BỘ GIÁO DỤC VÀ ĐÀO TẠO ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

---o0o---

TIỂU LUẬN MÔN PHÁT TRIỂN ỨNG DỤNG IOT

ĐỀ TÀI: HỆ THỐNG KHÓA CỬA THÔNG MINH DỰA TRÊN ESP32 VÀ RFID

Thừa Thiên Huế, tháng 04 năm 2025

LỜI CAM ĐOAN

Tôi cam đoan rằng đề tài "**Sử dụng module RFID để mở khóa cửa, tích hợp xác thực qua Wi-Fi hoặc Bluetooth**" do tôi tự thực hiện dưới sự hướng dẫn của giảng viên **Võ Việt Dũng**. Các nội dung nghiên cứu, số liệu và kết quả là trung thực. Các số liệu, công trình sử dụng của tác giả khác đều được trích dẫn nguồn gốc rõ ràng. Tất cả phần mềm, mã nguồn được sử dụng trong phần tiểu luận này đều có giấy phép miễn phí. Nếu phát hiện có bất kì sự gian lận nào, tôi xin chịu hoàn toàn trách nhiệm

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Thầy **Võ Việt Dũng**, giảng viên học phần Phát Triển Ứng Dụng IoT, người đã tận tình giảng dạy, hướng dẫn và truyền đạt những kiến thức quý báu trong suốt quá trình học tập.Nhờ sự tận tâm và chỉ dẫn của Thầy, em đã có cơ hội tiếp cận và hiểu sâu hơn về những nguyên lý cơ bản cũng như các ứng dụng thực tiễn trong lĩnh vực IoT. Em cũng xin cảm ơn Thầy đã luôn khuyến khích, động viên và hỗ trợ em vượt qua những khó khăn trong quá trình nghiên cứu và triển khai dự án này.

Mục Lục

MỞ ĐẦU

Trong kỷ nguyên **Internet vạn vật (IoT)**, các thiết bị gia dụng ngày càng được tích hợp công nghệ thông minh nhằm nâng cao mức độ tiện nghi và an toàn cho người sử dụng. Một trong những thiết bị tiêu biểu của xu hướng này là khóa cửa thông minh – thiết bị có khả năng thay thế hoàn toàn ổ khóa cơ truyền thống bằng các phương thức mở cửa hiện đại như thẻ từ, mã số, vân tay hoặc kết nối không dây. Không chỉ đơn thuần đảm nhận vai trò đóng/mở chốt cửa, khóa thông minh còn có thể kết nối mạng, cho phép giám sát và điều khiển từ xa thông qua điện thoại thông minh hoặc nền tảng đám mây. So với các ổ khóa cơ học truyền thống, hệ thống khóa cửa thông minh mang lại nhiều ưu điểm vượt trội về tính bảo mật, khả năng kiểm soát truy cập, và sự tiện dụng trong vận hành. Ví dụ, thẻ từ RFID với mã định danh duy nhất giúp ngăn chặn việc sao chép chìa khóa trái phép; người dùng có thể dễ dàng vô hiệu hóa thẻ bị mất, cấp quyền ra vào tạm thời cho khách, và theo dõi lịch sử mở khóa.

Trong phạm vi đề tài này, chúng tôi đề xuất thiết kế và mô phỏng một **hệ thống khóa cửa thông minh dựa trên vi điều khiển ESP32 và công nghệ RFID**. Vi điều khiển **ESP32** được lựa chọn nhờ khả năng tích hợp kết nối Wi-Fi và Bluetooth, hiệu năng xử lý cao và chi phí hợp lý, rất phù hợp cho các ứng dụng IoT nhúng. Trong khi đó, RFID (Radio-Frequency Identification) là công nghệ nhận dạng không dây, không tiếp xúc, cho phép người dùng sử dụng thẻ từ làm "chìa khóa" điện tử an toàn. Hệ thống kết hợp giữa ESP32 và đầu đọc RFID mang lại một giải pháp khóa cửa tiện lợi, dễ quản lý và có khả năng mở rộng, trong đó ESP32 đảm nhiệm vai trò xử lý logic điều khiển, đồng thời duy trì kết nối với nền tảng IoT (Blynk) để người dùng có thể giám sát và điều khiển từ xa. Việc xây dựng mô hình trên trình mô phỏng Wokwi giúp kiểm chứng hoạt động và hỗ trợ quá trình phát triển mà không cần phần cứng thực.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

An ninh nhà ở luôn là mối quan tâm hàng đầu, đặc biệt trong bối cảnh đô thị hóa và sự phát triển của Internet vạn vật (IoT). Thực tế cho thấy không ít người quên khóa cửa khi ra khỏi nhà hoặc không chắc chắn liệu cửa đã khóa hay chưa – đây là một yếu tố đe dọa đến an ninh gia đình. Hệ thống khóa cửa truyền thống dùng chìa cơ có nhiều hạn chế như dễ thất lạc hoặc sao chép chìa. Do đó, nhu cầu về một khóa cửa thông minh có thể tự động hóa và tăng cường bảo mật ngày càng trở nên cấp thiết. Khóa cửa thông minh tích hợp công nghệ IoT cho phép người dùng giám sát và điều khiển cửa từ xa, giúp kiểm tra trạng thái và khóa cửa qua điện thoại khi cần thiết, giảm thiểu rủi ro do quên khóa.

1.2. Mục tiêu nghiên cứu

Mục tiêu của đề tài bao gồm: (1) Thiết kế một hệ thống khóa cửa thông minh sử dụng vi điều khiển ESP32 kết hợp đầu đọc RFID để điều khiển khóa điện, cho phép mở khóa bằng thẻ RFID; (2) Tích hợp kết nối không dây (Wi-Fi, Bluetooth) nhằm mở khóa hoặc giám sát từ xa thông qua nền tảng IoT (ứng dụng Blynk); (3) Mô phỏng hoạt động của hệ thống trên môi trường giả lập phần cứng Wokwi, bao gồm cả việc mô phỏng các cảm biến phụ trợ; (4) Phân tích kết quả hoạt động, đánh giá ưu nhược điểm kỹ thuật và đề xuất hướng hoàn thiện. Hệ thống đề xuất sẽ cho phép mở khóa bằng thẻ RFID hoặc qua ứng dụng di động trên nền tảng đám mây, đồng thời ghi nhận trạng thái khóa và các sự kiện ra/vào để nâng cao an ninh.

1.3. Phạm vi và giới hạn

Đề tài tập trung vào mô hình mẫu (prototype) khóa cửa thông minh ở mức độ phòng thí nghiệm/giả lập. Hệ thống được xây dựng và thử nghiệm trên trình mô phỏng Wokwi, do đó không triển khai trên thiết bị thật. Phạm vi bao gồm phần cứng ảo (ESP32, module RFID RC522, khóa điện từ hoặc servo mô phỏng chốt cửa, các cảm biến phụ nếu có) và phần mềm (chương trình nhúng trên ESP32, ứng dụng Blynk). Giới hạn đề tài: chưa tập trung vào vấn đề bảo mật nâng cao (như mã hóa RFID, xác thực hai yếu tố) và độ bền phần cứng do không triển khai thực tế. Tuy nhiên, mô hình mô phỏng sẽ cố gắng phản ánh đầy đủ chức năng chính của hệ thống khóa thông minh, tạo tiền đề cho việc phát triển và mở rộng trong tương lai ngoài thực tế.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1. Khóa cửa thông minh

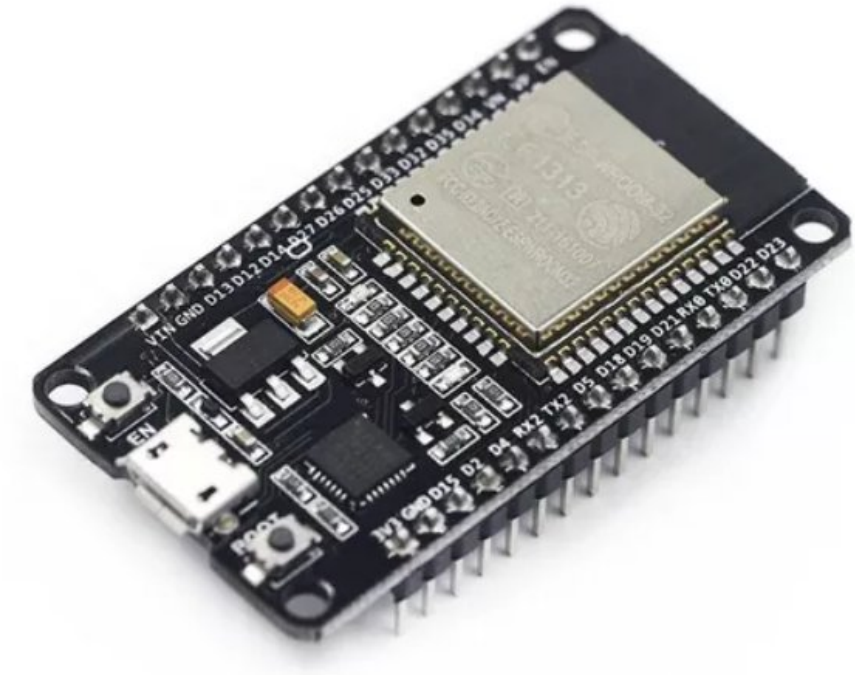
Khóa cửa thông minh là một loại khóa điện tử (electromechanical lock) được thiết kế để điều khiển đóng/mở chốt cửa thông qua **các tín hiệu điện tử hoặc không dây** thay vì chìa khóa cơ thông thường. Cụ thể, khóa thông minh có thể hoạt động dựa trên bàn phím số, cảm biến sinh trắc học (ví dụ: vân tay), thẻ từ (RFID), Bluetooth hoặc Wi-Fi từ điện thoại di động đã đăng ký. Nhờ được kết nối và tích hợp công nghệ, khóa thông minh mang lại trải nghiệm tiện lợi cho người dùng và nâng cao mức độ bảo mật so với ổ khóa truyền thống. Thay vì phải mang theo chìa khóa vật lý (có thể bị mất hoặc sao chép), người dùng có thể mở khóa bằng mã PIN, dấu vân tay hoặc điện thoại một cách nhanh chóng.

Hình 2.1. Một số ảnh minh họa khóa thông minh
(Nguồn: https://lumi.vn/khoa-cua-thong-minh.html)

Ưu điểm chính của khóa cửa thông minh gồm có: (1) **Tăng cường an ninh** – nhiều mẫu khóa ứng dụng xác thực nâng cao (vân tay, khuôn mặt, mã OTP) đảm bảo chỉ người được ủy quyền mới vào được nhà. Khóa thông minh giảm thiểu rủi ro sao chép chìa và có thể gửi **thông báo thời gian thực** về điện thoại khi có truy cập trái

phép. (2) Tiện lợi và không cần chìa khóa – người dùng không lo quên hay mất chìa; có thể mở khóa từ xa qua điện thoại hoặc tự động khóa khi rời đi, đồng thời cấp quyền truy cập tạm thời cho khách thông qua ứng dụng. (3) Tích hợp hệ sinh thái nhà thông minh – khóa thông minh có thể kết nối với các thiết bị IoT khác (camera, chuông cửa, báo động) và nền tảng nhà thông minh (Google Assistant, Amazon Alexa), tạo nên hệ thống an ninh đồng bộ. Nhờ đó, chủ nhà có thể thiết lập kịch bản tự động, chẳng hạn khóa cửa sẽ tự mở khi nhận diện chủ nhà về hoặc kích hoạt báo động nếu phát hiện đột nhập. Thực tế, khóa thông minh đang dần trở thành xu hướng chủ đạo với thị trường dự báo tăng trưởng mạnh, phản ánh nhu cầu cao về giải pháp ra vào không chìa (keyless entry) và quản lý truy cập linh hoạt trong kỷ nguyên IoT. Tuy nhiên, cần lưu ý một số thách thức: khóa thông minh phụ thuộc vào nguồn điện – nếu mất điện hoặc cạn pin có thể cần phương án dự phòng (chìa cơ hoặc pin dự phòng). Ngoài ra, do kết nối mạng, chúng có nguy cơ bị tấn công mạng nếu bảo mật không tốt. Vì vậy, khi triển khai thực tế, cần chọn các giải pháp có mã hóa và cơ chế an ninh mạnh, cũng như thiết kế tính năng cảnh báo, ghi log truy cập để tăng độ tin cậy.

2.2. Vi điều khiển ESP32



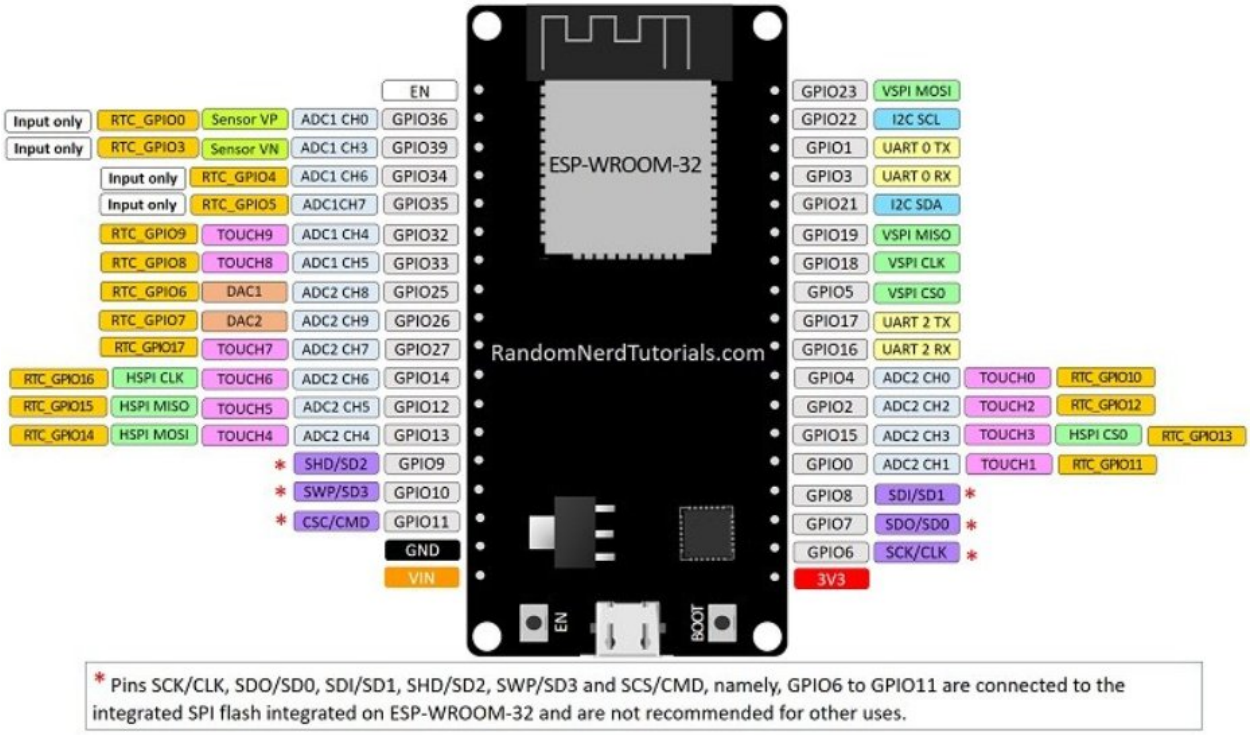
Hình 2.2. Minh họa thiết bị ESP32

(Nguồn: https://articulo.mercadolibre.com.mx/MLM-736691552-modulo-esp32-wifi-bluetooth-nodemcu-esp8266-arduino-esp-32-p-_JM)

ESP32 (Espressif) là một vi điều khiển nhúng hiệu năng cao, giá thành thấp, tích hợp sẵn kết nối Wi-Fi và Bluetooth hướng tới các ứng dụng IoT. Về cấu trúc phần cứng, ESP32 được trang bị CPU lõi kép Xtensa 32-bit LX6, tốc độ hoạt động lên tới 240 MHz, kèm theo 520 KB SRAM và bộ nhớ ROM tích hợp. Nhờ có hai nhân xử lý (dual-core) và khả năng quản lý năng lượng linh hoạt, ESP32 có thể thực thi đa tác vụ (như đọc cảm biến đồng thời duy trì kết nối mạng) một cách hiệu quả. Chip này hỗ trợ đầy đủ chuẩn Wi-Fi 802.11 b/g/n (băng tần 2.4 GHz) và Bluetooth 4.2 Dual Mode (Classic + BLE), cho phép nó giao tiếp không dây đa dạng.

ESP32 được đánh giá là vi điều khiển "giàu tính năng" (feature-rich MCU) với mức độ tích hợp cao: nó tích hợp sẵn các bộ khuếch đại, chuyển mạch ăng-ten, balun RF, bộ quản lý nguồn... trên một module nhỏ gọn. Thiết kế này giúp đơn giản hóa mạch ngoại vi và giảm yêu cầu linh kiện rời. ESP32 còn nổi bật với khả năng hoạt động trong môi trường khắc nghiệt (nhiệt độ -40°C đến +125°C) và chế độ tiết kiệm năng lượng linh hoạt (nhiều mức độ ngủ/nghỉ, điều chỉnh xung nhịp động) cho các ứng dụng thiết bị di động, đeo được.

Với những ưu điểm trên, ESP32 trở thành nền tảng lý tưởng cho dự án khóa cửa thông minh. Module ESP32-DevKit phổ biến có đầy đủ chân GPIO, ADC, DAC, UART, SPI, I2C… giúp kết nối linh hoạt với cảm biến RFID, động cơ/relay điều khiển khóa, và giao tiếp mạng. Trong hệ thống này, ESP32 đóng vai trò bộ điều khiển trung tâm, tiếp nhận tín hiệu từ đầu đọc RFID, xử lý logic xác thực, điều khiển khóa chốt cửa, đồng thời quản lý kết nối Wi-Fi để giao tiếp với dịch vụ đám mây (Blynk) nhằm phục vụ chức năng giám sát/tương tác từ xa.

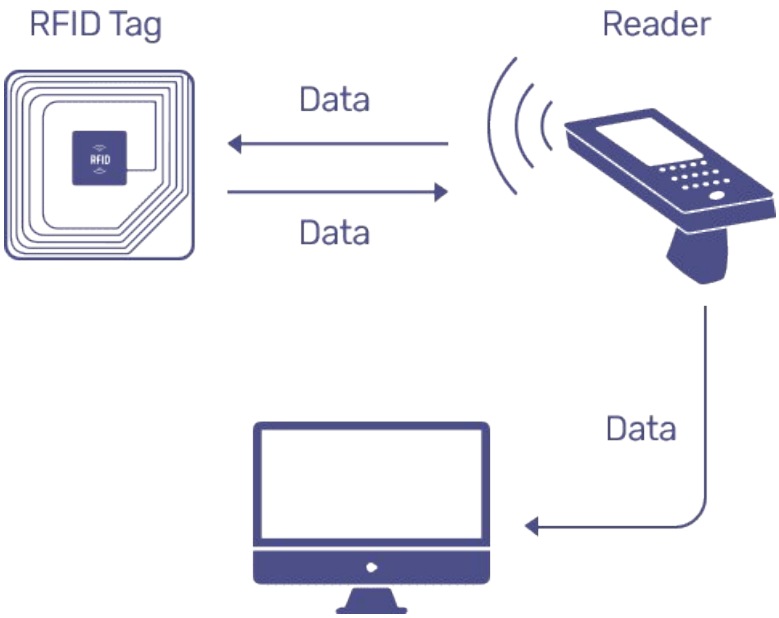


Hình 2.3. Minh họa sơ đồ chân của mạch ESP32 – Devkit v1 - DOIT

(Nguồn: https://khuenguyencreator.com/tong-quan-ve-so-do-chan-esp32-va-ngoai-vi/s)

2.3. Nguyên lý RFID

RFID (Radio-Frequency Identification – nhận dạng tần số vô tuyến) là công nghệ nhận dạng đối tượng thông qua sóng radio. Hệ thống RFID cơ bản gồm hai thành phần chính: đầu đọc RFID (RFID reader) và thẻ RFID (RFID tag). Đầu đọc thường tích hợp một ăng-ten và một bộ thu phát (transceiver), có nhiệm vụ phát ra sóng điện từ tần số radio để "kích hoạt" thẻ RFID ở gần đó. Mỗi thẻ RFID gắn trên đối tượng cần nhận dạng chứa một vi mạch lưu trữ mã định danh (ID) duy nhất và một ăng-ten nhỏ. Khi thẻ nằm trong vùng phủ sóng của đầu đọc, năng lượng từ sóng radio sẽ được ăng-ten thẻ hấp thụ (đối với thẻ thụ động, không có nguồn pin). Thẻ sau đó phản hồi lại đầu đọc bằng cách phát ra tín hiệu mang mã ID của nó. Đầu đọc thu nhận tín hiệu phản hồi này qua ăng-ten của mình, giải điều chế và chuyển thành dữ liệu số (mã ID) để hệ thống xử lý.



Hình 2.4. Minh họa quy trình hệ thống nhận dạng tần số vô tuyến – RFID

(Nguồn: https://lidinco.com/cong-nghe-rfid-la-gi-2)

Quy trình hoạt động RFID có thể tóm tắt như sau: đầu đọc gửi sóng radio → thẻ RFID trong vùng phủ sóng được kích hoạt và truyền dữ liệu ID → đầu đọc nhận và giải mã ID. Các thẻ RFID thường dùng trong kiểm soát ra vào (như thẻ từ mở cửa) thuộc loại RFID thụ động (Passive RFID) tần số 13.56 MHz, không dùng pin mà lấy năng lượng từ sóng do đầu đọc cung cấp. Khoảng cách đọc thẻ thụ động thường ngắn (vài cm đến vài chục cm tùy loại thẻ và antenna). Ngược lại, RFID chủ động có pin riêng có thể phát tín hiệu mạnh hơn, cho tầm đọc xa hơn, nhưng đắt tiền hơn.

Trong hệ thống khóa thông minh này, ta sử dụng module RFID RC522 (thẻ MIFARE) – một đầu đọc RFID phổ biến cho vi điều khiển. RC522 giao tiếp với ESP32 qua giao thức SPI, cung cấp khả năng đọc mã UID (Unique ID) 4 hoặc 7 byte từ các thẻ MIFARE Classic. Nguyên lý hoạt động: khi người dùng áp thẻ lên đầu đọc, ESP32 sẽ nhận được mã UID từ RC522, sau đó so sánh với danh sách UID hợp lệ đã được lập trình sẵn. Nếu UID trùng khớp với thẻ được cấp quyền, hệ thống sẽ kích hoạt mở khóa; nếu không, khóa giữ nguyên trạng thái đóng và có thể báo hiệu (đèn LED đỏ, còi) để cảnh báo truy cập thất bại.

RFID mang lại ưu điểm là nhanh và thuận tiện – chỉ cần đưa thẻ gần đầu đọc trong tích tắc mà không cần tiếp xúc vật lý hay thao tác phức tạp. Tuy nhiên, thẻ RFID đơn thuần có thể bị chia sẻ hoặc đánh cắp, do đó tùy mức độ an ninh yêu cầu, hệ thống có thể kết hợp thêm mã PIN hoặc phương thức khác để nâng cao bảo mật.

2.4. Kết nối Wi-Fi và Bluetooth trong hệ thống

Hệ thống khóa thông minh thường tận dụng cả Wi-Fi và Bluetooth – hai chuẩn kết nối không dây phổ biến – để tối ưu trải nghiệm người dùng:

- Wi-Fi: Cho phép thiết bị kết nối Internet băng thông cao, phạm vi rộng. Với ESP32, kết nối Wi-Fi được dùng để tương tác từ xa thông qua dịch vụ đám mây (ví dụ Blynk). Ưu điểm của Wi-Fi là người dùng có thể điều khiển khóa từ bất cứ đâu có Internet – ví dụ, mở khóa cho người thân vào nhà dù đang ở cơ quan. Wi-Fi cũng cho phép gửi thông báo thời gian thực (qua Blynk hoặc MQTT) về tình trạng cửa. Tuy nhiên, Wi-Fi có nhược điểm là tiêu thụ năng lượng lớn hơn và phụ thuộc hạ tầng mạng (router Internet). Nếu mất kết nối Wi-Fi, chức năng điều khiển/giám sát từ xa sẽ tạm thời không hoạt động
- Bluetooth (BLE): ESP32 hỗ trợ Bluetooth Low Energy – kết nối không dây tầm gần (vài mét) với ưu thế tiết kiệm năng lượng hơn Wi-Fi. Trong phạm vi hệ thống khóa, Bluetooth có thể dùng để mở khóa khi người dùng ở gần cửa, ví dụ qua ứng dụng trên smartphone kết nối trực tiếp đến ESP32 (không qua Internet). Bluetooth cho trải nghiệm mở khóa nhanh tại chỗ và vẫn hoạt động ngay cả khi mất mạng Wi-Fi, đồng thời tiêu thụ ít điện nên phù hợp nếu khóa dùng pin. Nhược điểm của BLE là phạm vi giới hạn (~10 mét trở xuống) – không thể điều khiển khi ở xa – và tốc độ truyền thấp hơn. Do đó, BLE thích hợp làm giải pháp bổ sung: chẳng hạn khóa có thể tự động mở khi nhận thấy điện thoại (BLE) của chủ nhà ở ngay trước cửa.

Tóm lại, Wi-Fi và Bluetooth mang đến hai phương thức kết nối bổ trợ cho nhau. Trong thiết kế này, Wi-Fi trên ESP32 được sử dụng cho chức năng điều khiển/giám sát từ xa qua nền tảng Blynk (cloud), còn Bluetooth có thể được tích hợp để thử nghiệm tính năng mở khóa tầm gần (thông qua ứng dụng di động hoặc key fob BLE) mà không cần qua Internet. Sự kết hợp này đảm bảo hệ thống linh hoạt: điều khiển được cả khi ở xa lẫn khi ở gần, và vẫn hoạt động cơ bản kể cả khi một phương thức kết nối gặp sự cố.

2.6. Nền tảng IoT Blynk

Hình 2.5. Minh họa ứng dụng Blynk

(Nguồn: https://nshopvn.com/blog/huong-dan-cai-dat-va-su-dung-blynk-new-2-0-tren-arduino-ide-voi-esp8266)

Để xây dựng chức năng quản lý và điều khiển khóa từ xa, đề tài lựa chọn nền tảng Blynk – một giải pháp IoT dựa trên đám mây cho phép điều khiển thiết bị qua ứng dụng di động một cách nhanh chóng. Blynk.io cung cấp hạ tầng gồm máy chủ đám mây, ứng dụng di động và thư viện client cho vi điều khiển. Mục tiêu của Blynk là giúp người dùng IoT có thể dễ dàng tạo giao diện giám sát/điều khiển mà không cần tự phát triển app từ đầu. Cụ thể, nền tảng Blynk có 3 thành phần chính:

- Ứng dụng Blynk (Blynk App): Ứng dụng trên Android/iOS cho phép người dùng tạo giao diện điều khiển tùy biến cho dự án IoT của mình bằng cách kéo thả các widget (nút bấm, slider, đồ thị, đèn LED ảo, hiển thị giá trị cảm biến...). Trong dự án này, ta có thể thiết kế giao diện gồm nút bật/tắt khóa, đèn báo trạng thái khóa, và hiển thị nhật ký hoặc cảm biến.
- Máy chủ Blynk (Blynk Server): Đây là thành phần trung gian điều phối liên lạc giữa ứng dụng và thiết bị phần cứng. Blynk cung cấp máy chủ đám mây miễn phí (blynk.cloud) giúp lưu trữ dự án và truyền tải lệnh/ dữ liệu. Khi người dùng nhấn nút trên app, tín hiệu sẽ gửi lên server rồi chuyển tiếp đến vi điều khiển, và ngược lại thiết bị có thể gửi dữ liệu (như trạng thái cửa) lên server để hiển thị trên app. Blynk cho phép kết nối nhiều thiết bị

và quản lý chúng thông qua tài khoản đám mây.

- Thư viện Blynk (Blynk Libraries): Phía vi điều khiển (ESP32) sẽ sử dụng thư viện Blynk để kết nối tới server và xử lý lệnh. Lập trình viên chỉ cần sử dụng API đơn giản để kết nối Wi-Fi và đăng ký vào server Blynk, để xử lý khi có dữ liệu từ app gửi xuống, hoặc để gửi dữ liệu lên app. Trong chương trình mẫu, khi thẻ RFID hợp lệ được quét, ESP32 có thể gọi để cập nhật widget (ví dụ LED ảo báo "Đã mở khóa") trên giao diện app. Ngược lại, người dùng nhấn nút mở khóa trên app sẽ kích hoạt trên ESP32 để thực hiện mở khóa.

Blynk mang lại sự thuận tiện vượt trội: thay vì phải tự viết ứng dụng điện thoại, người phát triển chỉ cần tập trung lập trình logic thiết bị. Toàn bộ hạ tầng server và app đã sẵn sàng, giúp triển khai nhanh chức năng IoT cho hệ thống. Đối với khóa thông minh, Blynk cho phép: Mở khóa từ xa (nhấn nút trên app để mở khóa qua Wi-Fi), Giám sát trạng thái cửa (đang khóa hay mở, lịch sử mở khóa), và thậm chí thiết lập cảnh báo (thông báo đẩy nếu cửa mở trái phép). Blynk hỗ trợ điều khiển thời gian thực gần như tức thì – tên gọi Blynk xuất phát từ "blink of an eye" (trong nháy mắt) ám chỉ tốc độ phản hồi nhanh

Trong đề tài, ta sẽ sử dụng Blynk Cloud để tránh triển khai server cục bộ, và dùng phiên bản Blynk mới nhất (Blynk IoT) với Template và Device cung cấp mã BLYNK_TEMPLATE_ID, BLYNK_AUTH_TOKEN để cấu hình thiết bị. Ứng dụng Blynk sẽ có giao diện đơn giản gồm nút nhấn ảo điều khiển khóa (Virtual Pin V2 chẳng hạn) và đèn LED ảo hiển thị trạng thái khóa (V0, V1), cùng các widget khác nếu cần. ESP32 khi chạy sẽ kết nối Wi-Fi rồi kết nối đến server Blynk bằng token đã cấp. Qua đó, mọi thao tác từ người dùng trên app đều được chuyển đến ESP32, giúp tương tác với hệ thống khóa mọi lúc mọi nơi.

2.7. Mô phỏng trên Wokwi

Hình 2.6. Minh họa trang web mô phỏng Wokwi

(Nguồn: https://wokwi.com)

Wokwi là một môi trường thuận lợi cho việc thử nghiệm IoT ảo. Người dùng có thể "kết nối" các cảm biến và module với ESP32 bằng cách vẽ dây nối trên sơ đồ, sau đó chạy chương trình để quan sát tương tác. Các cảm biến phụ trợ trong mô hình này chủ yếu nhằm mô phỏng tín hiệu đầu vào (ví dụ phát hiện cửa đang mở, có người đến gần) để hệ thống phản ứng (gửi cảnh báo, bật đèn LED, v.v.). Điều này giúp kiểm chứng rằng thiết kế khóa thông minh có thể mở rộng để tích hợp vào hệ sinh thái nhà thông minh toàn diện với nhiều loại cảm biến.

Wokwi cung cấp nhiều linh kiện ảo có thể thêm vào sơ đồ: từ LED, LCD, cảm biến DHT22 (nhiệt độ/độ ẩm), cảm biến PIR, nút nhấn, biến trở... Người phát triển có thể tùy ý cấu hình những cảm biến này trong file cấu hình của dự án Wokwi. Nếu một cảm biến chưa được Wokwi hỗ trợ trực tiếp (ví dụ module RC522 RFID chưa có mô hình sẵn), ta có thể giả lập gián tiếp – như với RFID, có thể dùng Serial giả lập để gửi mã thẻ vào ESP32 thay cho đầu đọc thực. Nhờ vậy, hầu hết kịch bản hoạt động của hệ thống đều có thể thử nghiệm ngay trên trình duyệt mà không cần phần cứng thật.

CHƯƠNG 3: THIẾT KẾ VÀ MÔ PHỎNG HỆ THỐNG

3.1. Sơ đồ khối tổng thể

Hệ thống khóa cửa thông minh dựa trên ESP32 và RFID được thiết kế theo dạng các khối chức năng chính và cách chúng tương tác với nhau. Sơ đồ khối tổng thể như Hình 3.1 mô tả các thành phần và luồng tín hiệu chính trong hệ thống:

Hình 3.1. Các thành phần chính của hệ thống khóa cửa thông minh. Bộ Lock (bên trái) bao gồm ESP32, đầu đọc RFID RC522, khóa điện từ (hoặc động cơ chốt cửa) và module relay điều khiển khóa. Bộ Key (bên phải) là các thẻ RFID dùng làm "chìa khóa" mở cửa.

(Nguồn: https://esp32io.com/tutorials/esp32-rfid-nfc-door-lock-system#:~:text=A%20door%20lock%20system%20includes,two%20main%20parts)

Trong đó, có thể phân chia thành hai phần chính:

- (1) Phần khóa (Door Lock) bao gồm ESP32 làm bộ điều khiển trung tâm, kết nối với đầu đọc RFID RC522 để nhận tín hiệu thẻ, và điều khiển khóa điện (solenoid hoặc strike) thôngÃ qua một rơ-le. Ngoài ra, ESP32 còn kết nối Wi-Fi để giao tiếp với đám mây Blynk và có thể nối với các cảm biến phụ (PIR, cửa,…) nếu có.
- (2) Phần chìa khóa (Door Key) là các thẻ RFID (dạng thẻ từ hoặc móc khóa tag) mang mã định danh, được phát cho những người được phép ra vào. Khi hệ thống hoạt động, hai phần này tương tác qua sóng RFID: người dùng chạm thẻ (Key) vào đầu đọc, hệ thống khóa (Lock) sẽ xác thực và thực hiện hành động mở cửa nếu hợp lệ.

3.2. Mô tả chức năng từng phần tử

Dựa trên sơ đồ khối, ta mô tả chức năng của từng phần tử chính như sau:

- ESP32: Là bộ xử lý trung tâm của hệ thống. Nó lưu trữ chương trình điều khiển với các chức năng: đọc dữ liệu từ đầu đọc RFID; so sánh mã thẻ với danh sách hợp lệ; điều khiển chốt khóa (mở hoặc khóa) bằng cách xuất tín hiệu tới rơ-le; quản lý kết nối Wi-Fi và giao tiếp với server Blynk để nhận lệnh mở khóa từ xa hoặc gửi trạng thái. ESP32 cũng xử lý tín hiệu từ các cảm biến khác (nếu có) và thực hiện các hành động tương ứng (ví dụ, phát còi báo động nếu cảm biến cửa báo cửa bị mở khi chưa được phép).
- Đầu đọc RFID RC522: Thiết bị này đảm nhận việc đọc mã RFID. Khi một thẻ RFID được đưa vào vùng gần, RC522 sẽ thu tín hiệu và trích xuất UID của thẻ. Module RC522 giao tiếp với ESP32 qua giao diện SPI (các chân MOSI, MISO, SCK, SDA/SS và dùng nguồn 3.3V). Vai trò của RC522 là cầu nối giữa thẻ RFID và vi điều khiển, giúp chuyển thông tin ID thẻ thành dữ liệu số cho ESP32.

Hình 3.2. Minh họa Đầu đọc RFID RC522

(Nguồn: https://linhkienthuduc.com/wp-content/uploads/2024/09/kiotviet_799da750f59df6a586705757d0be10d2.jpg)

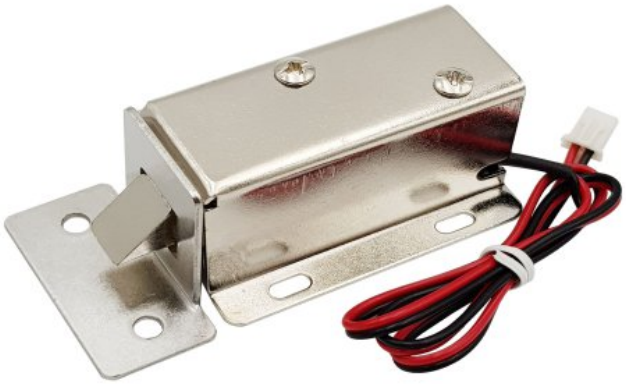
- Thẻ RFID: Mỗi thẻ chứa một mã ID duy nhất. Trong hệ thống, một số thẻ được đăng ký là hợp lệ (có quyền mở khóa). Khi thẻ hợp lệ được quét, thẻ đóng vai trò như "chìa khóa" điện tử: gửi mã ID để hệ thống nhận dạng và

cấp phép mở cửa. Nếu thẻ không có trong danh sách, hệ thống coi đó là truy cập không hợp lệ.

Hình 3.3. Minh họa một số loại thẻ RFID trên thị trường hiện nay

(Nguồn: https://ezonetech.com.vn/the-rfid/)

- Khóa điện: Đây là chốt cửa điều khiển bằng điện. Thông thường, khóa điện từ cần nguồn 12V và được đóng/ngắt thông qua một module rơ-le kết nối với ESP32. Khi rơ-le đóng mạch, dòng điện 12V chạy qua cuộn hút của khóa điện sẽ mở chốt cửa (hoặc nhả chốt tùy loại), cho phép cửa mở. Khi rơ-le ngắt, khóa trở lại trạng thái khóa bình thường (chốt cửa đóng). Trong mô hình giả lập, có thể thay khóa điện bằng động cơ servo để biểu diễn trạng thái khóa/mở bằng góc quay.

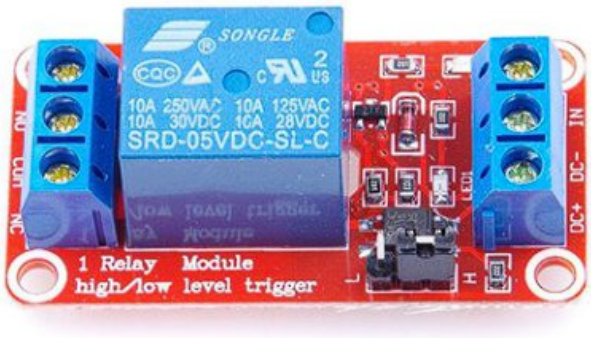


Hình 3.4. Minh họa khóa điện từ Solenoid

(Nguồn: https://nshopvn.com/wp-content/uploads/2019/03/khoa-chot-dien-tu-ly-03-24vdc-gsg6-1-1-scaled.jpg)

- Module Rơ-le (Relay): Rơ-le đóng vai trò công tắc đóng cắt mạch điện cho

khóa. Do ESP32 xuất tín hiệu 3.3V không thể điều khiển trực tiếp khóa 12V, rơ-le sẽ cách ly và dùng tín hiệu ESP32 (IN) để đóng/ngắt mạch nguồn 12V cấp cho khóa điện. Khi ESP32 gửi tín hiệu mở khóa (ví dụ đặt chân GPIO điều khiển lên mức HIGH), rơ-le sẽ tác động cho dòng chạy qua khóa → mở chốt. Rơ-le cũng đảm bảo an toàn khi cách ly phần điện áp cao và vi điều khiển.



Hình 3.5. Minh họa Module Rơ-le (Relay)

(Nguồn: https://dientu360.com/module-relay-1-kenh-5v-220vac10a-cach-ly-quang)

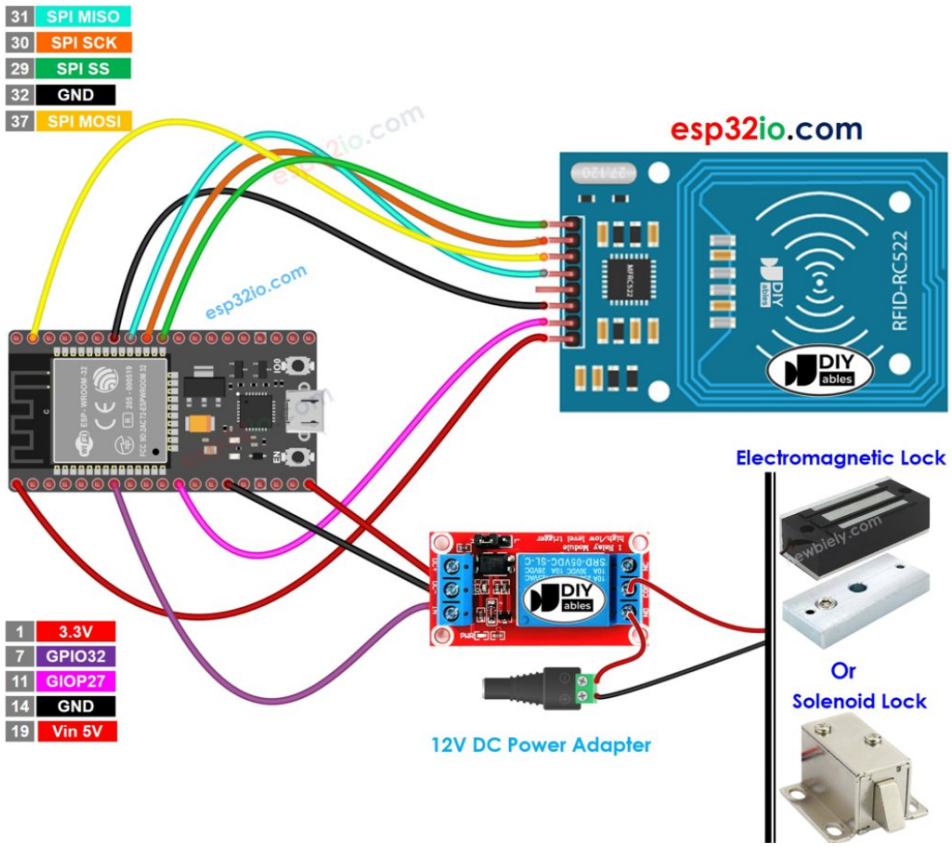
- Nguồn cấp: Hệ thống cần nguồn 3.3V cho ESP32 và RC522, đồng thời nguồn 5V hoặc 12V cho khóa điện (tùy loại) và rơ-le. Trong thực tế, có thể dùng adapter 12V rồi qua module regulator (7805 hay DC-DC) để tạo ra 5V, 3.3V tương ứng. Trên Wokwi, nguồn được giả lập thông qua cấu hình.
- Ứng dụng Blynk (đám mây): Không trực tiếp nằm trong sơ đồ phần cứng nhưng là thành phần không thể thiếu về mặt hệ thống. Ứng dụng Blynk trên điện thoại kết nối đến server đám mây, từ đó liên kết tới ESP32. Chức năng: gửi lệnh mở khóa từ xa khi người dùng bấm nút, hiển thị trạng thái khóa (đang mở/đóng), và có thể lưu log sự kiện (thời điểm mở khóa, ai mở). Blynk giúp mở rộng phạm vi kiểm soát của hệ thống ra ngoài phạm vi vật lý của thẻ RFID, tạo nên kênh điều khiển thứ hai song song với RFID.

Nhìn chung, các phần tử trên kết hợp tạo thành một quy trình hoàn chỉnh: Xác thực người dùng (qua thẻ RFID hoặc app) → Quyết định cho phép truy cập (ESP32 kiểm tra hợp lệ) → Tác động cơ cấu khóa (kích rơ-le mở chốt) → Phản hồi người dùng

(đèn LED, LCD hiển thị, thông báo trên app). Chức năng từng phần tử liên kết chặt chẽ đảm bảo hệ thống hoạt động nhịp nhàng và an toàn.

3.3. Thiết kế sơ đồ điện và luồng dữ liệu

Để triển khai các chức năng trên, ta tiến hành thiết kế sơ đồ mạch điện chi tiết của hệ thống. Hình 3.2 trình bày sơ đồ nguyên lý kết nối giữa ESP32 và các thành phần ngoại vi (RC522, rơ-le, servo/khóa điện, LED, buzzer, cảm biến):



Hình 3.2. Sơ đồ nối dây của hệ thống khóa cửa thông minh dùng ESP32 và RFID (mô phỏng trên Wokwi). Đầu đọc RFID RC522 kết nối ESP32 qua giao tiếp SPI (các chân MOSI, MISO, SCK và SS). Module rơ-le được nối với một chân GPIO (ví dụ GPIO27) của ESP32 để điều khiển khóa điện 12V (nguồn 12V cấp qua rơ-le tới khóa). Ở đây minh họa khóa điện có thể là khóa điện từ hoặc solenoid (hình ảnh bên phải). ESP32 kết nối Wi-Fi để giao tiếp Blynk (không thể hiện trên sơ đồ phần cứng).

(Nguồn: https://esp32io.com/tutorials/esp32-rfid-nfc-door-lock-system#:~:text=A%20door%20lock%20system%20includes,two%20main%20parts)

Trên sơ đồ (Hình 3.2), ta thấy các mối nối chính:

- RFID RC522: các chân SPI của RC522 (MISO, MOSI, SCK, SS) được kết nối lần lượt tới các chân phần cứng SPI của ESP32 (theo minh họa: MISO

→ GPIO19, MOSI → GPIO23, SCK → GPIO18, SS → GPIO5). Chân RST của RC522 nối với một GPIO tự do (ví dụ GPIO22) để ESP32 có thể reset module khi cần. Nguồn cấp 3.3V và GND của RC522 nối với 3.3V và GND tương ứng trên ESP32. Kết nối này đảm bảo ESP32 có thể đọc UID thẻ từ RC522.

- Module Rơ-le: dùng loại rơ-le đóng cắt mức thấp 5V. Chân tín hiệu IN của rơ-le nối với một chân GPIO (ví dụ GPIO27) của ESP32. Chân VCC rơ-le nối 5V (hoặc 3.3V tùy module), GND nối GND chung. Mạch khóa điện: một đầu khóa điện nối với nguồn 12V, đầu kia nối với chân COM của rơ-le. Chân NO (Normally Open) của rơ-le nối về GND. Khi ESP32 đưa GPIO27 lên HIGH, rơ-le đóng, nối COM với NO (tức nối khóa điện với GND) tạo thành mạch kín 12V qua khóa → khóa hoạt động (mở chốt). Khi GPIO27 LOW, mạch hở, khóa mất nguồn → khóa chốt lại.
- Động cơ Servo (giả lập chốt khóa): Trong mô hình Wokwi có thể dùng servo nhỏ (SG90) thay cho khóa để quan sát trực quan. Servo có 3 dây: VCC 5V, GND, và dây tín hiệu điều khiển PWM nối tới một chân PWM của ESP32 (ví dụ GPIO12). Chương trình sẽ điều khiển servo xoay góc 0° (hoặc một góc cố định) khi khóa, và 90° khi mở, tương ứng trạng thái chốt gài vào hoặc rút ra. (Lưu ý servo chỉ dùng trong giả lập minh họa, thực tế khóa điện sẽ do rơ-le điều khiển như trên).
- Buzzer và LED (cảnh báo): Ta có thể gắn thêm 1 còi buzzer và 2 đèn LED (xanh/đỏ) vào các chân ESP32 (ví dụ LED xanh GPIO4, LED đỏ GPIO5, buzzer GPIO21). LED xanh bật khi mở khóa thành công, LED đỏ bật khi thẻ sai. Buzzer có thể kêu khi có xâm nhập sai quá 3 lần. Các thành phần này tạo phản hồi âm thanh/hình ảnh trực tiếp tại chỗ cho người dùng.
- Cảm biến cửa (reed switch): nếu sử dụng, sẽ nối giữa một chân GPIO và GND, ở trạng thái thường đóng hoặc thường hở tùy cấu hình trên cửa. Khi cửa mở, trạng thái chân sẽ thay đổi (từ HIGH xuống LOW hoặc ngược lại), ESP32 đọc được và biết cửa đang mở. Kết hợp thông tin này với trạng thái khóa (mở hay đóng) để cảnh báo nếu cửa bị mở mà khóa chưa mở (có thể do cạy cửa).

- Kết nối Wi-Fi: không thể hiện bằng dây, nhưng ESP32 sử dụng module Wi-Fi nội tại để kết nối router. Thông tin SSID, password và Blynk Auth Token được cấu hình trong code. Khi chạy, ESP32 sẽ thu phát dữ liệu qua Wi-Fi module – đây là quá trình logic (đèn Wi-Fi trên ESP32 DevKit sẽ nhấp nháy khi truyền dữ liệu).

Luồng dữ liệu trong hệ thống diễn ra theo các kịch bản chính sau:

- Mở khóa bằng thẻ RFID: Khi người dùng quẹt thẻ vào đầu đọc, RC522 sẽ thu thập mã UID của thẻ và gửi qua giao tiếp SPI vào ESP32. Chương trình trên ESP32 nhận UID, so sánh với danh sách UID hợp lệ (lưu trong bộ nhớ flash hoặc mã nguồn). Nếu khớp, ESP32 lập tức kích hoạt rơ-le (hoặc servo) để mở khóa cửa trong một khoảng thời gian ngắn (vài giây) rồi khóa lại. Đồng thời, ESP32 có thể gửi tín hiệu lên Blynk (virtual pin) để thông báo "Đã mở bằng thẻ ID: XYZ". Nếu UID không hợp lệ, ESP32 sẽ không mở khóa, bật LED đỏ/còi, và cũng có thể gửi cảnh báo lên Blynk về thẻ không hợp lệ.
- Mở khóa từ xa qua Blynk: Khi chủ nhà dùng ứng dụng Blynk nhấn nút "Unlock" (mở khóa) trên điện thoại, ứng dụng sẽ gửi lệnh tới Blynk server, rồi server chuyển tiếp lệnh xuống thiết bị ESP32 (nếu đang trực tuyến). Thư viện Blynk trên ESP32 nhận lệnh qua cơ chế BLYNK_WRITE gắn với virtual pin tương ứng. Lúc này, ESP32 xác nhận yêu cầu hợp lệ (đúng Auth Token) và thực thi việc mở khóa như trên (kích rơ-le mở khóa vài giây). Trạng thái mới của khóa cũng được ESP32 gửi ngược lên server để app cập nhật (ví dụ đổi biểu tượng ổ khóa sang mở). Sau khi hết thời gian mở, ESP32 khóa lại và cập nhật trạng thái.
- Giám sát và báo cáo: Định kỳ (ví dụ mỗi 1 giây), ESP32 có thể gửi trạng thái khóa (đang khóa hay mở) và trạng thái cảm biến (cửa đóng/mở, phát hiện chuyển động, nhiệt độ, v.v.) lên Blynk. Nhờ đó, người dùng mở app sẽ xem được tình trạng hiện tại. Nếu có bất thường (như cửa mở bất thường), ESP32 có thể kích hoạt còi tại chỗ và gửi thông báo đẩy (Notification) qua Blynk đến điện thoại người dùng. Luồng dữ liệu này đảm bảo hệ thống phản ứng kịp thời và người dùng ở xa vẫn nắm bắt được

diễn biến ở nhà.

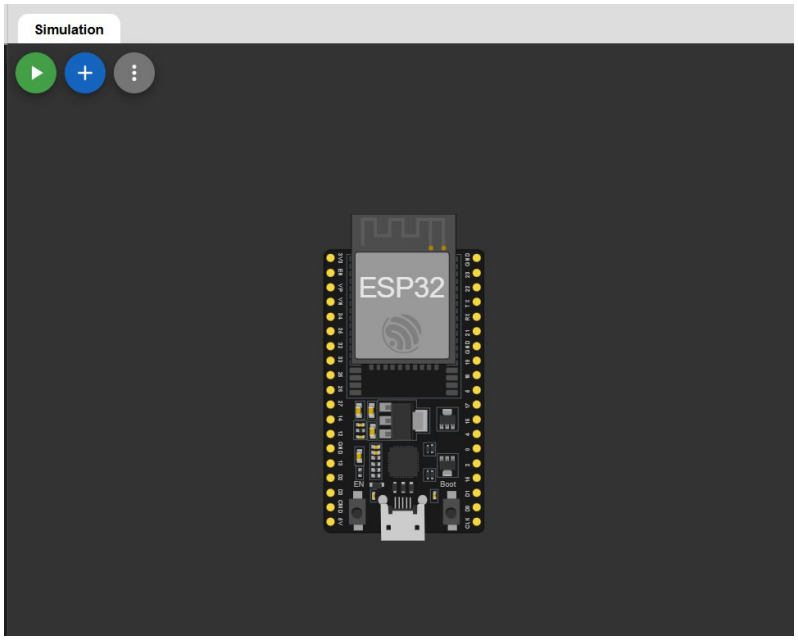
- Cập nhật danh sách thẻ: Nếu cần thêm/xóa thẻ hợp lệ, có thể thực hiện bằng cách cập nhật firmware (trong mô hình đơn giản). Ở mức độ cao hơn, có thể làm một giao diện web hoặc app để thêm người dùng mới, sau đó ESP32 đồng bộ danh sách (yêu cầu phương thức lưu trữ phức tạp hơn, ngoài phạm vi đề tài).

Như vậy, sơ đồ điện và luồng dữ liệu kết hợp đã cho thấy cách các thành phần tương tác và dòng thông tin di chuyển trong hệ thống. Mọi quyết định thông minh đều tập trung ở ESP32 (bộ não), trong khi RFID và Blynk đóng vai trò giác quan nhận diện người dùng ở gần và ở xa.

3.4. Mô phỏng hệ thống trên Wokwi

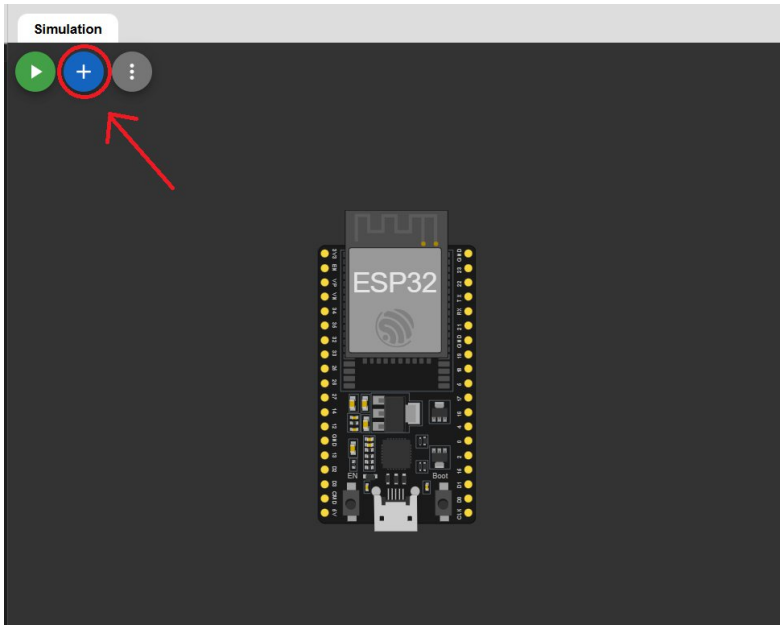
Sau khi hoàn thiện thiết kế, bước tiếp theo là mô phỏng hệ thống trên Wokwi để kiểm chứng hoạt động. Wokwi là trình mô phỏng trực tuyến mạnh mẽ hỗ trợ ESP32 và nhiều linh kiện, cho phép ta chạy thử chương trình như trên phần cứng thực. Quy trình mô phỏng trên Wokwi bao gồm các bước chính sau:

- **Bước 1**: Chuẩn bị dự án Wokwi: Truy cập trang Wokwi.com và tạo một Project mới. Chọn loại board là ESP32 DevKit v1 (loại thông dụng). Giao diện Wokwi sẽ hiện một sơ đồ với board ESP32 và có thể thêm các linh kiện cần thiết từ thư viện.



Hình 3.3. Minh họa bước 1 tạo sơ đồ với board ESP32

- **Bước 2**: Thêm linh kiện vào sơ đồ: Sử dụng Diagram Editor của Wokwi để tìm và thêm các phần: RC522 RFID module, LED, buzzer, servo, LCD,... tùy nhu cầu mô phỏng. Ta sắp xếp các linh kiện xung quanh ESP32 và vẽ dây nối tương tự sơ đồ mạch đã thiết kế (Hình 3.2). Ví dụ, nối các chân MISO, MOSI... như thiết kế, nối servo vào GPIO12, LED vào GPIO4,5, v.v. Wokwi cho phép đổi màu dây và gắn nhãn để tiện theo dõi.

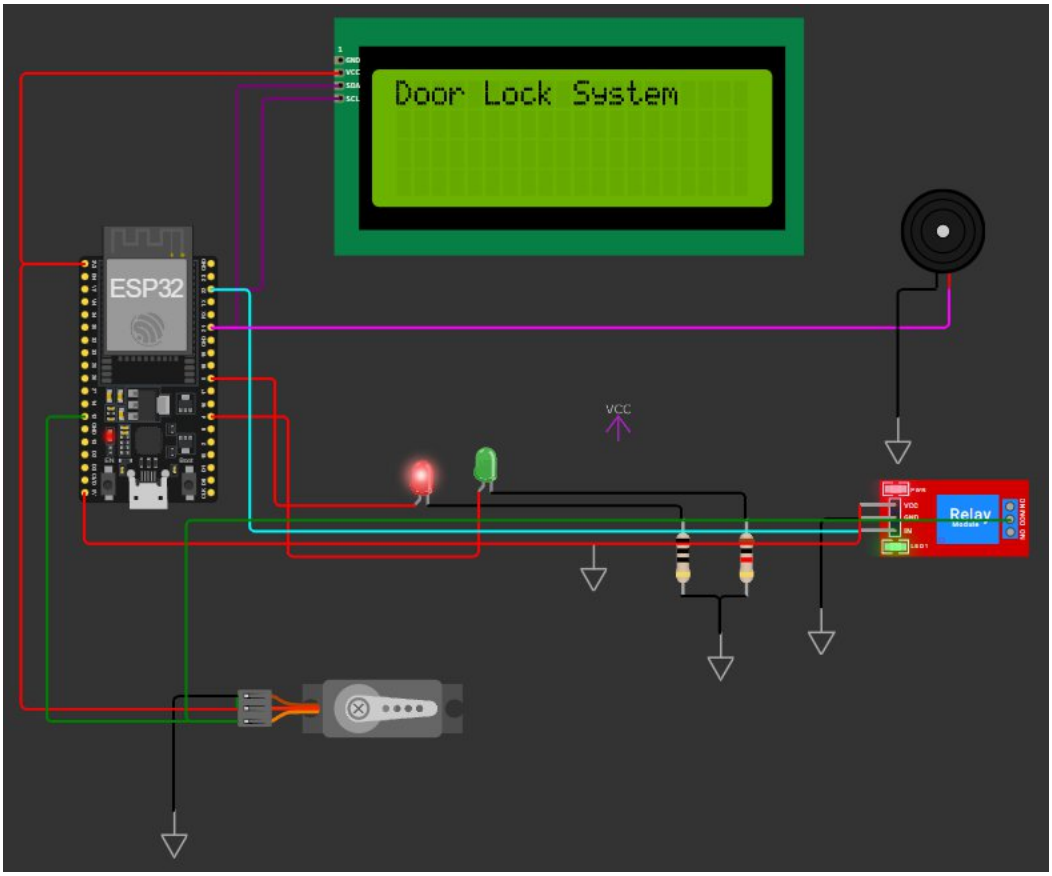


Hình 3.4. Minh họa bước thêm linh kiện vào sơ đồ (ở trong hình, ta sẽ tiến hành nhấp vào dấu cộng + để thêm linh kiện phù hợp trong quá trình thiết lập)

- **Bước 3**: Thiết lập code chương trình: Trong phần code, nhập hoặc phát triển mã điều khiển cho ESP32. Code sẽ bao gồm: cấu hình Wi-Fi và Blynk (với Auth token), thiết lập các chân, khai báo hàm đọc, hàm điều khiển khóa (kích servo/rơ-le), hàm xử lý Blynk, và vòng loop. Do mô phỏng RC522 không có sẵn, một kỹ thuật thường dùng là chạy vòng lặp đọc Serial để bắt chuỗi ký tự nhập vào, coi đó như UID thẻ. Trên Wokwi, ta có thể mở Serial Monitor và nhập một chuỗi (ví dụ "E280689401A9") để giả lập quét một thẻ có UID đó.
- **Bước 4**: Cấu hình mạng và Blynk trên Wokwi: Wokwi có hỗ trợ mô phỏng kết nối Wi-Fi tới Internet (thông qua mạng máy tính). ESP32 sẽ kết nối được đến server Blynk. Trước đó, trên ứng dụng Blynk (hoặc web Blynk Console), ta đã tạo một Template cho "Smart Lock" và một Device để lấy mã BLYNK_TEMPLATE_ID và BLYNK_AUTH_TOKEN điền vào

code. Khi chạy mô phỏng, nếu kết nối thành công, ta có thể điều khiển nút trên app Blynk thật và thấy tác động trong Wokwi (vd servo quay).

- **Bước 5**: Chạy mô phỏng và quan sát: Nhấn nút Run (hình tam giác) để bắt đầu mô phỏng. Ta nên mở Serial Monitor để xem các log in ra (VD: "Welcome to Your Home, Please scan your RFID Card"). Giờ, thử các tình huống: nhập một UID hợp lệ trong Serial Monitor → quan sát servo quay mở khóa, LED xanh sáng, Serial in thông báo chào mừng tên thẻ, đồng thời trên app Blynk thấy đèn trạng thái chuyển màu báo cửa mở. Sau vài giây, servo quay lại khóa, LED tắt. Tiếp theo, nhập một UID không hợp lệ → hệ thống báo đèn đỏ, còi kêu, không mở khóa. Thử nhấn nút Unlock trên app Blynk → trên Serial Monitor thấy lệnh nhận, servo quay mở khóa (dù ta không nhập gì Serial). Điều này chứng tỏ kênh Wi-Fi/Blynk hoạt động song song kênh RFID.



Hình 3.5. Minh họa sơ đồ sau khi hoàn thành quá trình thiết lập trên mô phỏng Wokwi

Nhờ mô phỏng Wokwi, ta có thể sửa lỗi và tối ưu chương trình ngay trên máy tính. Nếu logic sai, có thể debug, in log. Wokwi thậm chí hỗ trợ giả lập debug (GDB) cho ESP32 nhưng trong phạm vi đề tài ta có thể không cần dùng đến. Quan trọng là

kết quả mô phỏng khẳng định hệ thống làm đúng như thiết kế mong muốn trước khi tiến hành xây dựng thực tế.

3.5. Trình bày tích hợp Blynk vào mô phỏng

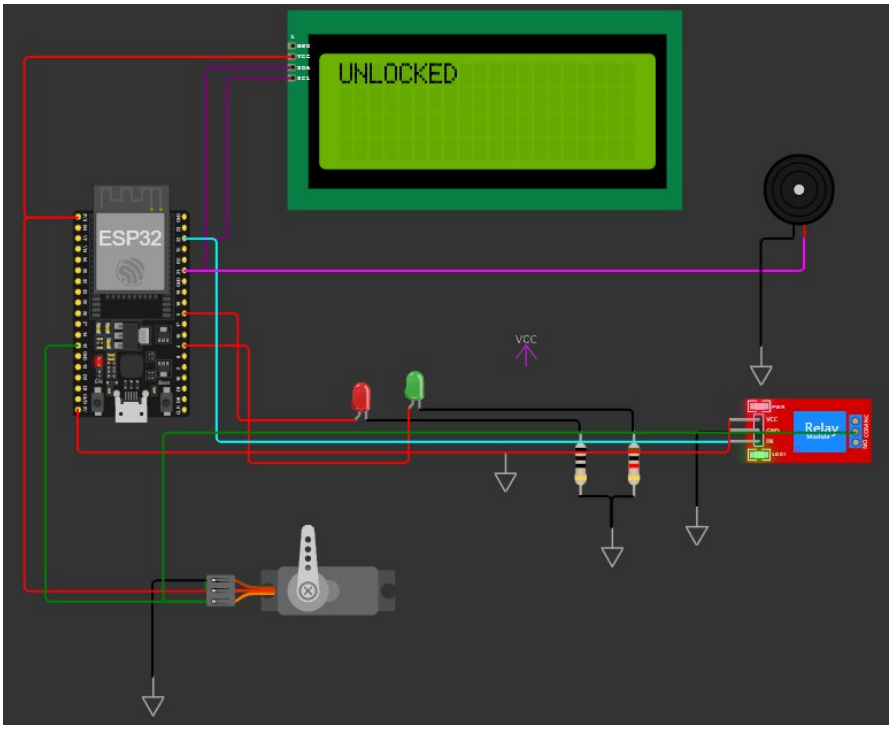
Việc tích hợp Blynk vào mô hình mô phỏng thực chất đã được thực hiện trong quá trình viết code và chạy thử ở bước trên, nhưng ta sẽ trình bày rõ hơn cách cấu hình và tương tác với Blynk:

Trước tiên, trên nền tảng Blynk (sử dụng phiên bản mới – Blynk IoT), ta tạo một template dự án với tên ví dụ “Smart RFID Lock”. Trong template định nghĩa các datastream (luồng dữ liệu) tương ứng với các Virtual Pin mà ta dùng trong code. Ví dụ: V0 cho trạng thái khóa (0/1 tương ứng đóng/mở), V1 cho đèn LED trạng thái, V2 cho nút điều khiển mở khóa. Sau đó tạo một device từ template để lấy mã Auth Token (mã xác thực duy nhất thiết bị dùng để kết nối đến Blynk cloud).

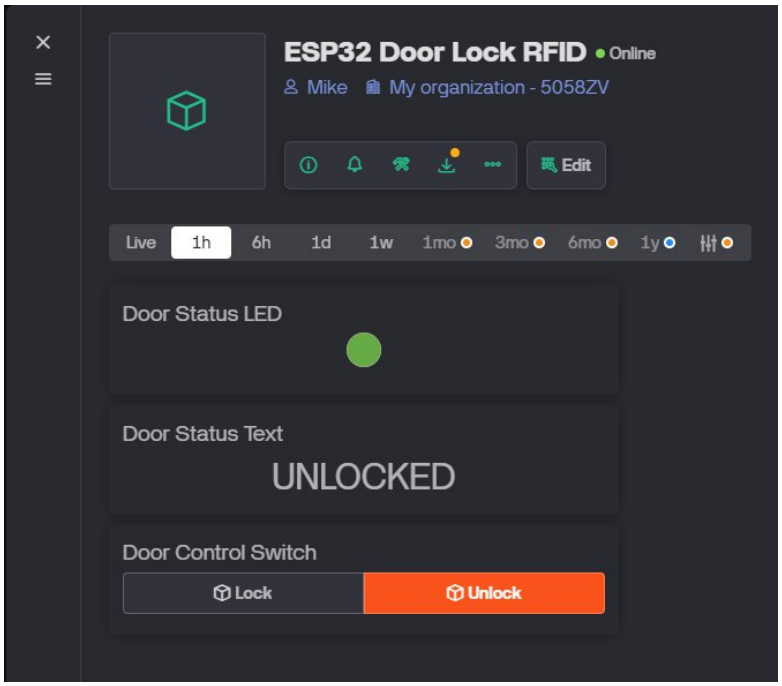
Sau khi tích hợp, khi chạy mô phỏng và thiết bị báo "Device connected", trên app Blynk ta sẽ thấy thiết bị online. Bây giờ, thử nhấn nút trên app: ESP32 nhận được lệnh và thực thi kích hoạt servo hoặc rơ-le. Đồng thời, ESP32 gửi lại để đặt trạng thái khóa là mở, ứng dụng sẽ phản ánh điều này bằng cách đổi màu icon hoặc hiển thị "Unlocked". Khi khóa lại, ESP32 gửi để đặt trạng thái khóa là khóa, hiển thị “Locked”

Tất cả những tích hợp trên đều mô phỏng tốt trong Wokwi vì Wokwi hỗ trợ ESP32 kết nối internet thật. Do đó, dự án mô phỏng không chỉ kiểm thử phần nhúng mà còn kiểm thử luôn tính năng IoT cloud của hệ thống. Kết quả là ta có một mô hình hoàn chỉnh có thể tương tác từ app điện thoại thật, đem lại trải nghiệm rất gần với hệ thống thực tế.

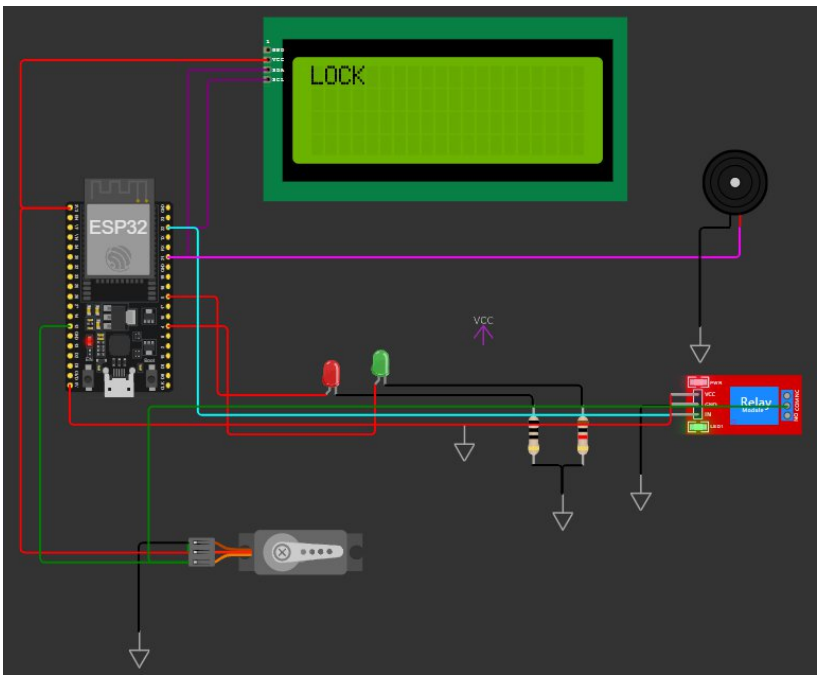
3.6. Kết quả của quá trình mô phỏng và thiết lập trên Blynk



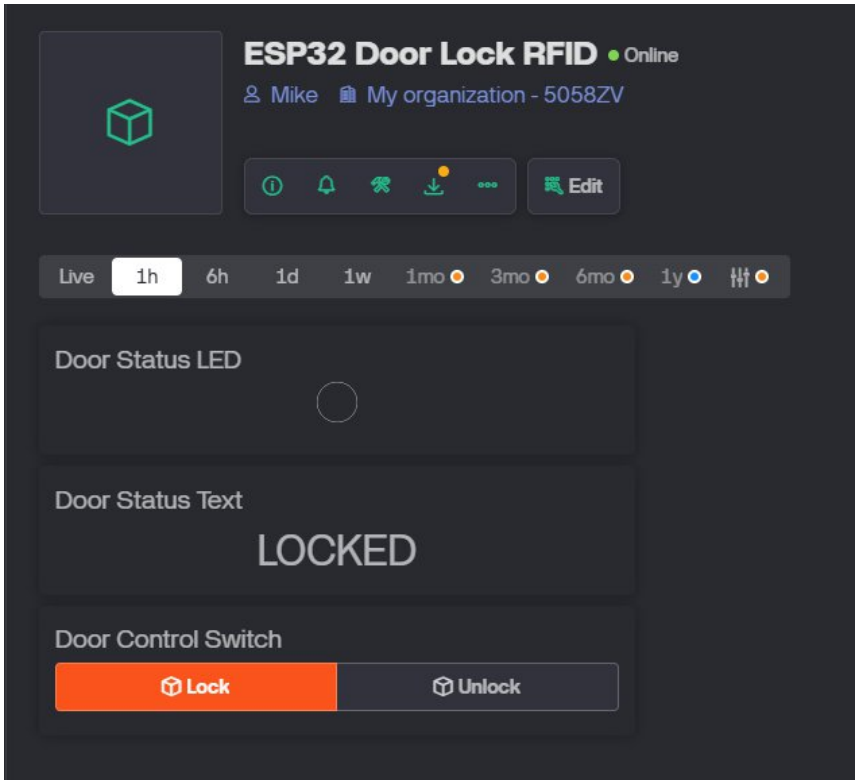
Hình 3.6. Minh họa kết quả khi chạy trên chương trình mô phỏng Wokwi – Chế độ mở khóa (Unlocked)



Hình 3.7. Minh họa kết quả khi chạy trên Blynk Cloud tương ứng với chế độ mở khóa (Unlocked) – Cấu hình như ở Hình 3.6



Hình 3.8. Minh họa kết quả khi chạy trên chương trình mô phỏng Wokwi – Chế độ khóa (Lock)



Hình 3.9. Minh họa kết quả khi chạy trên Blynk Cloud tương ứng với chế độ khóa (Locked) – Cấu hình như ở Hình 3.8

CHƯƠNG 4: PHÂN TÍCH VÀ THẢO LUẬN

4.1. Phân tích kết quả mô phỏng

Sau khi xây dựng và chạy mô phỏng, kết quả thu được cho thấy hệ thống hoạt động đúng như kỳ vọng thiết kế. Cụ thể:

Chức năng mở khóa bằng RFID: Mô phỏng cho thấy khi nhập chuỗi UID hợp lệ vào Serial (tương đương quét thẻ thật), ESP32 nhận diện đúng mã và kích hoạt mở khóa. Ta quan sát servo quay hoặc trạng thái rơ-le đổi (trên Wokwi có thể xem LED trạng thái rơ-le), nghĩa là chốt cửa mở. Đèn LED xanh ảo bật và thông báo Serial hiển thị "Welcome, Access Granted" cùng tên người dùng (nếu ta gán mỗi UID một tên) giống như dự kiến. Sau khoảng thời gian thiết lập (vd 5 giây), servo tự động quay về khóa, LED xanh tắt, đảm bảo cửa không mở quá lâu. Trường hợp nhập UID không hợp lệ, hệ thống từ chối: không kích hoạt servo/rơ-le, thay vào đó LED đỏ bật nháy và còi kêu báo hiệu. Trên Serial in ra thông báo "Access Denied" hoặc "Unrecognized card". Điều này chứng minh thuật toán xác thực RFID thực thi đúng: chỉ cho phép thẻ hợp lệ mở cửa.

Chức năng mở khóa qua ứng dụng Blynk: Trên điện thoại/web console, khi bấm nút "Unlock", ngay lập tức (độ trễ rất nhỏ, khoảng <1 giây) servo trên Wokwi quay mở khóa, cho thấy lệnh đã được truyền qua đám mây thành công. Đồng thời, app Blynk đổi trạng thái nút (ví dụ chuyển sang "Unlocked") và LED ảo sáng lên. Khi nhả nút (hoặc nếu dùng nút kiểu switch thì khi bật → mở, tắt → khóa), servo trở lại vị trí khóa. Kết quả này cho thấy kênh điều khiển từ xa vận hành hiệu quả – ESP32 duy trì kết nối tốt với server và xử lý đúng lệnh người dùng. Dù Wokwi là giả lập, nhưng việc tương tác với Blynk cloud là thật, chứng tỏ nếu triển khai phần cứng thật thì chức năng này hoàn toàn khả thi.

Đồng bộ hai phương thức: Một điểm đáng chú ý là hệ thống cho phép mở khóa bằng một phương thức và khóa lại bằng phương thức khác. Ví dụ: ta quét thẻ RFID mở cửa, cửa mở (servo quay). Sau đó, thay vì chờ tự động khóa, ta cũng có thể bấm nút "Lock" trên app Blynk để khóa cửa ngay – và servo lập tức phản hồi khóa lại. Ngược lại, có thể mở bằng app, rồi đóng bằng thẻ (giả lập quẹt thẻ "Đóng" – mặc dù thường thẻ không phân biệt mở/đóng, nhưng ta có thể thiết kế một thẻ master để khóa). Thử nghiệm cho thấy ESP32 xử lý các yêu cầu không xung đột này tốt: cơ chế

Blynk và RFID đều cuối cùng gọi chung các hàm unlockDoor() hoặc lockDoor(), do đó tình trạng khóa luôn được đảm bảo về một trạng thái nhất quán.

Giám sát trạng thái và nhật ký: Trên Serial monitor cũng như trên ứng dụng Blynk (phần Terminal hoặc History widget nếu có cấu hình), ta thấy ghi lại các sự kiện: "User A unlocked via RFID at 10:01:05", "Unlocked via App at 10:05:30 by Owner". Điều này đạt được nhờ code gửi các thông tin này lên Blynk (hoặc in ra Serial để giám sát trực tiếp). Nhật ký sự kiện giúp phân tích hành vi hệ thống: chẳng hạn tần suất mở khóa, thời điểm bất thường. Mô phỏng cho thấy không có sự kiện lỗi (ví dụ mở khóa khi không có lệnh, hay từ chối thẻ hợp lệ) – nếu có, đó là bug logic cần sửa. Qua các lần thử, hệ thống không mở nhầm khi thẻ sai, và cũng không bỏ sót khi thẻ đúng hoặc lệnh hợp lệ được gửi.

Hiệu suất và độ trễ: Trong mô phỏng, do không có độ trễ vật lý đáng kể, phản ứng của hệ thống rất nhanh. Quét thẻ → mở khóa gần như tức thì (dưới 0.2s). Lệnh app → khóa cũng nhanh (~1s). Quan sát tải CPU và bộ nhớ (Wokwi có hiển thị debug), ESP32 dùng rất ít (<20% CPU, RAM dư nhiều) cho tác vụ này, cho thấy dư địa để tích hợp thêm chức năng. Điều này dự kiến tương đồng trên thực tế, vì xử lý RFID và vài tác vụ IoT nhẹ không phải gánh nặng với ESP32.

Tóm lại, kết quả mô phỏng khẳng định tính đúng đắn của thiết kế. Hệ thống đáp ứng được yêu cầu đề ra: mở khóa bằng RFID và qua internet, đồng thời có cơ chế an toàn (tự khóa lại, cảnh báo khi sai). Mọi thành phần đã giao tiếp với nhau theo luồng dữ liệu dự kiến. Đây là bước tiền đề quan trọng trước khi triển khai thật, giúp chúng ta tự tin về giải pháp đề xuất.

4.2. Khả năng mở khóa qua RFID và xác thực không dây

Mở khóa qua RFID: Đây là phương thức xác thực cục bộ. Khi người có thẻ đến trực tiếp tại cửa, chỉ cần quét thẻ để vào. Ưu điểm: nhanh chóng (gần như tức thì), đơn giản (chỉ cần mang thẻ), và không phụ thuộc Internet. Kể cả khi nhà mất mạng hoặc server đám mây gặp sự cố, chủ nhà vẫn vào được bằng thẻ. RFID cũng không yêu cầu năng lượng nhiều (đầu đọc tiêu thụ ít, thẻ thụ động không cần pin). Nhược điểm: người dùng phải mang thẻ bên mình. Nếu quên hoặc mất thẻ, sẽ không vào được (trừ khi dùng phương thức khác). Thẻ RFID cũng có thể bị đánh cắp hoặc sao chép nếu kẻ xấu có kỹ thuật (dù MIFARE Classic có bảo mật cơ bản nhưng đã có trường hợp bị hack). Tuy nhiên, trong phạm vi hộ gia đình, nguy cơ sao chép thấp; chủ yếu rủi ro là

mất thẻ – khi đó cần xóa thẻ đó khỏi danh sách ngay (có thể thực hiện từ app nếu làm tính năng quản lý thẻ).

Mở khóa qua ứng dụng (Wi-Fi/Blynk): Đây là phương thức xác thực từ xa. Người dùng (chủ nhà) có thể ở bất kỳ đâu, chỉ cần điện thoại kết nối mạng và ứng dụng Blynk để mở khóa. Ưu điểm: linh hoạt – mở cửa cho người khác mà không cần hiện diện (ví dụ bạn đến chơi nhưng chủ vắng, chủ có thể mở từ xa cho vào), quản lý tập trung – ứng dụng có thể quản lý nhiều ổ khóa (nếu nhà có nhiều cửa) trên cùng giao diện, và an toàn – app được bảo vệ bằng tài khoản và kết nối Blynk an toàn (dùng token, có thể qua SSL). Nhược điểm: Phụ thuộc vào Internet – nếu mất mạng Wi-Fi nhà hoặc điện thoại không có 4G, sẽ không điều khiển được. Độ trễ cũng cao hơn một chút so với RFID trực tiếp. Ngoài ra, mở khóa từ xa yêu cầu sự cẩn trọng: lệnh mở do chủ nhà thực hiện, nhưng phải đảm bảo người được mở có mặt ở cửa – tránh mở nhầm cho người lạ. Hệ thống có thể nâng cấp bằng camera để chủ nhà nhìn thấy ai đang trước cửa trước khi bấm mở từ xa (nhưng đó là phạm vi khác).

Kết hợp RFID và Blynk, hệ thống trở thành một dạng xác thực hai lớp tùy chọn: Thông thường dùng RFID tại chỗ; trong trường hợp đặc biệt hoặc khẩn cấp mới dùng mở từ xa. Trong tương lai, có thể mở rộng cho phép một người cần cả thẻ RFID và xác nhận qua app mới mở (hai yếu tố), nhưng hiện tại hai phương thức tách biệt phục vụ tiện lợi.

Một lợi ích nữa của xác thực không dây là khả năng tích hợp với thiết bị khác: ví dụ, dùng Bluetooth trên điện thoại để mở khóa khi đứng gần (thay vì mở app, có thể tự động nếu trong khoảng BLE), hoặc dùng NFC trên điện thoại giả lập thẻ RFID (thay cho thẻ vật lý). Thực tế một số khóa thông minh thương mại đã cho phép dùng điện thoại như chìa khóa, bằng Bluetooth hay mã QR. Hệ thống của chúng ta với ESP32 hoàn toàn có thể tích hợp những tính năng này nhờ hỗ trợ đa kết nối.

Tổng kết, khả năng mở khóa đa dạng làm tăng tính khả dụng của hệ thống. RFID mang lại sự ổn định, nhanh tại chỗ; còn điều khiển không dây (Wi-Fi/Bluetooth) mở ra tiện ích hiện đại, phù hợp xu hướng nhà thông minh kết nối. Trong mọi trường hợp, an ninh vẫn được đảm bảo thông qua các cơ chế xác thực (token Blynk bảo vệ kênh Wi-Fi, danh sách UID bảo vệ kênh RFID). Nếu triển khai thực tế, nên bổ sung thêm các biện pháp an ninh mạng (như xác thực hai lớp cho tài khoản Blynk, mã hóa

giao tiếp MQTT nếu dùng) và an ninh vật lý (như chống sao chép thẻ, vỏ bảo vệ đầu đọc) để tăng cường hơn nữa.

4.3. Các vấn đề kỹ thuật và cách khắc phục

Rủi ro bảo mật RFID: Như đã đề cập, thẻ RFID MIFARE Classic có thể bị sao chép nếu kẻ gian có thiết bị đặc biệt (đọc trộm UID và giả mạo thẻ). Để khắc phục: dùng RFID ngẫu nhiên mã hóa (như MIFARE DesFire, iClass) an toàn hơn – nhưng đầu đọc và thẻ loại này đắt. Hoặc triển khai xác thực hai yếu tố: ví dụ thẻ + nhập mã PIN trên app, hoặc thẻ + quét vân tay (có module vân tay rời). Trong phạm vi khóa hộ gia đình phổ thông, giải pháp thực tiễn là nếu mất thẻ phải xóa thẻ đó khỏi hệ thống ngay. Bên cạnh đó, đặt đầu đọc ở bên ngoài cửa, còn ESP32 và rơ-le ở bên trong nhà để tránh trường hợp kẻ xấu tháo đầu đọc đấu dây trái phép – tín hiệu giữa RC522 và ESP32 có thể mã hóa hoặc kiểm tra serial để ngăn giả mạo.

Mất kết nối Internet: Nếu nhà mất mạng, chức năng mở từ xa qua Blynk không hoạt động. Điều này không ảnh hưởng đến RFID tại chỗ, nhưng chủ nhà ở xa tạm thời không can thiệp được. Khắc phục: có thể tích hợp SIM GSM/GPRS cho ESP32 (qua module SIM800) như một kênh dự phòng gửi SMS điều khiển, nhưng phức tạp. Hoặc đơn giản hơn, đảm bảo có người cầm thẻ dự phòng để mở khi cần. Ngoài ra, Blynk có thể cài đặt offline notifications – tức khi thiết bị offline, server báo để chủ biết nhà mất kết nối. Trong thời gian offline, hệ thống vẫn hoạt động cục bộ bình thường.

Sai sót logic lập trình: Lỗi code có thể gây tình huống nguy hiểm như mở khóa sai. Ví dụ, nếu biến lưu trạng thái bị lệch, có thể cửa mở mà hệ thống nghĩ là đóng và không tự khóa lại. Để phòng tránh, code cần được kiểm thử kỹ (điều đã làm qua mô phỏng). Thêm các cơ chế failsafe: như nếu trong 10 giây cửa ở trạng thái mở mà không đóng, ESP32 sẽ tự động khóa lại bất kể lý do gì. Hoặc nếu nhận lệnh mở hai lần liên tiếp, chỉ xử lý lần đầu cho đến khi khóa lại hẳn mới cho mở lần nữa (tránh rung relay liên tục). Nên xử lý ngắt debouncing cho nút (nếu có) và chống quét thẻ quá nhanh liên tục (để không bị spam mở/đóng nhanh gây hại cơ khí).

Nhìn chung, các vấn đề kể trên đều có giải pháp khả thi. Quan trọng là dự án nhận diện được để hoàn thiện thiết kế. Việc kết hợp mô phỏng và phân tích giúp giảm thiểu rủi ro khi triển khai thực tế và đảm bảo hệ thống cuối cùng đạt độ tin cậy, an toàn cao nhất có thể.

KẾT LUẬN

1. Kết luận

Đề tài "Hệ thống khóa cửa thông minh dựa trên ESP32 và RFID" đã trình bày quá trình xây dựng một giải pháp khóa cửa ứng dụng IoT từ khâu lên ý tưởng, nghiên cứu nền tảng đến thiết kế chi tiết và mô phỏng kiểm chứng. Trong Chương 1, chúng ta đã thấy được tầm quan trọng của khóa cửa thông minh trong bối cảnh hiện đại và mục tiêu hướng tới một hệ thống an toàn, tiện lợi cho người dùng. Chương 2 cung cấp cơ sở lý thuyết vững chắc: khái niệm khóa thông minh, đặc điểm phần cứng ESP32, nguyên lý công nghệ RFID, các phương thức kết nối không dây (Wi-Fi, Bluetooth) cũng như công cụ IoT (Blynk) được lựa chọn, giúp làm nền tảng cho thiết kế hệ thống. Chương 3 đi sâu vào thiết kế, với sơ đồ khối, sơ đồ mạch và giải thích luồng hoạt động, sau đó thực hiện mô phỏng trên Wokwi – đây là bước minh họa trực quan cho thấy hệ thống vận hành theo đúng ý đồ. Chương 4 phân tích kết quả mô phỏng, chứng minh rằng hệ thống đáp ứng các chức năng đề ra: mở khóa bằng thẻ RFID và qua ứng dụng di động, hoạt động ổn định và có phương án xử lý cho các tình huống. Đồng thời, chương này cũng thảo luận những vấn đề kỹ thuật thực tế (bảo mật thẻ, nguồn điện, lỗi mạng…) và so sánh hệ thống với các giải pháp khác, qua đó nhấn mạnh những ưu điểm và định vị của giải pháp trong bức tranh chung. Cuối cùng, các ứng dụng và ý tưởng mở rộng cho thấy tính thực tiễn và triển vọng phát triển của hệ thống.

Kết quả nổi bật đạt được gồm:

- Xây dựng thành công mô hình khóa thông minh mẫu dùng ESP32 + RFID, tích hợp IoT (Blynk) với đầy đủ chức năng cơ bản (điều khiển khóa, xác thực, cảnh báo).
- Mô phỏng hoạt động trơn tru trên Wokwi, kiểm chứng logic điều khiển và tính đúng đắn của thiết kế mà không cần phần cứng thực.
- Bài viết đã hệ thống hóa kiến thức liên quan (IoT, vi điều khiển, truyền thông không dây, an ninh hệ thống nhúng) và đưa ra phân tích chi tiết giúp người đọc hiểu rõ cả lý thuyết lẫn thực hành.
- Đề xuất được các phương án giải quyết vấn đề và hướng phát triển rất cụ thể, tạo tiền đề cho các nghiên cứu/triển khai tiếp theo.

Nhìn chung, mục tiêu nghiên cứu đề ra ban đầu đã được hoàn thành. Hệ thống mô phỏng đáp ứng yêu cầu chức năng và cho thấy tiềm năng ứng dụng tốt. Đây là bước khởi đầu quan trọng nếu muốn tiến tới triển khai một hệ thống khóa thông minh thực tế.

2. Hạn chế và đề xuất phát triển tương lai

Bên cạnh những kết quả đạt được, đề tài vẫn còn một số hạn chế nhất định do giới hạn thời gian và phạm vi:

- Thứ nhất, hệ thống mới dừng ở mức mô phỏng chứ chưa triển khai trên phần cứng thật để thử nghiệm trong môi trường thực tế. Do đó, chưa thể đánh giá được hết các yếu tố như độ bền thiết bị, khả năng chống chịu thời tiết (nếu lắp ngoài trời), tín hiệu RFID trong môi trường có nhiễu, v.v. Việc triển khai thực tế trong tương lai là rất cần thiết, bao gồm cả thiết kế vỏ hộp, lắp đặt tại cửa, và kiểm thử trong thời gian dài.
- Thứ hai, về bảo mật, hệ thống sử dụng phương thức RFID MIFARE Classic tương đối an toàn ở mức cơ bản nhưng chưa phải tối ưu. Giao tiếp giữa ESP32 và Blynk cloud sử dụng token nhưng có thể cần kích hoạt SSL để mã hóa hoàn toàn. Hiện tại, người phát triển (admin) có toàn quyền thêm/xóa thẻ bằng cách nạp lại code – chưa có cơ chế quản trị người dùng thân thiện. Trong tương lai, nên phát triển một giao diện quản trị (web hoặc app) để thêm/xóa thẻ, phân quyền mà không cần lập trình lại, kèm theo yêu cầu xác thực admin chặt chẽ (mật khẩu, 2FA).
- Thứ ba, khả năng kết nối với hệ sinh thái IoT khác chưa được triển khai. Hiện tại hệ thống hoạt động độc lập trên Blynk. Đề xuất tương lai: tích hợp thêm MQTT để giao tiếp với một Home Automation server (như Home Assistant). Điều này cho phép khóa thông minh thật sự hòa vào ngôi nhà thông minh: ví dụ, khi mở khóa thì tự động bật đèn phòng khách vào ban đêm, hoặc khi hệ thống báo động bật thì khóa sẽ vô hiệu hóa thẻ thường, chỉ chủ nhà mới mở được. MQTT là giao thức nhẹ và ESP32 hỗ trợ tốt, do đó thêm vào sẽ tăng tính linh hoạt (Blynk có thể chạy đồng thời với MQTT).

Bảng dưới đây liệt kê các cảm biến/thiết bị phụ được tích hợp hoặc mô phỏng trong hệ thống, kèm mô tả chức năng và cách thức mô phỏng trên Wokwi:

Cảm biến/Thiết bi phụ	Chức năng trong hệ thống	Cách mô phỏng trên Wokwi
Đầu đọc RFID RC522	Đọc mã UID từ thẻ RFID để xác thực người dùng	Chưa có mô hình RC522, dùng Serial nhập UID thay thế
Thẻ RFID	Thẻ từ đóng vai trò "chìa khóa" cho người dùng	Mô phỏng bằng chuỗi UID nhập vào Serial Monitor
Cảm biến cửa	Phát hiện trạng thái cửa đóng hay mở (nam châm gắn cửa)	Dùng switch nối vào GPIO để bật/tắt (đóng/mở)
Nút nhấn	Nút yêu cầu mở cửa từ bên trong (exit button)	Dùng push button ảo trên Wokwi, hoặc phím bàn phím
Đèn LED Xanh/Đỏ	Báo trạng thái xác thực (Xanh: hợp lệ/mở khóa, Đỏ: từ chối)	Sử dụng LED ảo, nối GPIO, quan sát sáng/tắt trên mô phỏng
Còi Buzzer	Báo động âm thanh khi có truy cập trái phép hoặc mở khóa	Dùng buzzer ảo, mô phỏng bằng âm thanh hoặc quan sát pin xuất
Servo (SG90)	Mô phỏng chốt khóa cửa chuyển động khi khóa/mở	Dùng servo ảo nối GPIO PWM, quan sát góc quay
Module Wi-Fi (ESP32)	Kết nối internet để giao tiếp Blynk	ESP32 giả lập tự có Wi-Fi, Wokwi cung cấp mạng ảo Wokwi-GUEST
Ứng dụng Blynk	Giao diện điều khiển và giám sát từ xa trên điện thoại	Sử dụng app Blynk thật, kết nối tới thiết bị ảo qua internet

Bảng ghi chú: Một số cảm biến trên không bắt buộc trong hệ thống cơ bản, nhưng có thể thêm vào để mở rộng chức năng. Mô phỏng Wokwi hỗ trợ linh hoạt việc thêm chúng và tạo tín hiệu ảo để thử nghiệm.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt

[1] KhueNguyenCreator – *Tổng quan về sơ đồ chân ESP32 và ngoại vi*. (2021)

Tài liệu Tiếng Anh

[2] Guangdong AP Tenon – *Advantages and Disadvantages of Smart Door Lock*. (2023)

[3] TechTarget – *What is RFID and How does it work?*. (2024)

[4] Kuriosity.sg – *Arduino Tutorial: ESP32 Blynk RFID App Lock*. (2022)

[5] ESP32IO.com – *ESP32 RFID/NFC Door Lock System Tutorial*. (2022)

[6] Ruaya, P. "Smart Lock Technology: Developing and Enhancing Home Security using Android-Based Controlled Door Locking App's." *Int. J. Adv. Res. Sci. Commun. Technol* (2023): 538-547.

[7] Blynk – *Blynk Documentation* (2024)

[8] Wokwi Forum – *Wokwi - An Amazing Free Online Microcontroller Simulator. (2022)*