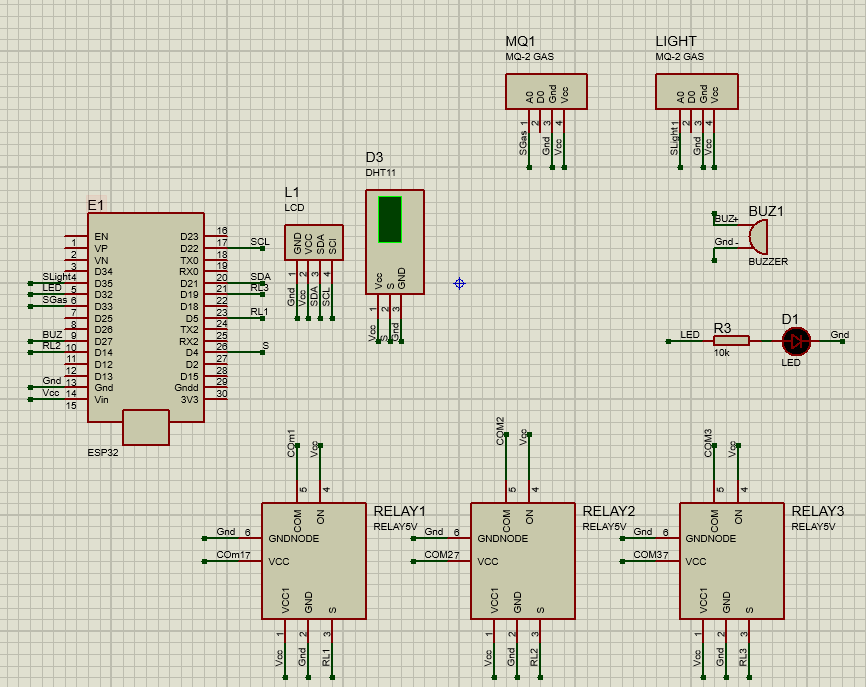
Hình 3. Tạo bot thành công

Chatbot Telegram đóng vai trò là giao diện người dùng trực quan, giúp người vận hành có thể giám sát và điều khiển hệ thống qua các lệnh đơn giản như bật/tắt bơm nước, kiểm tra trạng thái thiết bị và nhận thông báo tự động. Đây là giải pháp hiệu quả cho các ứng dụng như hệ thống tưới cây tự động, giám sát môi trường nhà kính hoặc điều khiển các thiết bị thông minh trong gia đình.

Dự án này không chỉ mang tính ứng dụng cao mà còn mở ra nhiều tiềm năng phát triển trong tương lai. Việc mở rộng thêm các tính năng như học máy (Machine Learning) để dự đoán tình trạng cây trồng, hay tích hợp thêm các cảm biến và thiết bị khác sẽ là những hướng phát triển đầy hứa hẹn.

Hy vọng rằng dự án này không chỉ giúp bạn hiểu rõ hơn về lập trình nhúng, giao tiếp API và xây dựng chatbot, mà còn truyền cảm hứng để bạn tiếp tục nghiên cứu và phát triển thêm những ứng dụng thông minh, hữu ích khác.

3.2.1.6 Vẽ mạch

. 

Hình 3. Mạch Proteus

Khởi tạo hệ thống: Khi bật nguồn, ESP32 kết nối với mạng WiFi, khởi tạo màn hình OLED và các thiết bị ngoại vi (camera, relay, buzzer, nút bấm, đèn flash).

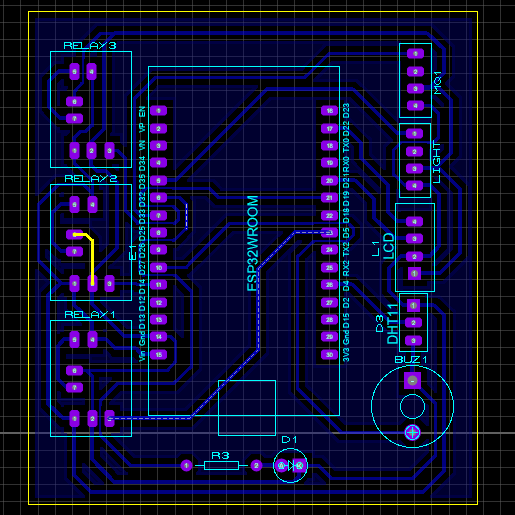
Điều khiển cửa: Khi nhận được lệnh từ Telegram ("/lock" hoặc "/unlock"), hệ thống sẽ khóa hoặc mở cửa và thông báo qua Telegram.

Chụp ảnh và gửi Telegram: Khi nhận được lệnh "/photo" từ Telegram, hệ thống chụp ảnh qua camera và gửi ảnh đó lên Telegram.

Theo dõi nút bấm: Khi nút bấm được nhấn, hệ thống sẽ kích hoạt buzzer và chụp ảnh gửi qua Telegram.

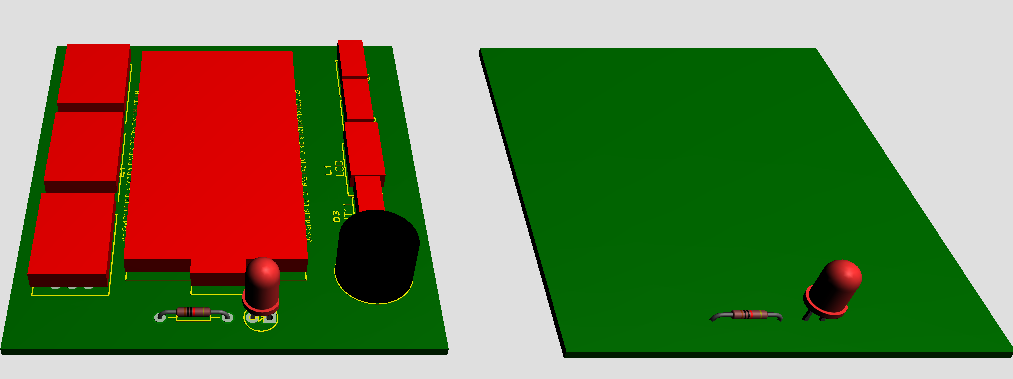
Màn hình OLED: Hiển thị trạng thái của hệ thống, ví dụ như "Flash On", "Buzzer Off", "Door Locked", v.v.

3.2.1.7 Mạch PCB



Hình 3. Mạch PCB

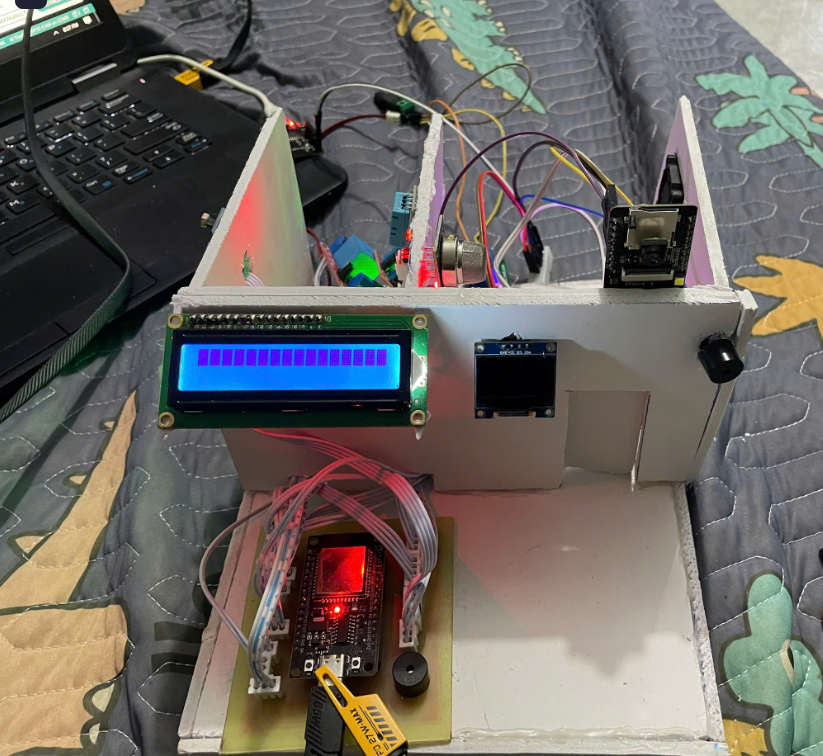
3.2.1.8 Mạch 3D



Hình 3. Mạch in 3D

3.2.2 Hoàn thiện sản phẩm

3.2.2.1 Sản phẩm thực tế



Hình 3. Sản Phẩm

Về sản phẩm thực tế thì hệ thống nhà thông minh bao gồm các chức năng:

- **Mở khóa cửa bằng ESP32-CAM**:

Chương trình tạo ra một hệ thống nhà thông minh với khả năng điều khiển từ xa qua Telegram. Nó cho phép người dùng mở khóa và khóa cửa thông qua các lệnh /lock và /unlock. Camera ESP32-CAM có thể chụp ảnh và gửi qua Telegram khi nhận lệnh /photo, đồng thời bật đèn flash để hỗ trợ chụp ảnh trong môi trường tối. Chương trình còn có chức năng bật tắt còi báo động (buzzer) qua lệnh /buzzer, và theo dõi trạng thái của một nút nhấn để tự động chụp ảnh khi có tín hiệu từ nút. Các trạng thái của hệ thống như cửa, đèn flash, còi báo được hiển thị trên màn hình OLED. Hệ thống này giúp tăng cường bảo mật và giám sát trong các ứng dụng nhà thông minh.

- **Có thể tự động bật đèn khi trời tối**:

Chức năng của đèn LED trong chương trình trên là dùng để cảnh báo và hỗ trợ chiếu sáng trong các tình huống nhất định. Cụ thể, đèn LED (được kết nối với chân GPIO 4) có thể bật tắt qua lệnh /flash trên Telegram, giúp người dùng bật đèn cảnh báo hoặc chiếu sáng khi cần thiết. Khi có tín hiệu từ nút nhấn, đèn LED cũng sẽ được bật để hỗ trợ chụp ảnh trong môi trường thiếu sáng. Chức năng này đóng vai trò quan trọng trong việc tăng cường hiệu quả hoạt động của hệ thống trong các tình huống khẩn cấp hoặc khi cần ghi lại hình ảnh trong điều kiện ánh sáng yếu.

- **Tự động bật còi báo khi khí gas và nhiệt độ vượt ngưỡng**:

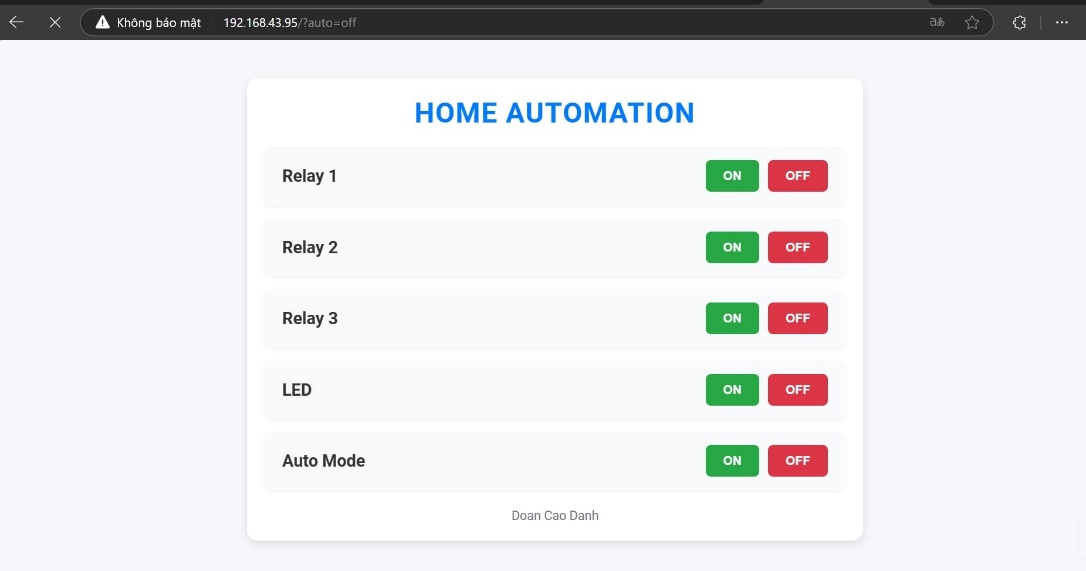
Sử dụng cảm biến MQ-2 và DHT để liên tục đo nồng độ khí gas và nhiệt độ trong không khí ở trong nhà, khi khí gas và nhiệt độ vượt ngưỡng nhất định thì cảm biến sẽ gửi dữ liệu về ESP32 và yêu cầu bật quạt, còi báo động để cảnh báo an toàn cũng như đảm bảo thiệt hại ít nhất về tài sản trong nhà.

**- Điều khiển và hiển thị dữ liệu đo được thông qua webserver và màn hình LCD:**

Chương trình trên điều khiển một hệ thống tự động hóa nhà thông minh sử dụng ESP32 và MicroPython. Các cảm biến được kết nối với ESP32 bao gồm cảm biến nhiệt độ và độ ẩm DHT11, cảm biến khí gas MQ-2, và cảm biến ánh sáng LDR. Chương trình thu thập dữ liệu từ các cảm biến này và hiển thị thông tin lên webser và màn hình LCD.

Ngoài ra, hệ thống hỗ trợ điều khiển các thiết bị như quạt, đèn LED thông qua các relay, có thể bật/tắt thủ công hoặc tự động dựa trên các giá trị cảm biến, đặc biệt là cảm biến ánh sáng. Các thao tác điều khiển được thực hiện thông qua giao diện web, cho phép người dùng bật/tắt các thiết bị từ xa. Hệ thống cũng cung cấp chế độ tự động cho phép bật/tắt đèn LED dựa trên mức độ ánh sáng xung quanh. Chương trình này mang lại sự tiện lợi trong việc giám sát và điều khiển các thiết bị điện trong nhà thông qua kết nối mạng.

3.2.2.2 WebServer



Hình 3. WebServer nhà thông minh

WebServer được phát triển để giúp người dùng có thể xem các thông tin của ngôi nhà mình như:   
 + Nhiệt độ, độ ẩm.

+ Nồng độ khí gas.

+ Giám sát nhà.

Đồng thời WebServer cũng dùng để điều khiển các thiết bị trong nhà như: mở cửa, bật đèn, bật quạt.

3.2.2.3 Bot Telegaram



Hình 3. Bot Telegram hoạt động

* **Nội dung cuộc trò chuyện:**

Cuộc trò chuyện chủ yếu xoay quanh việc điều khiển các thiết bị trong nhà thông qua các lệnh được gửi trực tiếp vào nhóm chat. Các lệnh này được nhận diện bằng dấu / (ví dụ: /flash, /unlock).

* **Các thiết bị được điều khiển:**

Cửa: Có thể khóa và mở khóa bằng lệnh /lock và /unlock.

Đèn flash: Có thể bật và tắt bằng lệnh /flash.

* **Tương tác của người dùng:**

Người dùng (có tên "nhacuadanh" và được xác định là admin) gửi các lệnh để điều khiển các thiết bị.

Bot hoặc hệ thống tự động hóa sẽ nhận lệnh và thực hiện tương ứng. Ví dụ, khi người dùng gửi lệnh /flash, đèn flash sẽ bật sáng.

* **Phỏng đoán về hệ thống:**

Dựa trên các thông tin trong hình, có thể suy ra rằng nhóm đang sử dụng một hệ thống tự động hóa nhà được tích hợp với Telegram. Hệ thống này cho phép người dùng điều khiển các thiết bị trong nhà một cách trực quan và tiện lợi thông qua các lệnh chat.

* **Các thành phần có thể có trong hệ thống:**

Một bot Telegram: Bot này được lập trình để nhận các lệnh từ nhóm chat và gửi các lệnh điều khiển đến các thiết bị thực tế.

Một trung tâm điều khiển: Đây có thể là một máy tính hoặc một thiết bị IoT (Internet of Things) được kết nối với các thiết bị trong nhà và bot Telegram. Trung tâm điều khiển sẽ nhận các lệnh từ bot và thực hiện các tác vụ tương ứng.

Các thiết bị thông minh: Bao gồm khóa cửa thông minh, đèn thông minh, và có thể cả các thiết bị khác như ổ cắm thông minh, điều hòa nhiệt độ... Các thiết bị này được kết nối với trung tâm điều khiển và có thể được điều khiển từ xa.

* **Ưu điểm của hệ thống này:**

Tiện lợi: Người dùng có thể điều khiển các thiết bị trong nhà mọi lúc mọi nơi chỉ cần có kết nối internet.

Linh hoạt: Hệ thống có thể được tùy chỉnh để đáp ứng các nhu cầu khác nhau của người dùng.

Dễ sử dụng: Giao diện chat đơn giản, dễ hiểu, không yêu cầu người dùng phải có kiến thức kỹ thuật sâu.

* **Nhược điểm tiềm ẩn:**

Bảo mật: Nếu không được bảo mật tốt, hệ thống có thể bị tấn công và người ngoài có thể điều khiển các thiết bị trong nhà.

Phụ thuộc vào kết nối internet: Nếu mất kết nối internet, hệ thống sẽ không hoạt động.

## Ghi chú và hạn chế

Mặc dù chương trình cung cấp nhiều tính năng hữu ích cho hệ thống tự động hóa, nhưng vẫn có một số hạn chế cần lưu ý. Đầu tiên, việc sử dụng các cảm biến như DHT11, MQ-2 và LDR có thể gây ra một số sai sót về độ chính xác và độ ổn định của dữ liệu, đặc biệt là khi môi trường thay đổi nhanh chóng. Cảm biến DHT11, ví dụ, chỉ có độ chính xác thấp và không thích hợp cho các ứng dụng yêu cầu dữ liệu nhiệt độ và độ ẩm chính xác cao. Thứ hai, hệ thống không có khả năng xử lý các tình huống phức tạp hoặc kết nối với nhiều thiết bị đồng thời, điều này có thể gây ra tình trạng trễ hoặc gián đoạn trong quá trình điều khiển. Việc sử dụng ESP32 với giới hạn tài nguyên cũng có thể ảnh hưởng đến hiệu suất khi kết nối nhiều cảm biến và thiết bị, đặc biệt là khi hệ thống hoạt động với nhiều tác vụ đồng thời. Cuối cùng, mặc dù hệ thống hỗ trợ điều khiển từ xa qua giao diện web, nhưng tính bảo mật của hệ thống có thể chưa được tối ưu hóa, khiến nó dễ bị tấn công từ các nguồn không xác định nếu không có các biện pháp bảo mật thích hợp.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1 Kết luận**

Sau một thời gian tìm hiểu, tham khảo tài liệu từ nhiều nguồn khác nhau, cũng như được sự giúp đỡ của giảng viên bộ môn và bạn bè, em đã hoàn thành đề tài: **“**Nhà Thông Minh**”**

**Nhìn chung đề tài của em có ưu điểm:**

Mạch có cấu tạo và hoạt động đơn giản không phức tạp nhưng tính chính xác và khả năng áp dụng thực tế cao.

Mạch được thiết kế với các module có biến trở, ta có thể dễ dàng điều chỉnh tùy thuộc vào yêu cầu và trường hợp cụ thể.

Các thiết bị dễ mua được ngoài thị trường, giá thành rẻ.

Dễ lắp ráp đối với người mới tập làm mạch điện tử.

Có thể mở rộng và phát triển dễ dàng.

Đề tài trình bày được những nguyên lý cơ bản nhất của một hệ thống nhà thông minh, từ đây có thể phát triển đề tài thành hệ thống cảnh báo thông minh phù hợp với nhu cầu của xã hội.

**2 Ứng dụng thực tế của mạch:**

Ứng dụng thực tế của chương trình này có thể được triển khai trong các hệ thống tự động hóa nhà thông minh, nơi người dùng muốn giám sát và điều khiển các thiết bị như quạt, đèn, hoặc máy bơm thông qua cảm biến nhiệt độ, độ ẩm, ánh sáng và khí gas. Chẳng hạn, trong một ngôi nhà thông minh, hệ thống có thể tự động bật quạt khi nhiệt độ vượt ngưỡng 30°C hoặc tắt đèn khi không có ánh sáng tự nhiên, giúp tiết kiệm năng lượng. Ngoài ra, chương trình cũng có thể được ứng dụng trong các khu vực cần kiểm soát không khí như nhà xưởng, phòng thí nghiệm hoặc các môi trường làm việc có yêu cầu nghiêm ngặt về khí gas. Khi phát hiện mức khí gas vượt quá ngưỡng, hệ thống sẽ tự động kích hoạt buzzer hoặc gửi thông báo đến người quản lý qua webserver để xử lý kịp thời. Trong các ứng dụng bảo mật, hệ thống có thể sử dụng cảm biến vân tay và nhận diện khuôn mặt để điều khiển cửa ra vào, giúp nâng cao tính an toàn cho các khu vực nhạy cảm như văn phòng, kho lưu trữ hay nhà riêng. Việc điều khiển qua Telegram mang lại sự tiện lợi, cho phép người dùng giám sát và điều khiển từ xa, ngay cả khi không có mặt tại vị trí của hệ thống

**3 Hướng phát triển**

Với đề tài này nếu có đủ thời gian nghiên cứu thì có thể mở rộng ra ngoài rộng rãi hơn như trong nhà ở, các căn hộ, chung cư cao cấp,..

# TÀI LIỆU THAM KHẢO

[1]: https://arduinokit.vn/thiet-ke-nha-thong-minh-bang-arduino-su-dung-esp32-va-blynk-2-0/

[2]: https://www.iotzone.vn/esp32/du-an-esp32/esp32-smart-hom

[3]:https://randomnerdtutorials.com/micropython-esp32-esp8266-dht11-dht22-web-server/