

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



HCMUTE

ĐỒ ÁN MÔN HỌC 1

**THIẾT KẾ HỆ THỐNG MÁY BÁN HÀNG TỰ
ĐỘNG MINI**

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN MÁY TÍNH

Sinh viên: **HỒ GIA HUYỀN**
MSSV: 22119083

HUỲNH MINH QUÝ
MSSV: 22119125

TP. HỒ CHÍ MINH – 06/2025

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN MÔN HỌC 1

**THIẾT KẾ HỆ THỐNG MÁY BÁN HÀNG TỰ
ĐỘNG MINI**

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **HỒ GIA HUYỀN**
MSSV: 22119083
HUỲNH MINH QUÝ
MSSV: 22119125

Hướng dẫn: **PGS.TS. TRƯƠNG NGỌC SƠN**

TP. HỒ CHÍ MINH – 06/2025

BẢN NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

LỜI CẢM ƠN

Nhóm em xin chân thành cảm ơn PGS.TS Trương Ngọc Sơn – giảng viên hướng dẫn đã tận tình chỉ bảo, hỗ trợ và định hướng cho chúng em trong suốt quá trình thực hiện đề tài. Sự hướng dẫn chuyên môn sâu sắc và những góp ý quý báu từ thầy là yếu tố quan trọng giúp em hoàn thiện đề tài này.

Em cũng xin gửi lời cảm ơn đến các thầy cô trong khoa đã tạo điều kiện học tập và nghiên cứu thuận lợi, cùng với bạn bè, gia đình đã luôn động viên, giúp đỡ em trong suốt quá trình thực hiện đề tài.

Mặc dù đã cố gắng nhưng không tránh khỏi những thiếu sót, em rất mong nhận được sự góp ý của thầy cô để hoàn thiện hơn trong những lần nghiên cứu sau.

Trân trọng cảm ơn!

MỤC LỤC

Chương 1	GIỚI THIỆU	1
1.1	GIỚI THIỆU.....	1
1.2	MỤC TIÊU ĐỀ TÀI.....	1
1.3	GIỚI HẠN ĐỀ TÀI.....	2
1.4	PHƯƠNG PHÁP NGHIÊN CỨU	2
1.5	ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	3
1.6	BỐ CỤC QUYỀN BÁO CÁO	3
Chương 2	CƠ SỞ LÝ THUYẾT.....	5
2.1	CÔNG NGHỆ RFID	5
2.1.1	Giới thiệu RFID	5
2.1.2	Nguyên lý hoạt động của RFID	6
2.1.3	Lý do lựa chọn RFID so với các phương pháp thanh toán khác	7
2.2	GOOGLE APPS SCRIPT (GAS).....	8
2.2.1	Giới thiệu	8
2.2.2	Kiến trúc và nguyên lý hoạt động	8
2.2.3	Nhiệm vụ của Google Apps Script trong hệ thống	8
2.2.4	Giao thức truyền thông	9
2.2.5	Ưu điểm và lý do chọn Google Apps Script.....	9
2.3	GIỚI THIỆU PHẦN CỨNG	10
2.3.1	Kit phát triển và vi điều khiển ESP32.....	10
2.3.2	Module RFID MFRC522	12
Chương 3	THIẾT KẾ HỆ THỐNG	14
3.1	THIẾT KẾ PHẦN CỨNG.....	14
3.1.1	Chức năng của phần cứng.....	14

3.1.2 Sơ đồ khối phần cứng	15
3.1.3 Thiết kế từng khối.....	16
3.1.3.1 Khối RFID	16
3.1.3.2 Khối nhập lệnh.....	16
3.1.3.3 Khối hiển thị	17
3.1.3.4 Khối âm báo	18
3.1.3.5 Khối truyền động	19
3.1.3.6 Khối nguồn.....	20
3.1.4 Sơ đồ nguyên lý toàn mạch.....	21
3.2 THIẾT KẾ PHẦN MỀM.....	22
3.2.1 Chức năng hoạt động của phần mềm.....	22
3.2.2 Lưu đồ hoạt động.....	23
Chương 4 KẾT QUẢ	28
4.1 KẾT QUẢ MÔ HÌNH THI CÔNG	28
4.1.1 Bo mạch phần cứng.....	28
4.1.2 Google sheet quản lý khách hàng phần mềm	30
4.1.3 Mô hình sản phẩm.....	30
4.2 HOẠT ĐỘNG CỦA HỆ THỐNG	31
4.2.1 Khởi động hệ thống.....	31
4.2.2 Nhận diện thẻ RFID	31
4.2.3 Chọn sản phẩm và thanh toán	32
4.2.4 Các trường hợp lỗi giao dịch	33
4.2.5 Nạp tiền cho khách hàng.....	34
4.3 THÔNG SỐ KẾT QUẢ VÀ THỰC NGHIỆM	35
4.3.1 Công suất tiêu thụ toàn hệ thống	35

4.3.2 Tỷ lệ lỗi khi trả hàng	36
4.3.3 Truy xuất thông tin người dùng từ Google Sheet	37
Chương 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	39
5.1 KẾT LUẬN	39
5.1.1 Kết quả đạt được so với mục tiêu ban đầu đặt ra	39
5.1.2 Ưu nhược điểm của hệ thống đã thiết kế	40
5.2 HƯỚNG PHÁT TRIỂN	41
PHỤ LỤC	42
TÀI LIỆU THAM KHẢO	44

DANH MỤC HÌNH

Hình 1: Mô hình RFID.....	5
Hình 2: Sơ đồ chân kit phát triển ESP32	10
Hình 3: Module RFID MFRC522.....	12
Hình 4: Sơ đồ khối toàn hệ thống	15
Hình 5: Sơ đồ nguyên lý kết nối module RFID MFRC522 với ESP32.....	16
Hình 6: Sơ đồ nguyên lý kết nối Keypad 4x4 với ESP32.....	16
Hình 7: Sơ đồ nguyên lý kết nối LCD 20x4 I2C với ESP32	17
Hình 8: Sơ đồ nguyên lý kết nối Buzzer với ESP32.....	18
Hình 9: Sơ đồ nguyên lý kết nối khối truyền động với ESP32.....	19
Hình 10: Sơ đồ nguyên lý khối nguồn	20
Hình 11: Sơ đồ nguyên lý toàn hệ thống	21
Hình 12: Sơ đồ nguyên lý được thiết kế trên phần mềm Altium	21
Hình 13: Lưu đồ tổng quát.....	24
Hình 14: Lưu đồ lấy thông tin Sheet.....	25
Hình 15: Lưu đồ cập nhật số dư sau giao dịch lên Sheet.....	26
Hình 16: PCB layout được thiết kế trên Altium Design	28
Hình 17: Sơ đồ chân nối và mạch 3D mô phỏng trên Altium	28
Hình 18: Sơ đồ mạch in lớp trên và lớp dưới của toàn mạch	29
Hình 19: Board mạch thi công thực tế lớp trên và lớp dưới của toàn mạch.....	29
Hình 20: Giao diện quản lý thông tin khách hàng bằng Google sheet	30
Hình 21: Mô hình hệ thống máy bán hàng tự động mini.....	30
Hình 22: Hệ thống khởi động	31
Hình 23: Hệ thống nhận diện thẻ RFID thành công	31
Hình 24: Hệ thống nhận diện thẻ RFID thất bại	32
Hình 25: Quy trình chọn sản phẩm và thanh toán thành công.....	33
Hình 26: Kết thúc giao dịch	33
Hình 27: Trường hợp số dư không đủ để mua sản phẩm.....	34
Hình 28: Trường hợp chọn mã sản phẩm không hợp lệ:	34
Hình 29: Các bước nạp tiền	35

DANH MỤC BẢNG

Bảng 1: Bảng thống kê tỷ lệ lỗi	36
Bảng 2: Bảng thống kê thời gian truy xuất thông tin khách hàng thứ 5	37
Bảng 3: Bảng thống kê thời gian truy xuất thông tin khách hàng thứ 100	37

CÁC TỪ VIẾT TẮT

RFID	Radio Frequency Identification
IC	Integrated Circuit
AI	Artificial Intelligent
LCD	Liquid Crystal Display
LF	Low Frequency
HF	High Frequency
UHF	Ultra High Frequency
NFC	Near Field Communication
QR	Quick-Response code
PIN	Personal Identification Number
GPIO	General Purpose Input/Output
IoT	Internet of Things
RAM	Random Access Memory
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
BLE	Bluetooth Low Energy
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
PWM	Pulse Width Modulation
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DC	Direct Current
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
SCLK	Serial Clock
SS	Slave Select
MOSI	Master Out Slave In
MISO	Master In Slave Out

RTC	Real-Time Clock
SDA	Serial Data
SCL	Serial Clock
ACK	Acknowledge
NACK	Not Acknowledge
UID	User Identification
LF	Low Frequency
HF	High Frequency
UHF	Ultra High Frequency
GAS	Google Apps Script
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure

Chương 1

GIỚI THIỆU

1.1 GIỚI THIỆU

Trong bối cảnh công nghệ tự động hóa phát triển, máy bán hàng tự động đang được ứng dụng rộng rãi, không chỉ ở các quốc gia phát triển mà còn tại Việt Nam ở các khu vực công cộng như trường học, bệnh viện, nhà ga, sân bay. Với ưu điểm tiết kiệm nhân công, phục vụ 24/7 và thao tác nhanh, các máy này góp phần nâng cao chất lượng dịch vụ và đáp ứng nhu cầu tiêu dùng tiện lợi. Các nghiên cứu hiện nay chú trọng cải tiến chức năng, các công nghệ thông minh như nhận diện hình ảnh, thanh toán không dùng tiền mặt và điều khiển từ xa qua internet ngày càng được tích hợp vào các hệ thống này. Tuy nhiên, xây dựng một mô hình máy bán hàng tự động đơn giản với các chức năng cơ bản vẫn là hướng nghiên cứu hữu ích cho sinh viên, giúp họ áp dụng kiến thức về vi điều khiển, lập trình nhúng, thiết kế hệ thống và thi công mạch. Đề tài “Thiết kế hệ thống máy bán hàng tự động mini” với thanh toán RFID hướng tới mô phỏng các chức năng như nhận diện thẻ, chọn sản phẩm, trả hàng và hiển thị thông tin giao dịch, đồng thời tạo nền tảng phát triển các tính năng thông minh trong tương lai.

1.2 MỤC TIÊU ĐỀ TÀI

- Thiết kế hệ thống nhận diện thẻ RFID để thực hiện thanh toán.
- Thiết kế hệ thống điều khiển và chọn sản phẩm thông qua bàn phím ma trận.
- Thiết kế cơ cấu trả hàng tự động sau khi thanh toán thành công.
- Hiển thị thông tin giao dịch, trạng thái hoạt động và số dư tài khoản RFID trên màn hình LCD.

- Tích hợp hệ thống âm báo để phản hồi các thao tác, xác nhận giao dịch thành công hoặc cảnh báo khi có lỗi xảy ra.
- Kết nối và quản lý thông tin người dùng, số dư tài khoản, cũng như chức năng nạp tiền thông qua Google Sheets sử dụng Google Apps Script.

1.3 GIỚI HẠN ĐỀ TÀI

- Mô hình được xây dựng ở dạng thu nhỏ, phù hợp với môi trường học tập và nghiên cứu.
- Hệ thống chỉ hỗ trợ thanh toán bằng công nghệ RFID, không tích hợp các phương thức thanh toán khác như tiền mặt hay thẻ ngân hàng.
- Hệ thống cũng chỉ thực hiện các chức năng cơ bản như nhận diện thẻ, chọn sản phẩm, kiểm tra thông tin và trả hàng, nạp tiền cho thẻ.
- Số lượng và loại sản phẩm giới hạn, chỉ phục vụ mục đích mô phỏng, không đa dạng như các hệ thống bán hàng tự động thực tế.
- Chỉ quản lý thông tin cơ bản của thẻ RFID như số dư và mã nhận diện, chưa tích hợp các tính năng bảo mật nâng cao.
- Không thiết kế app và giao diện để quản lý thẻ, chỉ giới hạn ở việc dùng platform có sẵn hoặc google sheet.

1.4 PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp tổng hợp tài liệu lý thuyết: Thu thập, nghiên cứu và hệ thống hóa các tài liệu, giáo trình, bài báo khoa học liên quan đến công nghệ RFID, hệ thống bán hàng tự động và các phương pháp điều khiển, quản lý dữ liệu.

Phương pháp khảo sát hệ thống thực tế: Tìm hiểu, quan sát các mô hình máy bán hàng tự động hiện có, ghi nhận những ưu điểm, nhược điểm và cách thức hoạt động của các hệ thống này làm cơ sở để thiết kế hệ thống mô phỏng.

Phương pháp tham khảo ý kiến chuyên gia: Trao đổi, thảo luận với giảng viên hướng dẫn trong lĩnh vực điện tử, vi điều khiển và công nghệ RFID để hoàn thiện giải pháp kỹ thuật.

Phương pháp thiết kế và mô phỏng: Tiến hành thiết kế sơ đồ khối, xây dựng và kiểm thử mô hình hệ thống bằng cách lập trình và lắp ráp các linh kiện phần

cứng như vi điều khiển ESP32, module RFID, màn hình hiển thị và hệ thống cơ cấu trả hàng.

Phương pháp thực nghiệm: Thực hiện các thí nghiệm đo đạc, kiểm tra và đánh giá khả năng hoạt động của hệ thống, từ đó điều chỉnh và tối ưu thiết kế để đạt được hiệu suất và độ chính xác cao nhất.

1.5 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu:

- Công nghệ RFID và ứng dụng trong hệ thống thanh toán không tiếp xúc.
- Ứng dụng Google Apps Script kết hợp Google Sheets như một giải pháp quản lý dữ liệu đơn giản trong hệ thống nhúng.
- Cấu trúc và nguyên lý hoạt động của máy bán hàng tự động.
- Các phương pháp điều khiển, quản lý và xử lý dữ liệu trên ESP32.
- Các thiết bị phần cứng như module RFID, màn hình LCD hiển thị, hệ thống cơ cấu trả hàng và các linh kiện điện tử liên quan.

Phạm vi nghiên cứu:

- Thiết kế và xây dựng mô hình máy bán hàng tự động mini, tập trung vào cơ chế thanh toán bằng thẻ RFID.
- Hệ thống được giới hạn ở việc nhận diện thẻ RFID, chọn sản phẩm, kiểm tra số dư thẻ, tình trạng thẻ, xác nhận giao dịch và thực hiện trả hàng.
- Không mở rộng các phương thức thanh toán khác ngoài RFID.
- Thực hiện nghiên cứu và phát triển mô hình ở quy mô nhỏ, phù hợp với môi trường học tập và nghiên cứu, chưa áp dụng trong thực tế thương mại.
- Dữ liệu người dùng và giao dịch được quản lý ở mức cơ bản, không sử dụng cơ sở dữ liệu chuyên dụng.

1.6 BỐ CỤC QUYỀN BÁO CÁO

Chương 1. TỔNG QUAN ĐỀ TÀI.

Chương này giới thiệu về đề tài nghiên cứu, bao gồm mục tiêu thực hiện, giới hạn đề tài, phương pháp nghiên cứu, đối tượng và phạm vi nghiên cứu.

Chương 2. CƠ SỞ LÝ THUYẾT.

Trình bày các kiến thức nền tảng liên quan đến đề tài, bao gồm công nghệ RFID, nguyên lý hoạt động của các linh kiện và giao thức truyền thông được sử dụng trong hệ thống.

Chương 3. THIẾT KẾ HỆ THỐNG.

Trình bày về sơ đồ khối, cách tính toán giá trị cho các khối, sơ đồ nguyên lý của mạch giao tiếp vi điều khiển với các module, sơ đồ nguyên lý của toàn hệ thống, lập trình phần mềm, lưu đồ hệ thống.

Chương 4. KẾT QUẢ.

Trình bày kết quả đạt được sau khi hoàn thành hệ thống, bao gồm kiểm thử phần cứng, phần mềm, đánh giá hiệu suất và so sánh với mục tiêu ban đầu.

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.

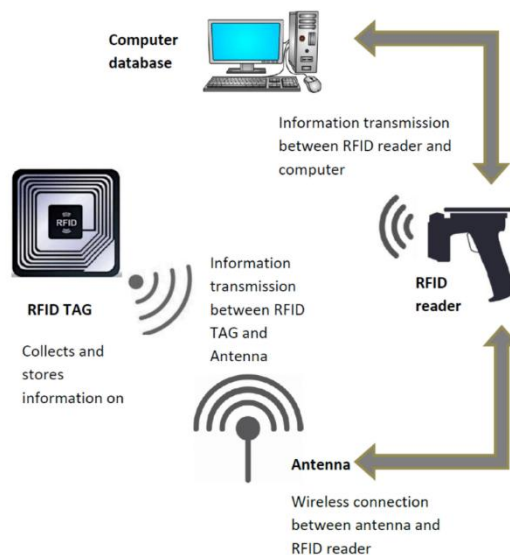
Tổng kết lại kết quả nghiên cứu, đánh giá những ưu điểm và hạn chế của hệ thống. Đồng thời, đề xuất các hướng phát triển trong tương lai để cải thiện và mở rộng ứng dụng của hệ thống.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1 CÔNG NGHỆ RFID

2.1.1 Giới thiệu RFID



Hình 1: Mô hình RFID

RFID (Radio Frequency Identification) là công nghệ nhận diện bằng sóng vô tuyến, cho phép trao đổi dữ liệu giữa thẻ RFID và đầu đọc RFID mà không cần tiếp xúc vật lý. RFID có thể đọc những thông tin xuyên qua các vật cản không dẫn điện như giấy, nhựa hay vải và các điều kiện môi trường thách thức khác mà mã vạch và các công nghệ khác không thể phát huy hiệu quả. Nhờ khả năng hoạt động nhanh, chính xác và bảo mật cao, RFID được ứng dụng rộng rãi trong nhiều lĩnh vực như kiểm soát ra vào, quản lý kho hàng và đặc biệt là thanh toán tự động^[1].

Hệ thống RFID cơ bản bao gồm hai thành phần chính là thẻ RFID (tag) và thiết bị đọc RFID (reader). Thẻ RFID chứa một con chip lưu trữ thông tin định danh cùng với một anten nhỏ để giao tiếp bằng sóng vô tuyến. Thiết bị đọc RFID

phát ra một tín hiệu điện từ tại tần số xác định (thường là 13.56 MHz đối với hệ thống sử dụng MFRC522), tín hiệu này cung cấp năng lượng cho thẻ RFID (nếu là loại thụ động), sau đó thẻ sẽ gửi dữ liệu phản hồi về cho đầu đọc. Dữ liệu thu được thường là một mã số duy nhất (UID)^[1].

Hệ thống RFID được chia thành hai loại chính là RFID thụ động và RFID chủ động. Trong đó, thẻ RFID thụ động không có nguồn điện riêng mà hoàn toàn phụ thuộc vào năng lượng do đầu đọc cung cấp, tầm hoạt động của loại thẻ này khoảng vài cm. Điều này giúp thẻ có chi phí thấp, nhỏ gọn và tuổi thọ dài. Ngược lại, thẻ RFID chủ động có nguồn điện tích hợp, cho phép truyền tín hiệu mạnh hơn và trong phạm vi xa hơn, có thể lên đến hàng trăm mét, nhưng lại đắt tiền và cồng kềnh hơn^[1].

RFID hoạt động ở nhiều dải tần khác nhau như LF (125-134 kHz), HF (13.56 MHz) và UHF (860-960 MHz), trong đó dải HF là phổ biến nhất trong các ứng dụng dân dụng. Dải HF được sử dụng trong thẻ MIFARE – một loại thẻ RFID thông dụng tương thích với module MFRC522. Với dải tần này, khoảng cách nhận diện hiệu quả thường từ 2 đến 5 cm, tùy thuộc vào loại thẻ và điều kiện môi trường^[1].

Với sự phát triển mạnh mẽ của công nghệ, RFID đã trở thành một phần không thể thiếu trong các hệ thống nhận dạng hiện đại. Trong khuôn khổ đề án này, RFID đóng vai trò là trung tâm trong quá trình xác thực người dùng và xử lý thanh toán tự động, góp phần xây dựng một hệ thống máy bán hàng mini thông minh, nhanh chóng và tiện lợi cho người sử dụng.

2.1.2 Nguyên lý hoạt động của RFID

Nguyên lý hoạt động của công nghệ RFID dựa trên việc truyền và nhận dữ liệu qua sóng vô tuyến giữa hai thành phần chính là thẻ RFID (gọi là tag) và đầu đọc RFID (gọi là reader). Khi hệ thống bắt đầu hoạt động, thiết bị đọc RFID phát ra một sóng điện từ có tần số xác định – thường là 13.56 MHz đối với các hệ thống sử dụng thẻ MIFARE, như trong module MFRC522. Sóng này đóng vai trò như một tín hiệu dò tìm đồng thời cũng cung cấp năng lượng cho các thẻ RFID thụ động nằm trong phạm vi hoạt động của reader.

Thẻ RFID không có nguồn điện riêng mà lấy năng lượng từ sóng radio do đầu đọc cung cấp. Khi nhận được năng lượng đủ để kích hoạt mạch điện bên trong, thẻ bắt đầu phản hồi dữ liệu. Quá trình truyền thông tin từ thẻ về đầu đọc được thực hiện thông qua một kỹ thuật gọi là điều chế lại sóng phản xạ (backscatter modulation). Trong kỹ thuật này, thẻ thay đổi cách phản xạ sóng radio đến từ reader bằng cách điều chỉnh trở kháng nội mạch của mình. Những biến đổi này tạo nên các tín hiệu dữ liệu mang thông tin được mã hóa, như số định danh duy nhất (UID) được lưu trong bộ nhớ của thẻ.

Khi đầu đọc nhận được sóng phản xạ đã được điều chế, nó sẽ giải mã tín hiệu thành dữ liệu số. Dữ liệu này sau đó có thể được xử lý bởi vi điều khiển như ESP32 để thực hiện các hoạt động điều khiển khác tùy trong ứng dụng thực tế.

Điểm đặc biệt trong nguyên lý hoạt động của RFID là quá trình giao tiếp hoàn toàn không cần tiếp xúc vật lý, giúp tăng độ bền của thiết bị cũng như sự tiện lợi cho người dùng. Thẻ có thể được nhận diện chỉ bằng cách đưa lại gần đầu đọc trong khoảng cách từ 0 đến vài cm, không yêu cầu căn chỉnh chính xác hay quét thủ công như mã vạch^[1].

Tóm lại, công nghệ RFID hoạt động như một hệ thống truyền năng lượng và dữ liệu không dây hai chiều giữa đầu đọc và thẻ, với tốc độ nhanh, chính xác và không cần chạm, mở ra nhiều ứng dụng tiện ích trong thực tế từ kiểm soát ra vào đến tự động hóa thanh toán, như trong hệ thống bán hàng mini của đồ án này.

2.1.3 Lý do lựa chọn RFID so với các phương pháp thanh toán khác

RFID là lựa chọn tối ưu cho hệ thống máy bán hàng tự động mini nhờ vào tốc độ giao dịch nhanh chóng và sự tiện lợi, khi không cần tiếp xúc trực tiếp như quét thẻ từ hay nhập mã PIN, giúp giảm đáng kể thời gian thanh toán. Bên cạnh đó, công nghệ này còn đảm bảo độ chính xác và tính bảo mật cao nhờ khả năng mã hóa dữ liệu, hạn chế nguy cơ sao chép hoặc gian lận. So với thẻ từ, RFID có độ bền vượt trội do không bị hao mòn cơ học trong quá trình sử dụng. Hơn nữa, hệ thống RFID cho phép quản lý tài khoản dễ dàng, hỗ trợ kiểm soát số dư và nạp tiền một cách nhanh chóng. Đặc biệt, RFID có thể hoạt động độc lập mà không cần kết nối internet như QR Code, giúp hệ thống duy trì sự ổn định ngay cả trong

môi trường không có mạng. Với những ưu điểm này, RFID không chỉ nâng cao tốc độ giao dịch mà còn giảm thiểu sai sót, tối ưu hóa trải nghiệm người dùng trong các hệ thống thanh toán tự động.

2.2 GOOGLE APPS SCRIPT (GAS)

2.2.1 Giới thiệu

Google Apps Script (GAS) là một nền tảng phát triển ứng dụng dựa trên đám mây được cung cấp bởi Google. GAS cho phép lập trình viên sử dụng ngôn ngữ JavaScript để tự động hóa, mở rộng và tích hợp các dịch vụ trong hệ sinh thái Google như Google Sheets, Google Docs, Gmail, Google Drive và nhiều dịch vụ khác. GAS được xây dựng trên nền tảng điện toán đám mây của Google, hoạt động dựa trên engine JavaScript V8 hiện đại, giúp tạo ra các ứng dụng web và API hiệu quả^[2].

2.2.2 Kiến trúc và nguyên lý hoạt động

Google Apps Script hoạt động dựa trên cơ chế xử lý các yêu cầu HTTP gửi đến thông qua các hàm tiêu chuẩn như doGet(e) và doPost(e). Khi nhận được yêu cầu, GAS sẽ thực thi đoạn mã xử lý tương ứng và trả về kết quả dưới dạng dữ liệu văn bản hoặc JSON tùy theo thiết kế. Toàn bộ dữ liệu và mã nguồn được lưu trữ và thực thi trên nền tảng đám mây của Google, giúp người dùng dễ dàng triển khai và quản lý ứng dụng mà không cần sử dụng các máy chủ vật lý. Các kết nối thường sử dụng giao thức HTTPS nhằm đảm bảo tính bảo mật trong truyền tải dữ liệu^[3].

2.2.3 Nhiệm vụ của Google Apps Script trong hệ thống

Trong hệ thống quản lý bán hàng sử dụng ESP32, Google Apps Script đóng vai trò trung gian kết nối giữa thiết bị phần cứng và cơ sở dữ liệu người dùng được lưu trữ trên Google Sheets. GAS nhận và xử lý các yêu cầu HTTP gửi từ ESP32, bao gồm truy vấn thông tin người dùng dựa trên mã định danh thẻ RFID (UID), như họ tên và số dư tài khoản. Bên cạnh đó, GAS cũng nhận các yêu cầu cập nhật số dư sau mỗi giao dịch và thực hiện ghi dữ liệu mới vào Google Sheets tương ứng. Nhờ đó, GAS đảm bảo dữ liệu người dùng luôn được đồng bộ chính xác và

phản hồi kịp thời cho ESP32, tạo thành chu trình giao tiếp hai chiều ổn định và hiệu quả^[3].

2.2.4 Giao thức truyền thông

Việc giao tiếp giữa ESP32 và Google Apps Script được thực hiện qua giao thức HTTPS dựa trên HTTP/1.1 qua mạng WiFi. Mặc dù GAS yêu cầu kết nối bảo mật nhằm đảm bảo tính toàn vẹn và bảo mật dữ liệu truyền tải, trong thực tế lập trình cho ESP32 thường sử dụng hàm `client.setInsecure()` để bỏ qua bước xác thực chứng chỉ SSL. Điều này xuất phát từ giới hạn về tài nguyên phần cứng của ESP32, như bộ nhớ và khả năng xử lý, khiến việc xác thực đầy đủ chuỗi chứng chỉ trở nên khó khăn hoặc không khả thi. Tuy nhiên, việc bỏ qua xác thực SSL tiềm ẩn nguy cơ về bảo mật thông tin, đặc biệt trong môi trường mạng không an toàn, do dữ liệu có thể bị tấn công hoặc đánh cắp trong quá trình truyền tải^[3].

Về phương thức truyền tải, ESP32 sử dụng yêu cầu GET để truy xuất thông tin người dùng, với tham số UID truyền qua URL, đồng thời sử dụng POST với dữ liệu được mã hóa theo chuẩn `application/x-www-form-urlencoded` để cập nhật số dư sau giao dịch. Cách tiếp cận này giúp đơn giản hóa việc tích hợp giữa thiết bị nhúng ESP32 và dịch vụ đám mây của Google, đồng thời đảm bảo hiệu năng phù hợp trong môi trường hạn chế tài nguyên^[3].

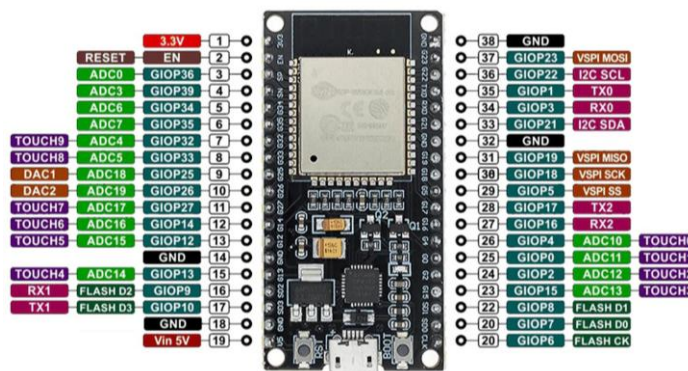
2.2.5 Ưu điểm và lý do chọn Google Apps Script

Google Apps Script được lựa chọn bởi nhiều lợi thế nổi bật trong việc xây dựng hệ thống. Thứ nhất, GAS hoàn toàn miễn phí, giúp giảm thiểu chi phí so với việc phải thuê server hoặc duy trì cơ sở dữ liệu riêng biệt. Thứ hai, GAS rất dễ triển khai và sử dụng bởi mọi thứ đều được chạy trên nền tảng đám mây của Google, không đòi hỏi người dùng phải quản lý hay cấu hình phần cứng phức tạp. Thứ ba, GAS tích hợp trực tiếp với Google Sheets, mang lại sự thuận tiện trong việc lưu trữ và quản lý dữ liệu người dùng dưới dạng bảng tính thân thiện và dễ chỉnh sửa. Thêm vào đó, GAS hỗ trợ giao thức HTTPS giúp nâng cao tính bảo mật trong quá trình truyền dữ liệu nếu được sử dụng đúng cách. Cuối cùng, do GAS

hoạt động trên nền tảng đám mây, người dùng không cần phải lo lắng về vấn đề bảo trì, vận hành phân cứng hay rủi ro hư hỏng thiết bị.

2.3 GIỚI THIỆU PHẦN CỨNG

2.3.1 Kit phát triển và vi điều khiển ESP32



Hình 2: Sơ đồ chân kit phát triển ESP32

ESP32 là một bo mạch vi điều khiển mạnh mẽ do Espressif Systems phát triển, dựa trên chip ESP32. Đây là phiên bản nâng cấp từ ESP8266, tích hợp WiFi và Bluetooth, cung cấp hiệu suất cao hơn, nhiều chân GPIO, và hỗ trợ đa dạng giao thức giao tiếp. Các kit phát triển ESP32 thường đi kèm cổng USB nạp/chạy, bộ nguồn ổn áp, và hệ thống chân GPIO thuận tiện để kết nối với cảm biến, module mở rộng và các thiết bị ngoại vi khác.

Đặc tính chung và các thông số cần quan tâm:

ESP32 là một vi điều khiển mạnh mẽ được trang bị vi xử lý Dual-core Xtensa LX6 với tốc độ tối đa lên đến 240 MHz, giúp xử lý tác vụ nhanh chóng và hiệu quả. Bộ nhớ RAM 520 KB SRAM và bộ nhớ Flash tích hợp từ 4 MB trở lên cung cấp không gian lưu trữ đủ lớn cho các ứng dụng nhúng. Khả năng kết nối không dây là một điểm nổi bật với Wi-Fi 802.11 b/g/n và Bluetooth 4.2 hỗ trợ cả chế độ BLE và Classic, giúp ESP32 giao tiếp linh hoạt với nhiều thiết bị khác nhau. Vi điều khiển hoạt động ở mức điện áp 3.3V, trong khi đó, kit phát triển cần được cấp nguồn 5V để đảm bảo hoạt động ổn định. Hệ thống giao tiếp đa dạng bao gồm UART, SPI, I2C, PWM, ADC 12 bit và DAC 8 bit, cho phép ESP32 tương tác với nhiều loại cảm biến và thiết bị ngoại vi. Với tổng cộng 38 chân GPIO, vi điều khiển

cung cấp nhiều tùy chọn kết nối cho các ứng dụng phức tạp. Dòng điện tiêu thụ khoảng 160 mA khi hoạt động, và mỗi chân GPIO có thể chịu dòng tối đa khoảng 40 mA, tuy nhiên, khuyến nghị không vượt quá 20 mA để tránh hư hỏng linh kiện^[4].

Nhiệm vụ trong hệ thống:

Trong hệ thống máy bán hàng tự động mini, ESP32 đóng vai trò là vi điều khiển trung tâm, chịu trách nhiệm xử lý toàn bộ quá trình giao dịch. Nó nhận tín hiệu từ đầu đọc RFID thông qua chuẩn SPI để xác định thông tin thẻ thanh toán, đảm bảo tính chính xác và bảo mật trong quá trình xác thực. Khi giao dịch được xác nhận, ESP32 điều khiển cơ chế trả hàng bằng cách kích hoạt động cơ, giúp sản phẩm được đưa ra một cách tự động. Ngoài ra, vi điều khiển này còn giao tiếp với màn hình LCD qua chuẩn I2C để hiển thị thông tin sản phẩm, số dư tài khoản và trạng thái giao dịch. Bên cạnh đó, ESP32 cũng quản lý nguồn và kết nối với các linh kiện khác, giúp hệ thống vận hành ổn định. Đặc biệt, với khả năng kết nối WiFi, ESP32 có thể truy xuất cơ sở dữ liệu trực tuyến, hỗ trợ nạp tiền vào thẻ cũng như lưu trữ thông tin khách hàng.

Lý do lựa chọn: Khả năng xử lý mạnh, dễ lập trình với môi trường Arduino IDE và có nhiều thư viện hỗ trợ kết nối WiFi, Bluetooth, giao tiếp cảm biến (I2C, SPI, UART), điều khiển thiết bị ngoại vi (PWM, ADC, DAC), giúp rút ngắn thời gian phát triển.

2.3.2 Module RFID MFRC522



Hình 3: Module RFID MFRC522

MFRC522 là một module RFID hoạt động ở tần số 13.56MHz, thường được sử dụng để đọc và ghi dữ liệu lên thẻ RFID. Module này hỗ trợ giao tiếp SPI, I2C và UART, nhưng phổ biến nhất là SPI.

Nguyên lý hoạt động:

Module là một reader hoạt động bằng cách phát sóng điện từ ở tần số 13.56MHz qua anten. Khi một thẻ RFID nằm trong phạm vi phủ sóng, module sẽ cung cấp năng lượng cho thẻ để kích hoạt mạch bên trong. Thẻ phản hồi lại thông tin bằng cách điều chỉnh sóng phản xạ. Module MFRC522 tiếp nhận tín hiệu phản hồi và giải mã thông tin từ thẻ^[5].

Đặc tính chung và các thông số cần quan tâm:

Module RFID MFRC522 hoạt động ở điện áp 3.3V với mức tiêu thụ dòng điện dao động từ 13mA đến 26mA, đảm bảo khả năng vận hành ổn định trong các hệ thống nhúng. Sử dụng tần số 13.56MHz, module này có thể nhận diện thẻ RFID trong phạm vi từ 0 đến 6cm, tùy thuộc vào loại thẻ sử dụng. MFRC522 hỗ trợ nhiều loại thẻ RFID phổ biến như MIFARE 1K, MIFARE 4K và MIFARE Ultralight, giúp mở rộng khả năng ứng dụng trong các hệ thống kiểm soát truy cập và thanh toán tự động. Với giao tiếp SPI, module này có thể kết nối dễ dàng với vi điều khiển, đảm bảo tốc độ truyền dữ liệu lên đến 10Mbit/s, giúp quá trình nhận diện và xử lý thông tin diễn ra nhanh chóng^[5].

Nhiệm vụ trong hệ thống: Nhận diện thẻ RFID của người dùng để xác thực giao dịch. Truy xuất dữ liệu từ thẻ RFID và gửi thông tin đến vi điều khiển ESP32 thông qua giao tiếp SPI để xử lý thanh toán.

Lý do lựa chọn: Chi phí thấp, phù hợp với các ứng dụng nhỏ và trung bình. Bên cạnh đó, module này rất dễ sử dụng với nhiều thư viện hỗ trợ sẵn, giúp đơn giản hóa quá trình lập trình và tích hợp vào hệ thống. Ngoài ra, MFRC522 có khả năng tương thích với nhiều loại thẻ RFID khác nhau, mang lại sự linh hoạt trong các ứng dụng thanh toán. Đặc biệt, module này còn có tính bảo mật cao khi hỗ trợ mã hóa dữ liệu, đảm bảo an toàn cho giao dịch.

Chương 3

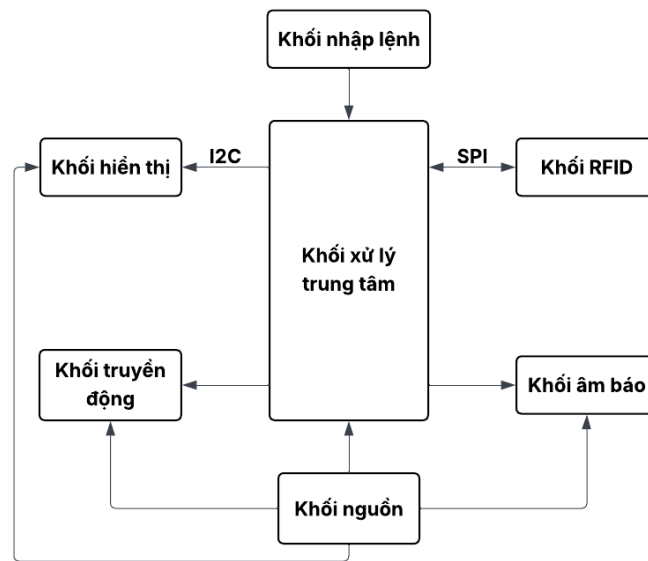
THIẾT KẾ HỆ THỐNG

3.1 THIẾT KẾ PHẦN CỨNG

3.1.1 Chức năng của phần cứng

Hệ thống máy bán hàng tự động mini được thiết kế nhằm thực hiện các chức năng tự động hóa trong quá trình mua bán. Người dùng có thể dễ dàng lựa chọn sản phẩm bằng cách nhập mã trên bàn phím Keypad, sau đó tiến hành thanh toán thông qua thẻ RFID. Đầu đọc RFID MFRC522 sẽ quét thông tin thẻ và truyền dữ liệu đến vi điều khiển trung tâm để kiểm tra số dư, xác thực và xử lý giao dịch. Khi thanh toán được xác nhận, hệ thống sẽ kích hoạt động cơ servo để đưa sản phẩm ra ngoài. Toàn bộ quá trình được hiển thị rõ ràng trên màn hình LCD với các thông tin như số dư, trạng thái thanh toán (thành công/thất bại) và hướng dẫn thao tác như “Chọn sản phẩm”, “Nhấn * để xác nhận”. Ngoài ra, buzzer sẽ phát tín hiệu âm thanh để thông báo các trạng thái như thao tác nhấn phím, thẻ không hợp lệ hoặc hoàn tất giao dịch, giúp nâng cao trải nghiệm sử dụng.

3.1.2 Sơ đồ khối phần cứng



Hình 4: Sơ đồ khối toàn hệ thống

Chức năng từng khối:

Khối xử lý trung tâm sử dụng vi điều khiển ESP32 có nhiệm vụ điều khiển toàn bộ hoạt động của hệ thống, đồng thời giao tiếp và trao đổi dữ liệu với các khối ngoại vi khác.

Khối đọc thẻ sử dụng module RFID MFRC522, hoạt động qua giao thức SPI để thu thập thông tin từ thẻ RFID và gửi về ESP32 để xử lý giao dịch.

Người dùng thực hiện thao tác lựa chọn sản phẩm hoặc xác nhận mua hàng thông qua khối nhập lệnh là bàn phím Keypad, dữ liệu nhập sẽ được truyền đến vi điều khiển để xử lý tiếp theo.

Khối hiển thị sử dụng màn hình LCD giao tiếp qua chuẩn I2C để hiển thị các thông tin cần thiết như số dư tài khoản, mã sản phẩm, kết quả giao dịch và hướng dẫn sử dụng.

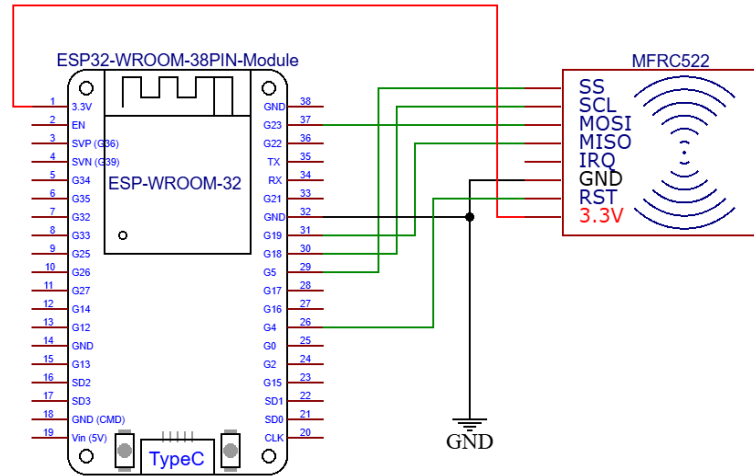
Khối âm báo sử dụng buzzer để phát ra tín hiệu âm thanh nhằm thông báo trạng thái như thao tác nhấn phím, lỗi thẻ hoặc thanh toán thành công.

Khi giao dịch được xác nhận, khối động cơ sẽ được kích hoạt để thực hiện cơ chế trả hàng cho người dùng.

Cuối cùng, khối nguồn đóng vai trò cung cấp điện áp ổn định 5V cho toàn hệ thống đảm bảo hoạt động liên tục và an toàn.

3.1.3 Thiết kế từng khối

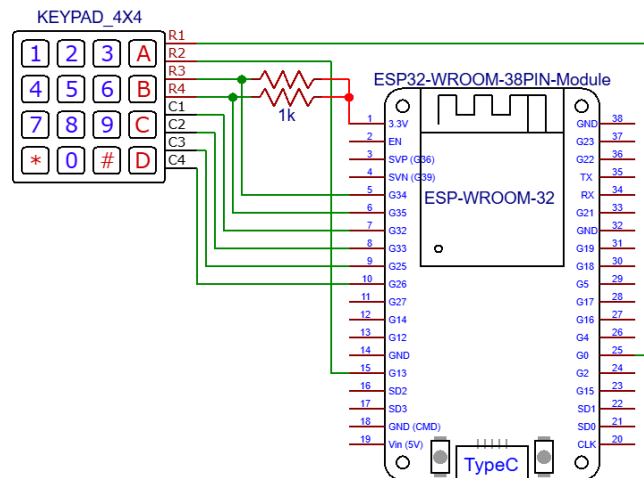
3.1.3.1 Khối RFID



Hình 5: Sơ đồ nguyên lý kết nối module RFID MFRC522 với ESP32

Khối RFID bao gồm đầu đọc thẻ MFRC522 và thẻ MIFARE hoạt động ở mức điện áp 3.3V, dòng hoạt động thực tế 17mA, hỗ trợ chuẩn giao tiếp SPI với khoảng cách đọc thẻ từ 0 đến 5cm. Khối này thực hiện chức năng đọc thông tin từ thẻ RFID khi người dùng đưa thẻ vào vùng quét. Giao tiếp với vi điều khiển ESP32 thông qua chuẩn SPI gồm các chân: SCK nối với G18, MISO nối với G19, MOSI nối với G23, SS nối với G5, RST nối với G22. Nguồn 3.3V lấy trực tiếp từ ESP32 và GND được nối chung mass với hệ thống. chân IRQ không sử dụng.

3.1.3.2 Khối nhập lệnh



Hình 6: Sơ đồ nguyên lý kết nối Keypad 4x4 với ESP32

Theo nguyên lý hoạt động, chân R1, R2, R3, R4 được nối với các chân GPIO ở chế độ INPUT_PULLUP để đảm bảo mức logic ổn định khi không có tác động từ người dùng. Tuy nhiên, hai chân GPIO34 và GPIO35 của ESP32 không hỗ trợ chế độ PULLUP nội. Do đó, ta cần gắn điện trở kéo lên ngoài để đảm bảo hoạt động đúng của hệ thống.

Vì các chân GPIO của ESP32 chỉ cho phép dòng tối đa khoảng 40mA, và để đảm bảo an toàn cũng như giảm tiêu thụ năng lượng, dòng qua mỗi chân được giới hạn ở mức dưới 20mA. Từ đó, ta tính được điện trở kéo lên tối thiểu như sau:

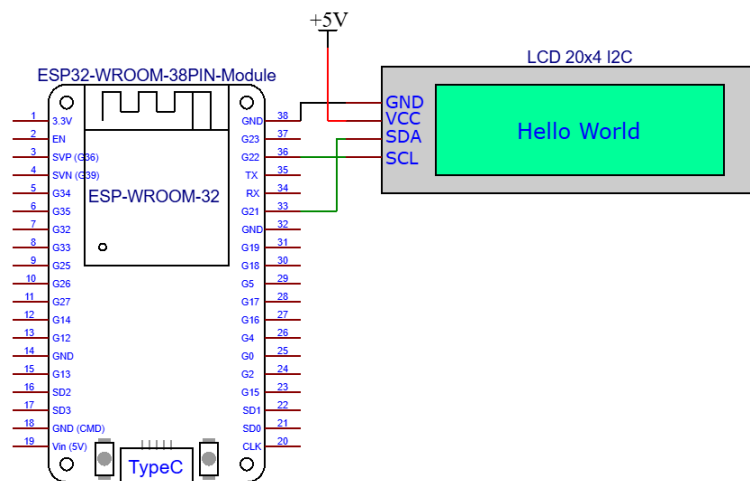
$$R = \frac{3.3V}{20mA} = 165\Omega$$

Để đảm bảo dòng qua GPIO nhỏ hơn 20mA, ta chọn điện trở lớn hơn hoặc bằng 165Ω. Trong thực tế, ta chọn giá trị tiêu chuẩn 1kΩ, khi đó dòng kéo lên là:

$$I = \frac{3.3V}{1000\Omega} = 3.3mA$$

Dòng này phù hợp với ngưỡng cho phép và đảm bảo hoạt động ổn định của nút nhấn.

3.1.3.3 Khối hiển thị

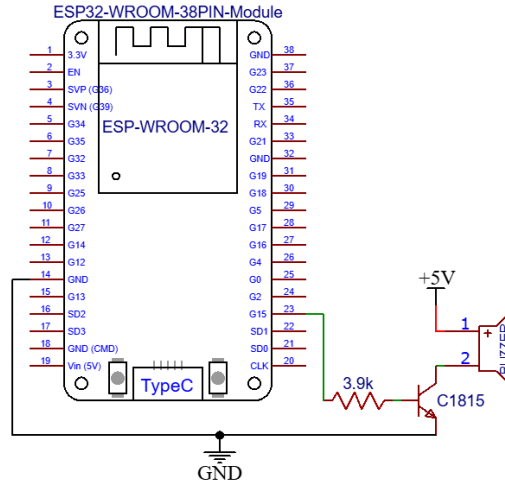


Hình 7: Sơ đồ nguyên lý kết nối LCD 20x4 I2C với ESP32

Khối hiển thị sử dụng màn hình LCD 20x4 tích hợp giao tiếp I2C, có chức năng hiển thị các thông tin quan trọng trong quá trình vận hành hệ thống như: trạng thái giao dịch, số dư tài khoản, hướng dẫn sử dụng hoặc thông báo lỗi.

Màn hình được cấp nguồn 5V, dòng hoạt động thực tế là 46.5mA và nối mass chung với hệ thống. Giao tiếp I2C giữa LCD và ESP32 được thiết lập qua hai chân: SDA nối với G21, SCL nối với G22.

3.1.3.4 Khối âm báo



Hình 8: Sơ đồ nguyên lý kết nối Buzzer với ESP32

Mạch điều khiển buzzer sử dụng transistor C1815 hoạt động dựa trên nguyên lý chuyển mạch của transistor NPN, chỉ dẫn khi điện áp chân Base lớn hơn chân Emitter khoảng 0.7V. Buzzer được mắc giữa nguồn 5V và chân Collector (C) của transistor, trong khi chân Emitter (E) được nối trực tiếp xuống GND. Tín hiệu điều khiển từ vi điều khiển ESP32 (chân G15) được đưa qua điện trở 3.9kΩ đến chân Base (B) của transistor để giới hạn dòng kích. Khi GPIO15 được đưa lên mức cao (HIGH), dòng điện từ GPIO sẽ chạy qua điện trở vào Base, làm cho transistor dẫn dòng từ Collector xuống Emitter, tức là kéo chân âm của buzzer xuống GND, tạo thành một mạch kín với nguồn 5V, khiến buzzer phát âm thanh. Ngược lại, khi GPIO15 ở mức thấp (LOW), transistor không dẫn, dòng không chạy qua buzzer nên buzzer không kêu. Cách điều khiển này giúp vi điều khiển không phải trực tiếp cấp dòng cho buzzer, bảo vệ chân GPIO khỏi quá tải và đảm bảo buzzer hoạt động ổn định với nguồn nuôi riêng biệt.

Để transistor hoạt động trong vùng bão hòa, dòng vào chân base phải lớn hơn hoặc bằng giá trị tối thiểu được xác định bởi:

$$I_B \geq \frac{I_C}{hFE}$$

Theo datasheet của transistor C1815 loại GR, hệ số khuếch đại dòng hFE nằm trong khoảng 200–400, do đó để đảm bảo chắc chắn, ta chọn giá trị thấp nhất là 200. Với dòng cực đại qua buzzer là $I_C = 30mA$, ta tính được dòng base tối thiểu:

$$I_{B_{min}} = \frac{30mA}{200} = 0.15mA$$

Từ đó, điện trở R_B lớn nhất tương ứng là:

$$R_{B_{max}} = \frac{3.3V - 0.7V}{0.15mA} \approx 17.3k\Omega$$

Mặt khác, GPIO của ESP32 chỉ chịu được dòng tối đa 40mA, và để đảm bảo an toàn, ta giới hạn dòng điều khiển nhỏ hơn 20mA. Khi đó, điện trở R_B tối thiểu là:

$$R_{B_{min}} = \frac{3.3V - 0.7V}{20mA} = 130\Omega$$

Ta lựa chọn giá trị điện trở thực tế là $R_B = 3.9k\Omega$, thỏa mãn điều kiện:

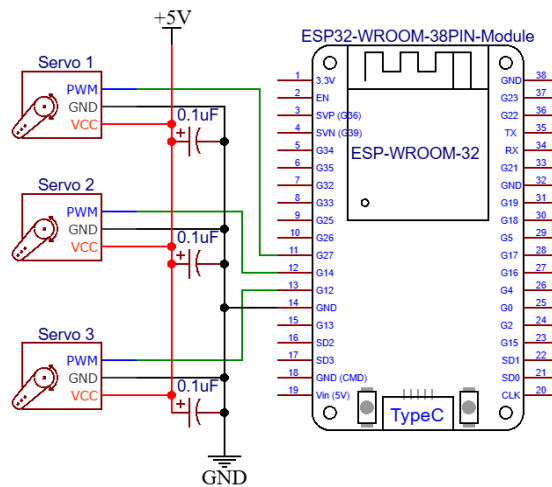
$$R_{B_{min}} < R_B < R_{B_{max}}$$

Với giá trị này, dòng điều khiển vào chân base là:

$$I_B = \frac{3.3V - 0.7V}{3.9k\Omega} \approx 0.67mA$$

Giá trị này đủ lớn để transistor dẫn bão hòa và vẫn nằm trong giới hạn dòng cho phép của chân GPIO.

3.1.3.5 Khối truyền động



Hình 9: Sơ đồ nguyên lý kết nối khối truyền động với ESP32

Khối truyền động của hệ thống sử dụng ba động cơ servo, được điều khiển trực tiếp từ vi điều khiển ESP32. Mỗi servo đảm nhận nhiệm vụ cơ học thực hiện chuyển động chính xác trong quá trình trả hàng.

Ba servo được cấp nguồn 5V chung và nối mass với hệ thống. Tín hiệu điều khiển PWM được lấy từ các chân:

Servo 1: G27.

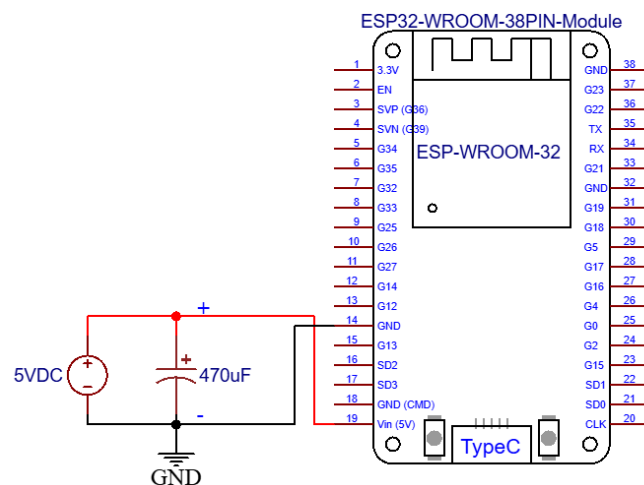
Servo 2: G14.

Servo 3: G12.

Để giảm nhiễu và ổn định điện áp khi servo hoạt động, mỗi servo được mắc song song với một tụ $0.1\mu\text{F}$ giữa nguồn VCC và GND. Thiết kế này đảm bảo servo hoạt động ổn định, tránh sụt áp đột ngột khi các động cơ hoạt động đồng thời.

Dòng trung bình của 3 động cơ servo khi không hoạt động khoảng $I_t = 1.83\text{mA}$ và khi thực hiện chuyển động là khoảng $I_d = 195\text{mA}$.

3.1.3.6 Khối nguồn

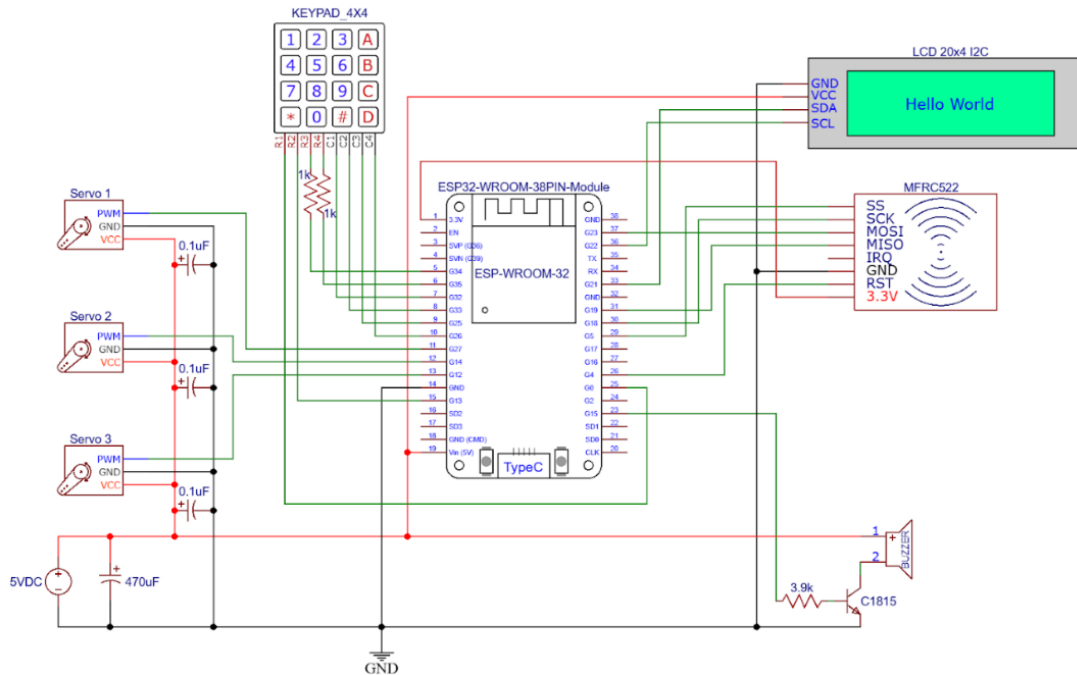


Hình 10: Sơ đồ nguyên lý khối nguồn

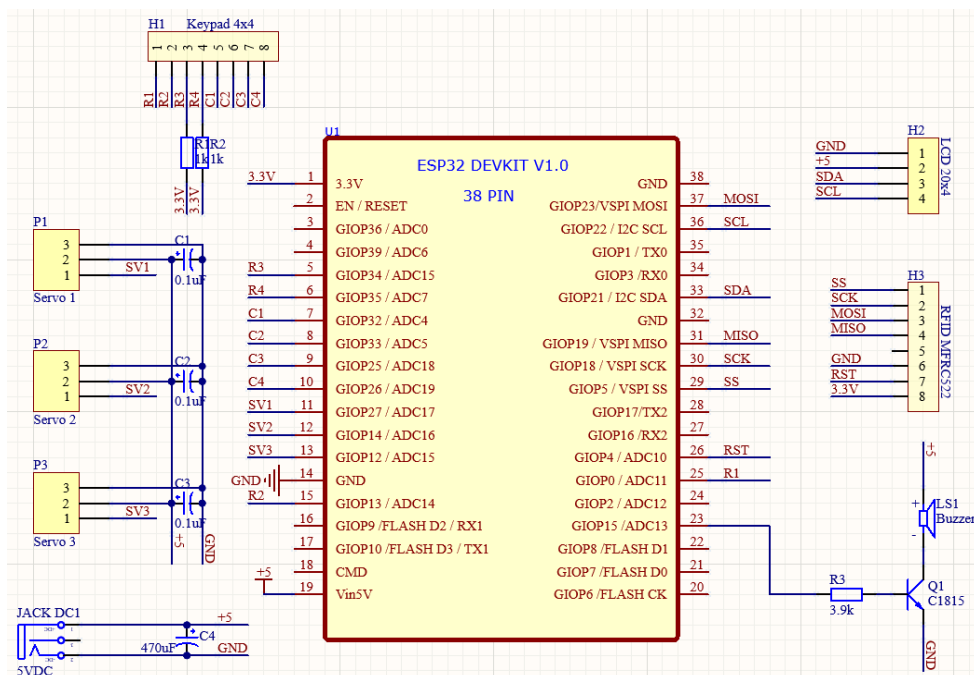
Khối nguồn sử dụng adapter 5V/3A để cấp điện cho toàn bộ hệ thống thông qua jack DC cái. Nguồn 5V này đảm bảo đủ dòng cung cấp cho vi điều khiển ESP32, các module ngoại vi và các động cơ servo hoạt động ổn định. Nguồn 5V được cấp qua chân Vin của ESP32, sau đó qua bộ ổn áp tích hợp để tạo ra các mức điện áp phù hợp như 3.3V cho các module khác trong hệ thống.

Để hạn chế nhiễu và ổn định dòng điện khi có tải thay đổi, một tụ 470 μ F được mắc song song giữa nguồn và mass. Tụ này giúp giảm thiểu tình trạng sụt áp và tăng độ ổn định cho toàn hệ thống, đặc biệt khi các thiết bị như động cơ servo hoặc module RFID tiêu thụ dòng lớn đột ngột.

3.1.4 Sơ đồ nguyên lý toàn mạch



Hình 11: Sơ đồ nguyên lý toàn hệ thống



Hình 12: Sơ đồ nguyên lý được thiết kế trên phần mềm Altium

Giải thích sơ đồ nguyên lý:

Module RFID MFRC522 kết nối với vi điều khiển thông qua chuẩn SPI được tích hợp sẵn trên kit phát triển, nguồn cấp cho đầu đọc RFID là 3.3V được cấp từ ESP32.

Màn hình hiển thị LCD 20x4 kết nối với ESP32 thông qua chuẩn I2C từ module I2C được tích hợp sẵn, nguồn cấp 5V được lấy từ chân Vin (5V) của ESP32.

Bàn phím ma trận Keypad 4x4 kết nối với ESP32 thông qua các GPIO, đối với hàng sẽ được kéo lên mức cao G0 (R1), G13 (R2), 2 chân input only G34 (R3), G35 (R4) cần được nối với trở kéo lên 1k Ω . Đối với cột sẽ được kéo xuống mức thấp G32 (C1), G33 (C2), G25 (C3), G26 (C4) để nhận lệnh từ người dùng.

Buzzer được nối với transistor C1815 theo nguyên lý chuyển mạch. Khi chân G15 của ESP32 ở mức cao, dòng điện đi qua điện trở 3.9k Ω kích hoạt chân Base transistor, làm transistor dẫn và buzzer kêu. Khi G15 ở mức thấp, transistor ngắt, buzzer ngừng kêu. Cách này giúp bảo vệ vi điều khiển giúp kéo dòng từ nguồn 5V thay vì bắt ESP32 cấp dòng trực tiếp.

Có 3 động cơ servo SG90 360 độ được cấp nguồn 5V từ khối nguồn tương ứng với 3 ô sản phẩm của máy bán hàng mini, mỗi servo được mắc song song với tụ 0.1 μ F để lọc nhiễu, giúp ổn định điện áp, tránh ảnh hưởng đến tín hiệu điều khiển khi dòng cấp thay đổi đột ngột. Các chân G27, G14, G12 của ESP32 lần lượt được kết nối với chân tín hiệu PWM của servo 1, servo 2 và servo 3.

Nguồn cấp cho toàn hệ thống sử dụng adapter 5V/3A.

3.2 THIẾT KẾ PHẦN MỀM

3.2.1 Chức năng hoạt động của phần mềm

Phần mềm của hệ thống được phát triển trên nền tảng vi điều khiển ESP32. Mục tiêu của phần mềm là điều phối toàn bộ hoạt động của hệ thống, bao gồm giao tiếp với người dùng, xử lý dữ liệu giao dịch và kết nối đến dịch vụ lưu trữ từ xa. Phần mềm vận hành theo mô hình lặp vô tận (vòng lặp loop()), trong đó liên tục

kiểm tra các sự kiện đầu vào, phản hồi và thực thi các tác vụ tương ứng một cách tự động.

Ngay khi khởi động, phần mềm tiến hành thiết lập kết nối WiFi để đảm bảo có thể truyền và nhận dữ liệu từ Google Apps Script. Đồng thời, các thiết bị ngoại vi như màn hình LCD, bàn phím keypad, module RFID, servo và buzzer cũng được khởi tạo để sẵn sàng phục vụ người dùng. Sau khi kết nối thành công, màn hình LCD sẽ hiển thị lời chào mừng và yêu cầu người dùng quét thẻ RFID.

Khi một thẻ RFID được đưa vào gần đầu đọc, phần mềm sẽ đọc và xử lý UID của thẻ, sau đó gửi thông tin này lên Google Apps Script thông qua một yêu cầu HTTP. Script này sẽ truy xuất dữ liệu người dùng từ Google Sheets, sau đó phản hồi lại tên và số dư của người dùng. Phần mềm tiếp nhận dữ liệu và hiển thị thông tin này trên màn hình LCD.

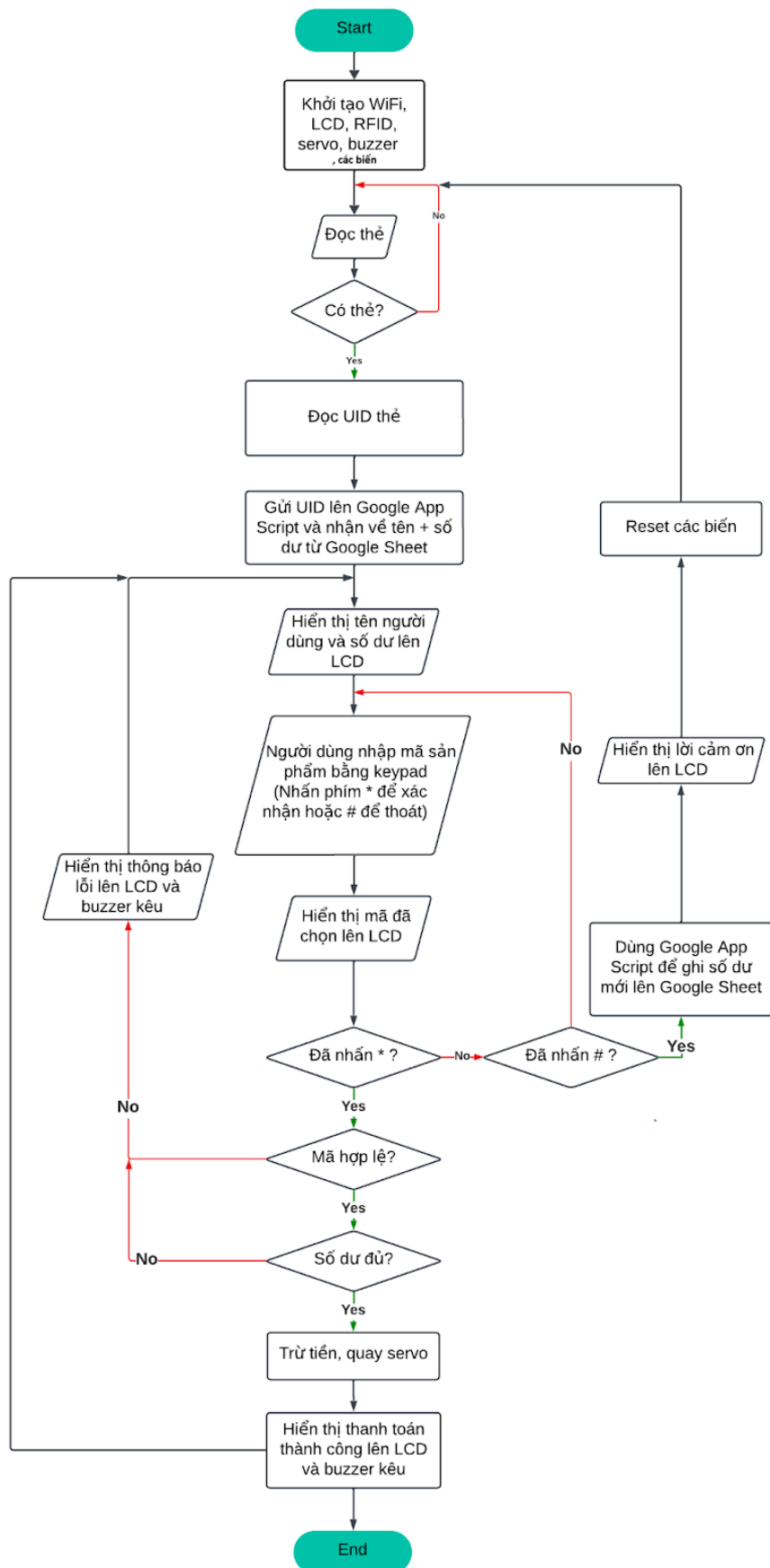
Sau khi nhận diện người dùng, phần mềm chuyển sang chế độ chờ thao tác. Người dùng có thể sử dụng bàn phím để nhập mã sản phẩm cần mua. Khi nhấn phím *, phần mềm kiểm tra mã sản phẩm có hợp lệ và số dư có đủ để thanh toán hay không. Nếu cả hai điều kiện đều được đáp ứng, phần mềm sẽ trừ số dư tạm thời, phát tín hiệu xác nhận bằng buzzer, điều khiển servo để trả sản phẩm, và cập nhật lại giao diện hiển thị.

Ngược lại, nếu mã không hợp lệ hoặc số dư không đủ, phần mềm sẽ phát cảnh báo và hiển thị thông báo lỗi. Người dùng có thể tiếp tục giao dịch hoặc nhấn phím # để kết thúc. Khi người dùng chọn kết thúc giao dịch, phần mềm sẽ gửi yêu cầu HTTP POST đến Google Apps Script để cập nhật số dư mới của người dùng lên Google Sheets. Cuối cùng, phần mềm sẽ xóa các biến tạm thời, reset trạng thái và quay trở về chế độ chờ quét thẻ ban đầu.

Nhìn chung, phần mềm đóng vai trò trung tâm điều khiển logic hệ thống, kết nối hiệu quả giữa phần cứng và dịch vụ lưu trữ từ xa, đồng thời mang đến trải nghiệm tương tác đơn giản, trực quan cho người dùng.

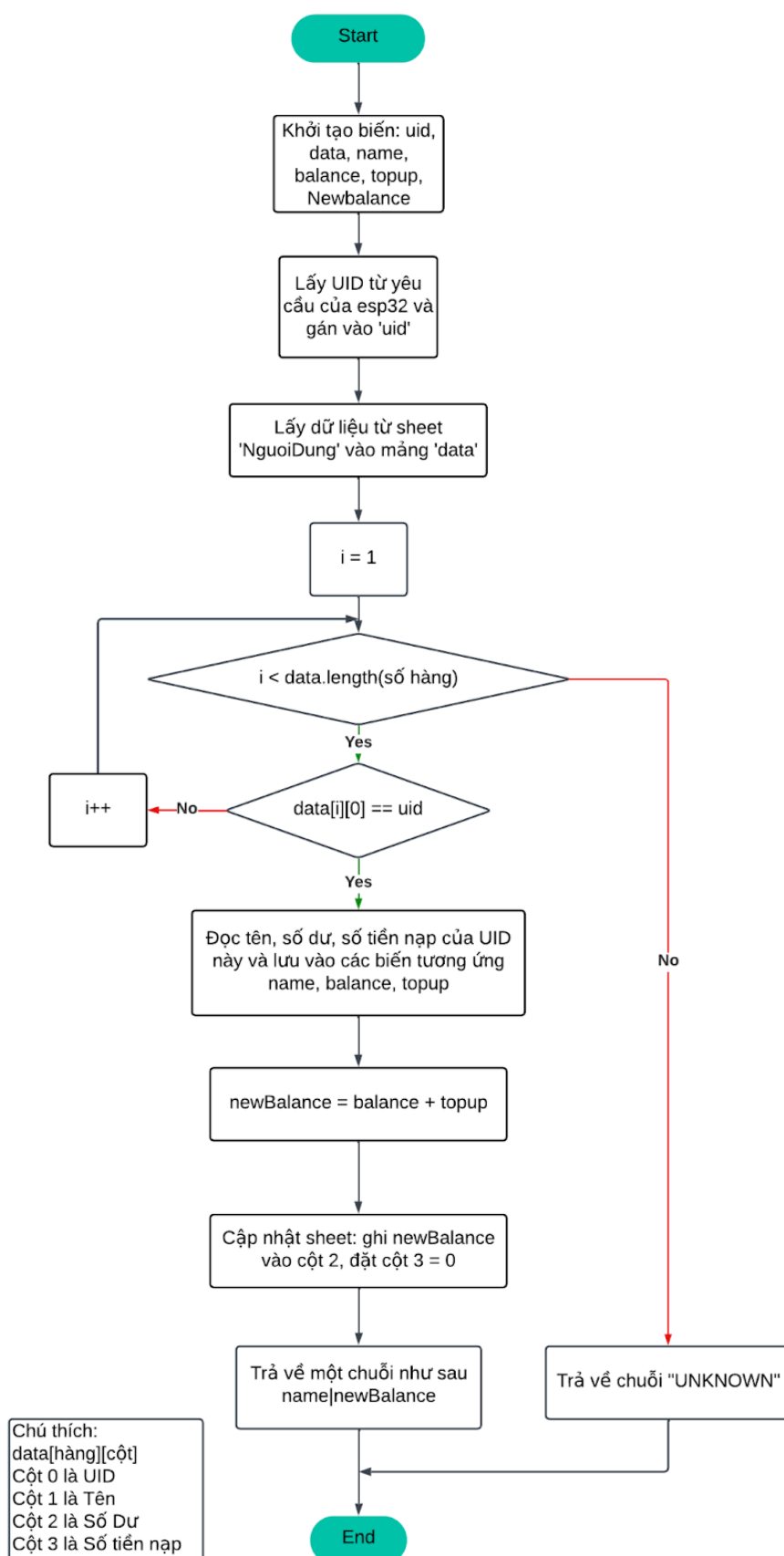
3.2.2 Lưu đồ hoạt động

Lưu đồ tổng quát:



Hình 13: Lưu đồ tổng quát

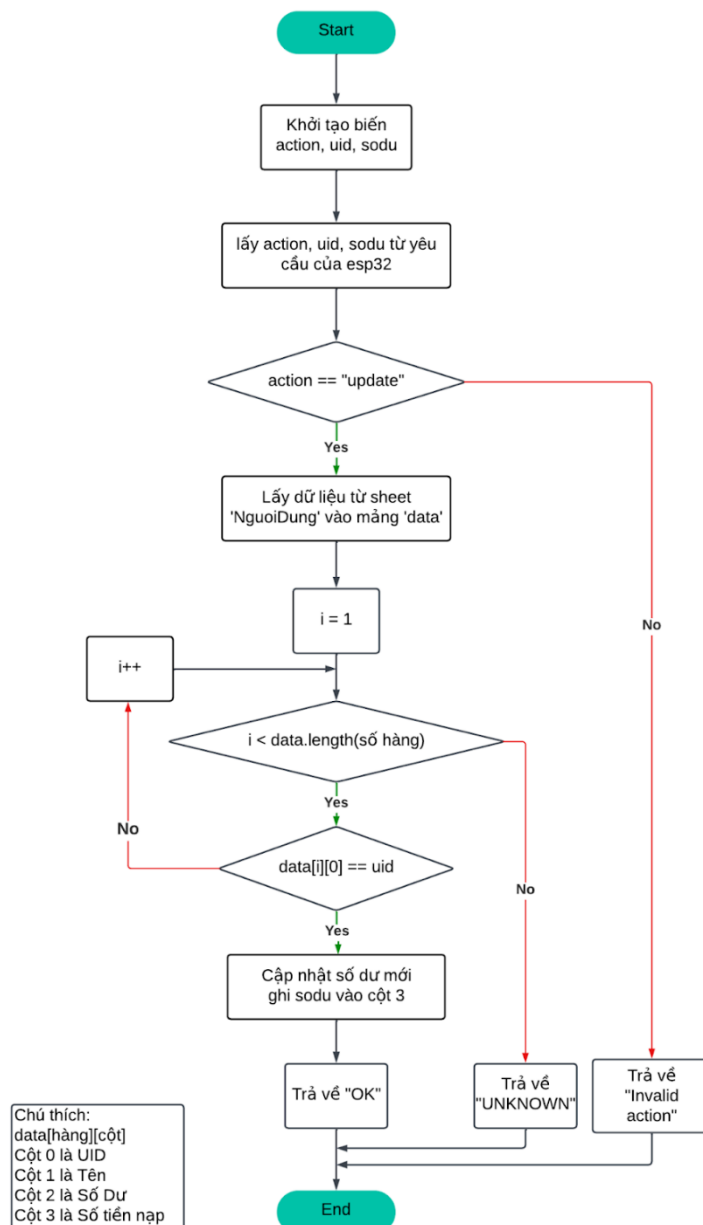
Lưu đồ lấy thông tin Sheet thông qua doGet(e) Google Apps Script:



Hình 14: Lưu đồ lấy thông tin Sheet

Hàm doGet(e) được sử dụng để xử lý yêu cầu truy vấn thông tin người dùng từ hệ thống. Khi hệ thống gửi UID thông qua URL, hàm sẽ tìm kiếm trong sheet "NguoiDung" để xác định người dùng tương ứng. Nếu tìm thấy, hàm sẽ cộng thêm số tiền nạp (nếu có), cập nhật số dư mới vào bảng tính và trả về tên cùng với số dư hiện tại dưới dạng chuỗi phân tách bằng dấu "|". Trường hợp không tìm thấy UID, hàm sẽ trả về "UNKNOWN". Cơ chế này hỗ trợ hệ thống truy xuất và nạp tiền vào tài khoản một cách tự động.

Lưu đồ cập nhật số dư sau giao dịch lên Sheet thông qua doPost(e) Google Apps Script:



Hình 15: Lưu đồ cập nhật số dư sau giao dịch lên Sheet

Hàm doPost(e) đảm nhận việc xử lý yêu cầu cập nhật số dư khi giao dịch thanh toán diễn ra. Khi hệ thống gửi một yêu cầu HTTP POST kèm theo tham số action == "update", UID người dùng và số dư mới, hàm sẽ tìm dòng dữ liệu tương ứng và cập nhật lại số dư trong cột thứ ba của bảng "NguoiDung". Nếu không tìm thấy người dùng, hàm trả về "UNKNOWN". Cách làm này giúp đồng bộ hóa dữ liệu người dùng giữa hệ thống bán hàng và Google Sheets một cách chính xác và hiệu quả.

Chương 4

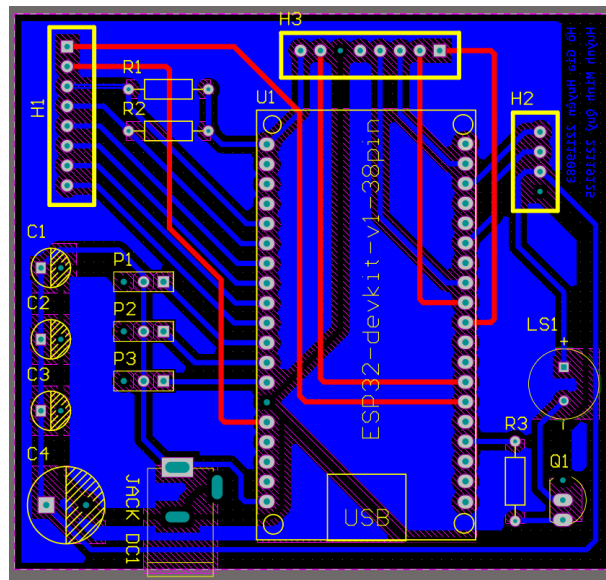
KẾT QUẢ

4.1 KẾT QUẢ MÔ HÌNH THI CÔNG

4.1.1 Bo mạch phần cứng

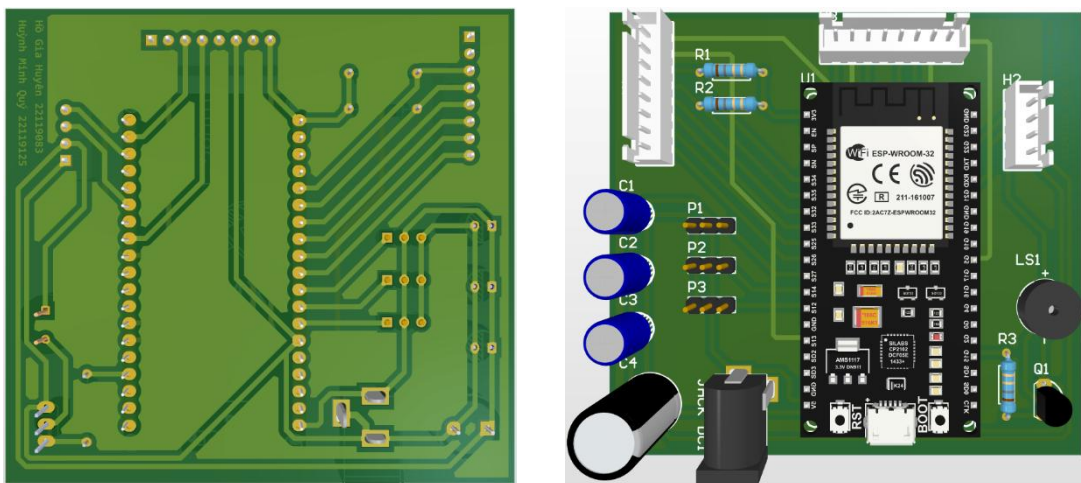
Mạch in được thiết kế trên phần mềm Altium Designer.

PCB layout của toàn mạch được thiết kế trên phần mềm:



Hình 16: PCB layout được thiết kế trên Altium Design

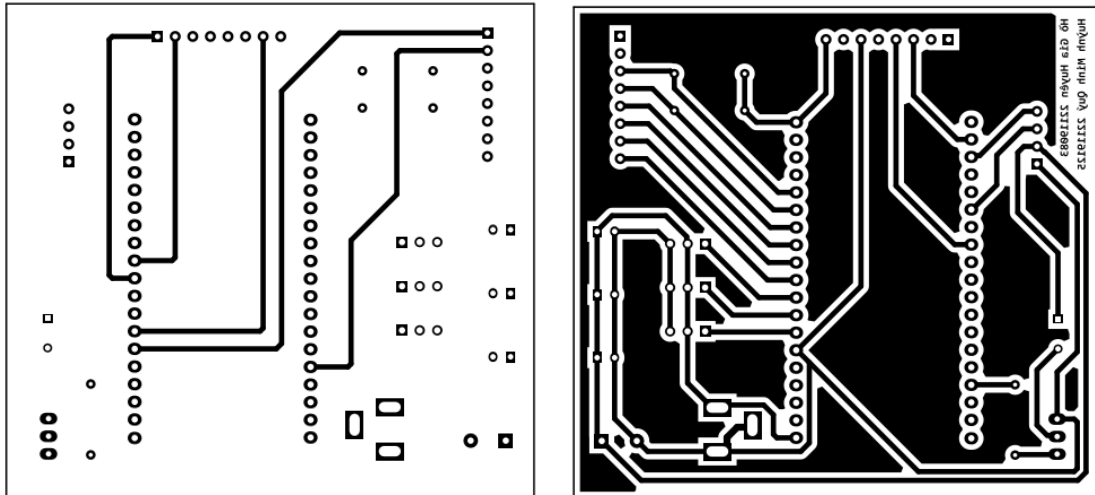
Sơ đồ mô phỏng bố trí linh kiện và chân nối của toàn mạch trên phần mềm:



Hình 17: Sơ đồ chân nối và mạch 3D mô phỏng trên Altium

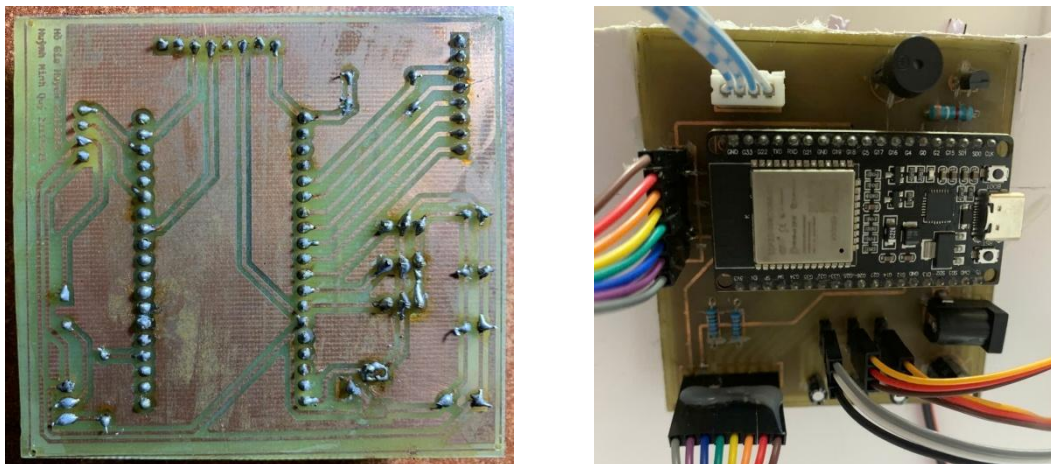
Board mạch được thiết kế xoay quanh vi điều khiển ESP32, với các linh kiện ngoại vi như servo, RFID, LCD, bàn phím và buzzer được kết nối thông qua các header rời. Việc sử dụng các header giúp thi công dễ dàng, thuận tiện trong lắp ráp, bảo trì và thay thế linh kiện khi cần thiết.

Sơ đồ thi công mạch in lớp trên và lớp dưới của toàn mạch:



Hình 18: Sơ đồ mạch in lớp trên và lớp dưới của toàn mạch

Kết quả thi công board mạch thực tế của hệ thống máy bán hàng tự động mini:



Hình 19: Board mạch thi công thực tế lớp trên và lớp dưới của toàn mạch

4.1.2 Google sheet quản lý khách hàng phần mềm

Danh sách khách hàng			
UID	Ten	SoDu	NapTien
23C3BB90	An	80	0
938E8790	Hung	25	0
C37B9990	Huy	54	0
07593903	Huyen	49268	0
13AABD90	Quy	401	0

Hình 20: Giao diện quản lý thông tin khách hàng bằng Google sheet

Trong đồ án này, Google Sheet được sử dụng như một cơ sở dữ liệu trực tuyến để lưu trữ và quản lý thông tin khách hàng, bao gồm UID, tên người dùng, số dư (SoDu) và số tiền chờ nạp (NapTien).

4.1.3 Mô hình sản phẩm



Hình 21: Mô hình hệ thống máy bán hàng tự động mini

Mặt trước của mô hình được trang bị giao diện người dùng gồm màn hình LCD hiển thị thông tin, bàn phím số để nhập lựa chọn sản phẩm, và một đầu đọc thẻ RFID để thực hiện thanh toán không tiếp xúc. Mô hình hỗ trợ ba loại sản phẩm (SP1, SP2, SP3) với giá lần lượt là \$1, \$2 và \$3, được phân loại và hiển thị rõ ràng.

Bên trong mô hình bao gồm các kệ chứa sản phẩm được tích hợp lò xo đẩy và động cơ servo, giúp tự động đưa sản phẩm ra khe nhận hàng sau khi thanh toán thành công. Toàn bộ hệ thống được điều khiển bởi board mạch sử dụng vi điều khiển ESP32.

4.2 HOẠT ĐỘNG CỦA HỆ THỐNG

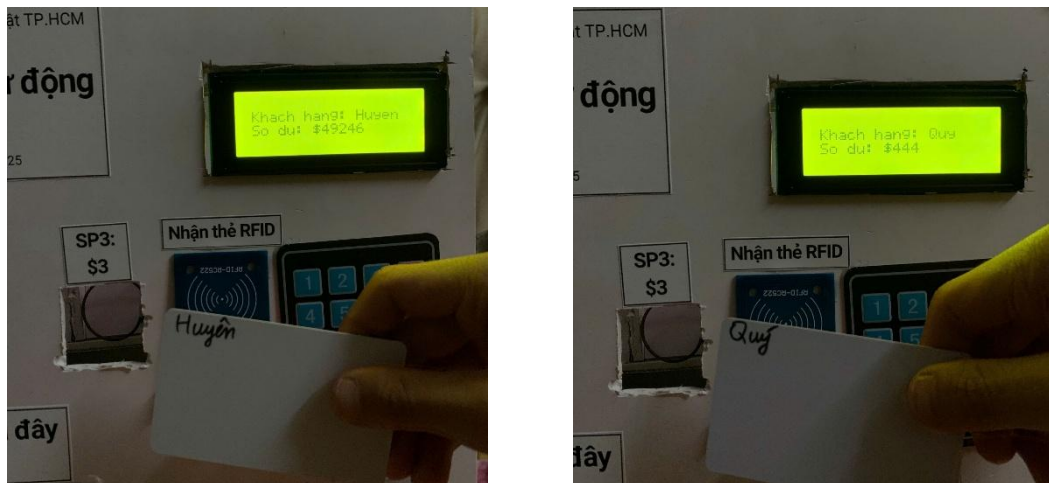
4.2.1 Khởi động hệ thống



Hình 22: Hệ thống khởi động

Hệ thống khởi động khi được cấp nguồn từ adapter 5V trực tiếp vào jack DC. Khi mới khởi động hệ thống sẽ tiến hành kết nối WiFi và hiển thị thông tin như trên hình 23.

4.2.2 Nhận diện thẻ RFID



Hình 23: Hệ thống nhận diện thẻ RFID thành công

Khi đưa một thẻ RFID lại gần module RFID, hệ thống sẽ truy xuất thông tin khách hàng có mã UID trùng khớp từ Google Sheet và hiển thị thông tin tên, số dư lên màn hình như hình 24. Nếu mã UID không tồn tại trong cơ sở dữ liệu thì hệ thống sẽ thông báo như hình 25.

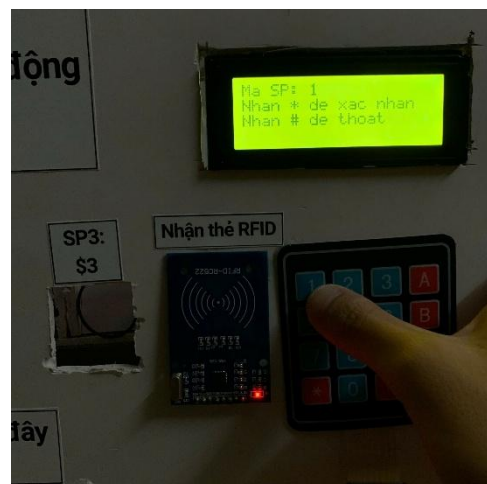


Hình 24: Hệ thống nhận diện thẻ RFID thất bại

4.2.3 Chọn sản phẩm và thanh toán

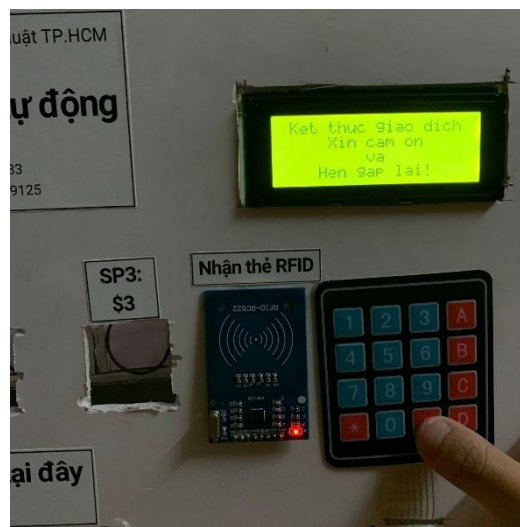
Quy trình chọn sản phẩm và thanh toán sau khi RFID nhận thành công:

Chọn sản phẩm muốn mua bằng phím “1”, “2”, “3”. Sau đó, nhấn phím “*” nếu muốn thanh toán hoặc phím “#” để thoát giao dịch, chọn sản phẩm khác. Khi thanh toán thành công màn hình sẽ hiển thị “Thanh toán OK” và sản phẩm tương ứng sẽ được đẩy xuống khu vực nhận hàng như hướng dẫn. Sau đó, khách hàng có thể tiếp tục mua thêm sản phẩm (nếu muốn). Khi hoàn thành việc mua hàng và thoát ra khỏi giao dịch thì số dư mới của khách hàng sẽ được cập nhật lên google sheet.





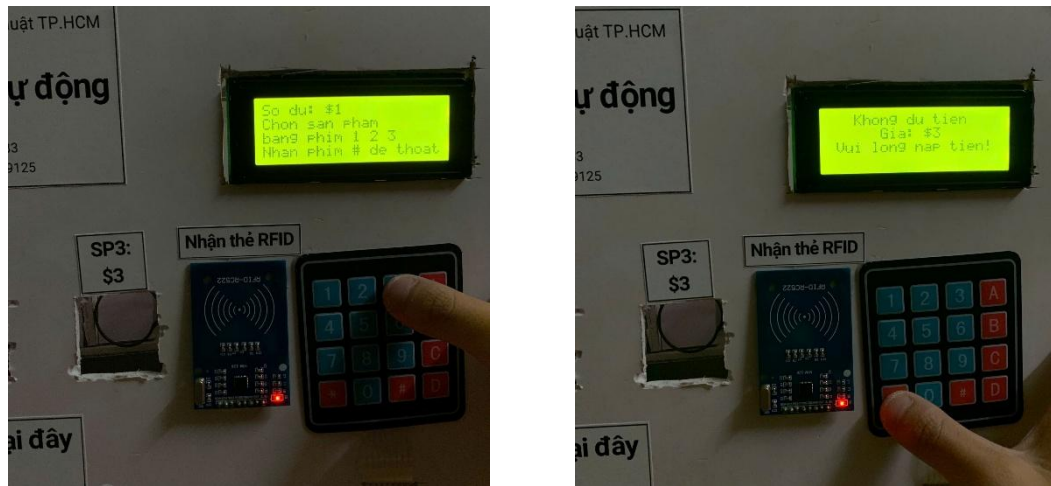
Hình 25: Quy trình chọn sản phẩm và thanh toán thành công



Hình 26: Kết thúc giao dịch

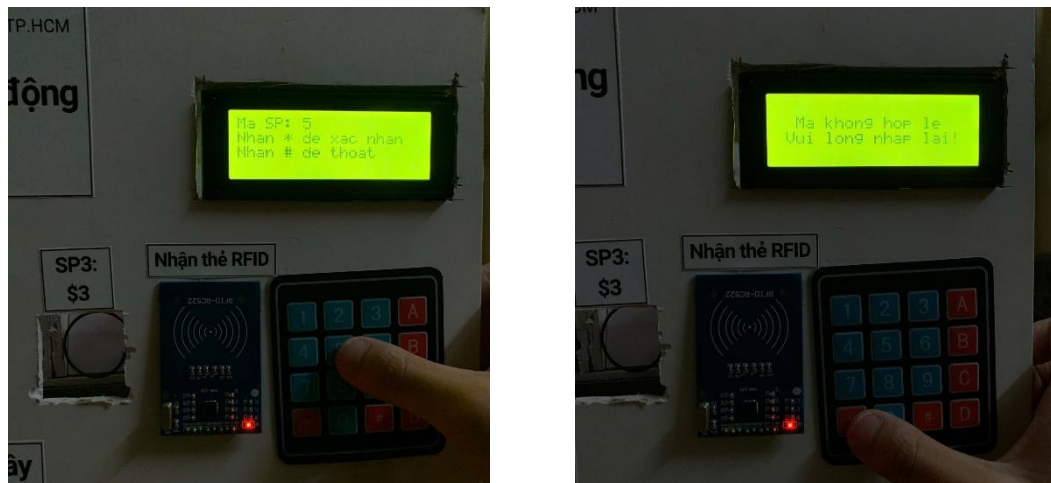
4.2.4 Các trường hợp lỗi giao dịch

Trường hợp số dư không đủ để mua sản phẩm: Khi số dư của khách hàng nhỏ hơn giá tiền của sản phẩm muốn mua thì hệ thống sẽ thông báo như hình và động cơ sẽ không đẩy hàng ra.



Hình 27: Trường hợp số dư không đủ để mua sản phẩm

Trường hợp chọn mã sản phẩm không hợp lệ: Khi nhập mã sản phẩm khác ba mã “1”, “2”, “3” thì hệ thống sẽ yêu cầu nhập lại.

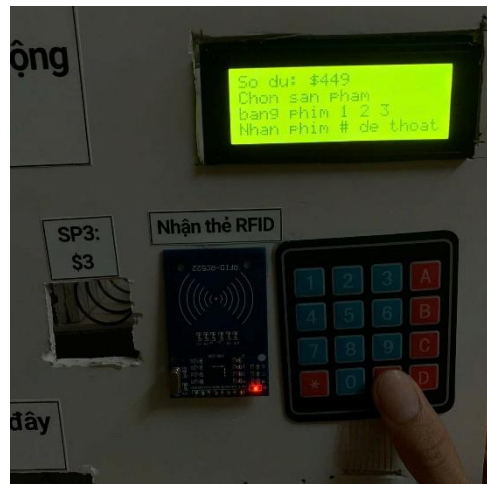


Hình 28: Trường hợp chọn mã sản phẩm không hợp lệ:

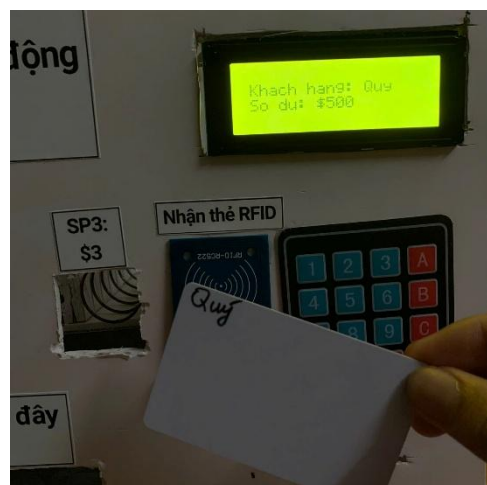
4.2.5 Nạp tiền cho khách hàng

Nhập số tiền muốn nạp vào cột “NạpTien” trên google sheet, hệ thống sẽ cần phải thoát ra trước khi số dư mới được cập nhật. Sau khi quét thẻ RFID của khách hàng cần nạp tiền lại thì số dư mới sẽ được cập nhật và cột “NạpTien” trả về giá trị 0.

Danh sách khách hàng				
UID	Ten	SoDu	NapTien	
23C3BB90	An	80	0	
938E8790	Hung	1	0	
C37B9990	Huy	54	0	
07593903	Huyen	48921	0	
13AABD90	Quy	449	51	



Danh sách khách hàng				
UID	Ten	SoDu	NapTien	
23C3BB90	An	80	0	
938E8790	Hung	1	0	
C37B9990	Huy	54	0	
07593903	Huyen	48921	0	
13AABD90	Quy	500	0	



Hình 29: Các bước nạp tiền

4.3 THÔNG SỐ KẾT QUẢ VÀ THỰC NGHIỆM

4.3.1 Công suất tiêu thụ toàn hệ thống

Công suất hệ thống ở trạng thái nhàn rỗi:

$$U_{IDLE} = 5.08V$$

$$I_{IDLE} = 0.18 A$$

$$P_{IDLE} = U \times I = 0.91W$$

Công suất hệ thống khi ở trạng thái truy xuất thông tin:

$$U_{truyxuat} = 5.08V$$

$$I_{truyxuat} = 0.26A$$

$$P_{truyxuat} = U \times I = 1.32W$$

Công suất hệ thống khi ở trạng thái quay servo trả hàng:

$$U_{quay} = 5.08V$$

$$I_{\text{quay}} = 0.41\text{A}$$

$$P_{\text{quay}} = U \times I = 2.08\text{W}$$

Hệ thống có công suất lớn nhất khi quay servo. Nếu hệ thống ở mức công suất lớn nhất trong 1 tháng (giả sử 30 ngày, tương đương 720 giờ), năng lượng tiêu thụ của toàn bộ thiết bị sẽ được tính như sau:

$$E_{\text{month}} = P_{\text{quay}} \times t = 2.08\text{W} \times 720\text{h} = 1497.6 \text{ Wh} = 1.4976 \text{ kWh}.$$

Như vậy hệ thống vẫn chỉ tiêu thụ khoảng **1.498 kWh** điện mỗi tháng. Từ kết quả này cho thấy dù công suất khi quay servo là trạng thái tiêu thụ lớn nhất so với các chế độ khác (idle hoặc truy xuất thông tin), mức tiêu thụ thực tế trong một tháng vẫn rất khiêm tốn. Trong vận hành thực tế, servo chỉ kích hoạt khi có giao dịch thành công, do đó xác suất hoạt động liên tục là rất thấp. Kết quả đo lường cho thấy, tổng lượng điện năng tiêu thụ của hệ thống khi kết hợp cả ba trạng thái idle, truy xuất và quay servo trong điều kiện vận hành bình thường sẽ rất nhỏ, thể hiện tính kinh tế và hiệu quả năng lượng của thiết kế.

4.3.2 Tỷ lệ lỗi khi trả hàng

Bảng 1: Bảng thống kê tỷ lệ lỗi

Ô hàng	Số lần thử	Số lần lỗi	Tỉ lệ thành công	Tỉ lệ lỗi
1	50	0	100%	0%
2	50	0	100%	0%
3	50	3	94%	6%
Tổng	150	3	98%	2%

Kết quả thử nghiệm cơ cấu trả hàng được tổng hợp trong bảng cho thấy hệ thống duy trì độ tin cậy rất cao, với tỷ lệ thành công chung đạt 98% trên tổng số 150 lần thử. Cụ thể, hai ô hàng đầu (ô 1 và ô 2) vận hành hoàn toàn chính xác, không phát sinh lỗi nào trong 50 lần thử mỗi ô, tương đương tỷ lệ thành công 100%. Chỉ riêng ô hàng số 3 ghi nhận 3 lần lỗi trên 50 lần thử, dẫn đến tỷ lệ thành công 94% và tỷ lệ thất bại 6% ở riêng ô này.

Việc ô số 3 có tỷ lệ lỗi cao hơn có thể xuất phát từ nguyên nhân: Lò xo đảm nhận việc đẩy hàng ra của ô thứ 3 bị cong một chút.

4.3.3 Truy xuất thông tin người dùng từ Google Sheet

Thử nghiệm độ trễ khi truy xuất thông tin:

Bảng 2: Bảng thống kê thời gian truy xuất thông tin khách hàng thứ 5

Lần	Thời gian truy xuất
1	5.41s
2	5.36s
3	5.11s
4	5.29s
5	5.64s
6	5.31s
7	5.38s
8	5.16s
9	5.21s
10	5.11s
Trung bình	5.298s

Thử nghiệm độ trễ truy xuất khách hàng thứ 100 (khi tăng số khách lên 100 người):

Bảng 3: Bảng thống kê thời gian truy xuất thông tin khách hàng thứ 100

Lần	Thời gian truy xuất
1	7.06s
2	5.36s
3	6.01s
4	5.63s
5	5.60s
6	5.13s
7	6.46s
8	5.63s
9	5.61s
10	5.76s
Trung bình	5.825s

Một danh sách nhỏ (5 người dùng) cho mức độ trễ ổn định quanh 5.3 giây. Khi tăng lên 100 người dùng, độ trễ tính trung bình tăng thêm khoảng 0.53 giây (tăng 10% so với ban đầu). Độ trễ khi truy xuất thông tin làm giảm trải nghiệm người dùng.

Nguyên do có độ trễ: Hàm doGet(e) trong Google Apps Script sử dụng getDataRange().getValues() để tải toàn bộ mảng dữ liệu vào bộ nhớ, sau đó lặp qua từng dòng để tìm UID khớp. Việc này khiến độ phức tạp xử lý trên server tăng dần theo số dòng, dẫn đến độ trễ lớn hơn. Tuy nhiên, phần lớn thời gian chờ vẫn nằm ở việc gửi/nhận HTTP qua mạng WiFi và thời gian thực thi ban đầu trên nền tảng Google, chứ không hoàn toàn do việc quét nhiều dòng. Độ trễ khi tăng khách hàng có tăng nhưng không quá nhiều, hệ thống có thể mở rộng và tăng số lượng khách nhất định.

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

5.1.1 Kết quả đạt được so với mục tiêu ban đầu đặt ra

Sau khi hoàn tất quá trình thiết kế, hiện thực và kiểm thử, hệ thống đã đáp ứng đầy đủ các mục tiêu ban đầu đã đề ra.

Đầu tiên, chức năng nhận diện thẻ RFID với module MFRC522 vận hành ổn định. Khi thẻ RFID được đưa vào vùng đọc, hệ thống luôn đọc đúng giá trị UID và gửi yêu cầu truy xuất dữ liệu đến Google Sheets qua Google Apps Script. Mặc dù quá trình truy xuất dữ liệu có độ trễ nhất định, nhất là khi số bản ghi tăng lên hoặc mạng Internet yếu nhưng độ trễ này vẫn nằm trong phạm vi chấp nhận được và không phát sinh lỗi trong suốt quá trình sử dụng.

Tiếp theo, việc chọn sản phẩm qua bàn phím ma trận 4x4 được thực hiện nhanh chóng và chính xác. Bất kể tốc độ nhập, hệ thống xử lý tín hiệu từ bàn phím tức thời, không xảy ra hiện tượng “đơ” hay nhập sai ký tự. Điều này khẳng định khả năng đọc phím và phản hồi logic của phần mềm đã được tối ưu và hoạt động ổn định.

Về cơ cấu trả hàng tự động, hệ thống sử dụng động cơ servo điều khiển các ngăn chứa hàng hóa. Khi thanh toán thành công, servo sẽ quay hai vòng để đẩy sản phẩm ra ngoài. Trong các bài thử nghiệm thực tế, tỷ lệ trả đúng sản phẩm đạt khoảng 98% qua 150 lần thử. Cơ cấu trả hàng chưa thực sự tốt do không điều khiển servo quay được đúng hai vòng (canh gần đúng hai vòng bằng cách cài đặt thời gian quay) và một phần là do lò xo thứ 3 bị cong.

Phần hiển thị giao dịch trên màn hình LCD 20x4 đã hoàn thiện đầy đủ các nội dung cần thiết: tên người dùng, số dư hiện có, mã sản phẩm được chọn và kết quả

thanh toán (thành công hay thất bại). Giao diện rõ ràng, mạch lạc, giúp người dùng dễ dàng theo dõi từng bước của quá trình mua hàng.

Cuối cùng, hệ thống âm báo qua buzzer cũng được tích hợp hiệu quả. Khi người dùng nhấn phím, quét thẻ hoặc hoàn tất giao dịch, buzzer phát tiếng bíp ngắn; ngược lại, khi gặp lỗi như số dư không đủ hoặc mã sản phẩm sai, buzzer phát tiếng dài để cảnh báo.

Tóm lại, hệ thống đã hoàn thành toàn diện sáu mục tiêu: (1) chọn sản phẩm qua bàn phím, (2) nhận diện và thanh toán bằng thẻ RFID, (3) trả hàng tự động bằng servo, (4) hiển thị thông tin giao dịch trên LCD, (5) âm báo khi có thao tác hoặc lỗi, và (6) quản lý dữ liệu người dùng, số dư và nạp tiền qua Google Sheets. Kết quả đạt được chứng tỏ tính khả thi và độ hoàn thiện về mặt chức năng của đề tài.

5.1.2 Ưu nhược điểm của hệ thống đã thiết kế

Ưu điểm: Hệ thống bán hàng tự thể hiện nhiều ưu điểm nổi bật. Trước hết, khả năng nhận diện thẻ RFID hoạt động ổn định (không gặp lỗi sau nhiều ngày thử nghiệm), đáp ứng nhanh chóng và chính xác trong việc xác thực người dùng. Việc sử dụng Google Sheets làm cơ sở dữ liệu trung tâm giúp giảm thiểu chi phí triển khai so với việc xây dựng server riêng, đồng thời cho phép quản lý, cập nhật số dư người dùng trực tuyến một cách thuận tiện. Cơ cấu trả hàng bằng động cơ servo tuy đơn giản nhưng vẫn đáp ứng được yêu cầu tự động hóa, tỷ lệ trả đúng sản phẩm đạt khoảng 98% qua 150 lần thử. Tổng công suất tiêu thụ khi vận hành ở trạng thái cao nhất (servo quay) chỉ khoảng 2,08 W, tương đương ~1,498 kWh một tháng, thể hiện tính tiết kiệm năng lượng.

Nhược điểm: hệ thống cũng tồn tại một số hạn chế cần khắc phục. Độ trễ khi truy xuất dữ liệu từ Google Sheets (khoảng 5–6 giây mỗi lần quét thẻ) có thể gây cảm giác chậm trễ, nhất là khi mạng Internet không ổn định hoặc số lượng bản ghi tăng lên. Cơ cấu servo hiện tại chưa điều khiển chính xác hai vòng quay (canh gần đúng hai vòng bằng cách cài đặt thời gian quay), và một số lỗi cơ học (như lò xo bị cong) đôi khi làm sản phẩm kẹt ở ngăn thứ ba, dẫn đến tỷ lệ thất bại 2% trong thử nghiệm. Việc bỏ qua xác thực SSL (`client.setInsecure()`) giúp ESP32 kết

nổi dễ dàng với Google Apps Script nhưng tiềm ẩn nguy cơ về bảo mật thông tin. Hệ thống phụ thuộc hoàn toàn vào kết nối Internet, nên trong trường hợp mất mạng hoặc Google Sheets gặp sự cố, chức năng xác thực và cập nhật số dư sẽ bị gián đoạn.

5.2 HƯỚNG PHÁT TRIỂN

Mặc dù hệ thống đã hoàn thành các chức năng cơ bản và vận hành ổn định, vẫn còn nhiều tiềm năng để cải tiến và mở rộng nhằm đáp ứng tốt hơn yêu cầu thực tế.

Trước hết, cần nâng cao độ chính xác trong cơ cấu trả hàng bằng cách gắn thêm encoder trên trục servo. Nhờ có phản hồi từ encoder, vi điều khiển có thể xác định chính xác góc quay và dừng động cơ đúng lúc, thay vì ước lượng theo thời gian như hiện tại. Bên cạnh đó, việc thiết kế lại cơ cấu lò xo và khay chứa cũng sẽ giúp đảm bảo sản phẩm luôn ở vị trí ổn định, giảm thiểu sai lệch khi trả hàng.

Về xử lý dữ liệu, Google Sheets có độ trễ khi số bản ghi lớn. Do đó, một hướng phát triển khả thi là chuyển sang sử dụng các nền tảng cơ sở dữ liệu như Firebase Realtime Database, Firestore hoặc Cloud SQL để tăng tốc độ truy xuất và khả năng mở rộng.

Ngoài ra, việc phát triển thêm ứng dụng web hoặc di động để người dùng theo dõi số dư, lịch sử giao dịch và nạp tiền cũng là cần thiết, nhằm nâng cao trải nghiệm người dùng thay vì chỉ thao tác qua Google Sheets. Đồng thời, hệ thống cũng nên mở rộng tích hợp thêm các phương thức thanh toán khác như mã QR, ví điện tử hoặc thẻ ngân hàng để đa dạng hóa lựa chọn cho người dùng.

Cuối cùng, việc tối ưu hóa mức tiêu thụ năng lượng, chẳng hạn tận dụng chế độ sleep của vi điều khiển khi hệ thống nhàn rỗi, sẽ giúp tăng thời gian hoạt động liên tục và phù hợp với các môi trường không có nguồn điện ổn định.

Những cải tiến này sẽ giúp hệ thống ngày càng hoàn thiện, tăng tính thực tiễn và khả năng ứng dụng trong môi trường thực tế.

PHỤ LỤC

Code Apps Script:

```
function doGet(e) {
  var uid = e.parameter.uid;
  Logger.log("Received UID: " + uid);
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("NguoiDung");
  var data = sheet.getDataRange().getValues();

  for (var i = 1; i < data.length; i++) {
    if (data[i][0] == uid) {
      var name = data[i][1];
      var balance = parseFloat(data[i][2]);
      var topup = parseFloat(data[i][3]);

      var newBalance = balance + topup;
      sheet.getRange(i + 1, 3).setValue(newBalance);
      sheet.getRange(i + 1, 4).setValue(0); // Reset topup column
after adding

      return ContentService.createTextOutput(name + "|" +
newBalance);
    }
  }

  return ContentService.createTextOutput("UNKNOWN");
}

function doPost(e) {
  var sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("NguoiDung");
  const action = e.parameter.action;
  const uid = e.parameter.uid;
  const sodu = e.parameter.sodu;

  if (action == "update") {
    const data = sheet.getDataRange().getValues();
    for (let i = 1; i < data.length; i++) {
      if (data[i][0] == uid) {
        sheet.getRange(i + 1, 3).setValue(sodu); // Cập nhật số dư
vào cột 3
        return ContentService.createTextOutput("OK");
      }
    }
    return ContentService.createTextOutput("UNKNOWN");
  }
}
```

```
    return ContentService.createTextOutput("Invalid action");  
}
```

TÀI LIỆU THAM KHẢO

- [1] R. Want, “*An introduction to RFID technology*”, IEEE, 2006.
- [2] Serge Gabet, “*Google Apps Script for Beginners*”, Packt Publishing, 2014.
- [3] Michael Maguire, “*Google Sheets Programming With Google Apps Script*”, 2016.
- [4] Nshop Linh kiện điện tử, “*Kit RF thu phát WiFi Bluetooth Esp32 Type C*”, nshop.com, 2019.
- [5] NXP Semiconductor, “*MFRC522 Standard performance MIFARE and NTAG frontend*”, NXP Semiconductor Datasheet, 2016.
- [6] Hồ Đình Khải, “*Hệ thống bán hàng tự động*”, Khoa Điện – Điện tử, trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh, 2018.