

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO ĐỒ ÁN THIẾT KẾ LUẬN LÝ

Đề tài
Ổ KHÓA ĐIỆN TỬ

GVHD: Nguyễn Thành Lộc

Họ và tên	MSSV
Dương Khôi Nguyên	2312333
Trần Hoàng Bá Huy	2311249
Nguyễn Hảo Khang	2311444



Mục lục

1	Giới thiệu	2
1.1	Tính năng của sản phẩm	2
1.2	Giới hạn của đề tài	3
2	Cấu tạo và Nguyên lý hoạt động	4
2.1	Nguyên lý hoạt động	4
2.2	Linh kiện phần cứng	4
2.3	Giải thuật Knutt-Morris-Pratt	9

1 Giới thiệu

Khóa cửa là công cụ hữu dụng nhất trong việc đảm bảo nơi cư trú của một cá nhân an toàn với thế giới bên ngoài. Từ thô sơ đến phức tạp, khóa cửa biến đổi từ những then gỗ đơn giản cho đến hiện tại, trong bối cảnh an ninh và tiện lợi được phép tồn tại song song, các giải pháp khóa cửa thông minh đang dần thay thế các loại khóa cơ truyền thống sử dụng chìa. Đây là lý do nhóm chọn đề tài "**Ổ khóa điện tử**"

Đề tài tập trung vào việc nghiên cứu và xây dựng một hệ thống khóa cửa tự động sử dụng vi điều khiển. Sản phẩm sử dụng bo mạch STM32F103C8T6, với họ vi điều khiển lõi Arm[®] Cortex[®]-M3 STM32F1 làm đơn vị xử lý trung tâm. Hệ thống được thiết kế để cung cấp một giải pháp an ninh linh hoạt, kết hợp giữa việc xác thực bằng mật khẩu số và khả năng dự phòng bằng chìa khóa cơ.

1.1 Tính năng của sản phẩm

- **Mở khóa bằng mật khẩu:** Người dùng nhập mật khẩu qua một bàn phím (keypad). Hệ thống cho phép nhập một lượng tối đa các ký tự và tìm mật khẩu dựa trên những ký tự nhập vào.
- **Mở khóa bằng chìa khóa cơ:** Hệ thống tích hợp một ổ khóa cơ học truyền thống. Khi sử dụng chìa khóa cơ, một cảm biến sẽ được kích hoạt để mở khóa và khởi động lại hệ thống nếu đang bị vô hiệu hóa tạm thời.
- **Quản lý mật khẩu:** Cho phép người dùng tạo lập và thay đổi mật khẩu bên trong ổ khóa theo các quy trình phổ biến hiện nay.
- **Cảnh báo an ninh:** Nhập sai mật khẩu nhiều lần liên tiếp, hệ thống sẽ phát cảnh báo và vô hiệu hóa bàn phím, trước khi vô hiệu hóa hoàn toàn bàn phím sau một số lần nhập cụ thể và phải sử dụng chìa khóa cơ để mở lại.
- **Giao diện người dùng:** Thông tin của ổ khóa và mật khẩu đang được nhập sẽ được hiển thị qua một màn hình LED 16×2 .
- **Các tính năng tiện ích:**

- Trang bị một nút bấm ở mặt trong ổ khóa để mở khóa cửa từ bên trong
- Do ổ khóa hoạt động trên một nguồn điện hữu hạn và cần cấp lại sau một thời gian hệ thống trang bị một mạch giám sát pin và đưa ra cảnh báo khi pin yếu.

1.2 Giới hạn của đề tài

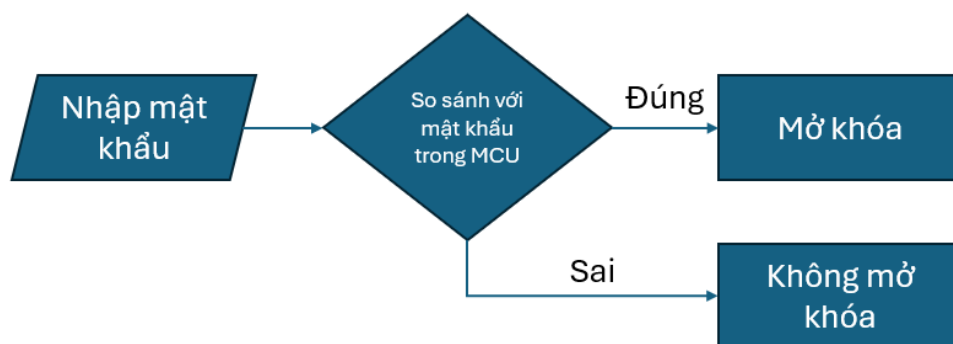
Phạm vi của đề tài được giới hạn trong các khía cạnh kỹ thuật sau:

- **Nền tảng phần cứng:** Thiết kế tập trung vào vi điều khiển STM32F103C8T6. Các linh kiện ngoại vi được lựa chọn cụ thể bao gồm LCD 16x2 (giao tiếp I²C) , khóa Solenoid 12V LY - 03 , và Keypad 4×4 .
- **Phạm vi xác thực mật khẩu:** Mật khẩu được lưu trong MCU có độ dài cố định là bốn ký tự. Mặc dù người dùng có thể nhập chuỗi dài hơn, cơ chế xác thực chỉ dựa trên việc tìm kiếm chuỗi bốn ký tự này bằng thuật toán Knuth – Morris – Pratt , không sử dụng các phương thức mã hóa phức tạp.
- **Nguồn cấp:** Hệ thống được thiết kế ưu tiên cho việc sử dụng bốn pin AA với mục tiêu hoạt động tối thiểu sáu ngày. Đề tài không tập trung vào các giải pháp sử dụng nguồn điện lưới hoặc pin sạc dung lượng cao.

2 Cấu tạo và Nguyên lý hoạt động

2.1 Nguyên lý hoạt động

Ổ khóa điện tử cho phép người dùng mở khóa cửa bằng cách sử dụng mật khẩu hoặc sử dụng chìa khóa cơ. Quy trình xử lý đơn giản của ổ khóa được mô tả trong sơ đồ sau:



Hình 1: Sơ đồ hoạt động của ổ khóa điện tử

Người dùng sẽ nhập một chuỗi hai mươi ký tự trong đó có chuỗi con bốn ký tự là mật khẩu của hệ số thập lục phân. Sau đó, vi xử lý trung tâm sẽ tìm kiếm mật khẩu trong chuỗi vừa nhập bằng thuật toán **Knutt - Morris - Pratt** và so sánh với dữ liệu được lưu trong bộ nhớ của vi. Nếu đúng, vi xử lý trung tâm sẽ gửi tín hiệu để mở chốt khóa và ngược lại. Để thao tác dễ dàng hơn, người sử dụng cũng sẽ được thấy và được cảnh báo các trạng thái của ổ khóa thông qua một màn hình LCD.

2.2 Linh kiện phần cứng

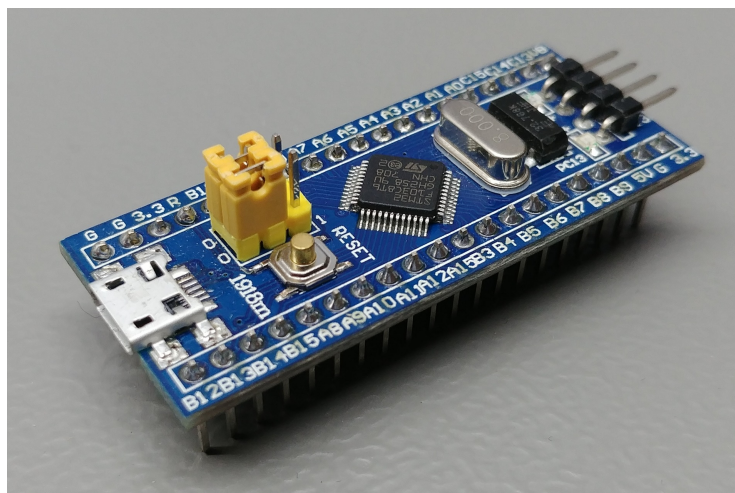
2.2.1 Vi điều khiển STM32F103C8T6

Vi điều khiển STM32F103C8T6 thuộc dòng STM32F1 của STMicroelectronics, là một lựa chọn phổ biến cho các dự án hệ thống nhúng đòi hỏi sự cân bằng giữa hiệu năng, bộ nhớ, và chi phí. Thiết bị được tích hợp lõi Arm[®] Cortex[®]-M3 32-bit RISC hiệu năng cao hoạt động ở tần số 72MHz, bộ nhớ nhúng tốc độ cao và có nhiều chân

I/O ngoại vi mở rộng, được kết nối qua hai bus APB. Vi điều khiển đóng vai trò chính trong việc nhận, xử lý và truyền dữ liệu thông qua giao thức GPIO và I²C.

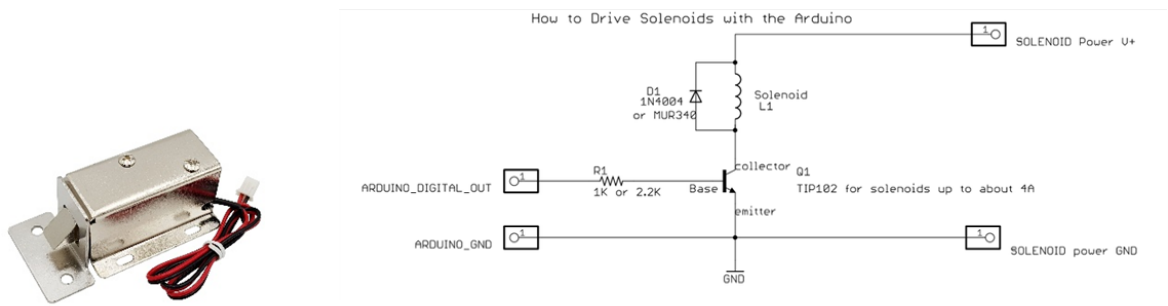
Đặc điểm kỹ thuật:

- Lõi xử lý trung tâm: Arm[®] Cortex[®]-M3
 - Tần số tối đa 72 MHz, hiệu suất truy cập bộ nhớ đạt 1.25 DMIPS/MHz (Dhrystone 2.1) khi không có chu kỳ đang chờ.
- Bộ nhớ:
 - Flash: 64KB
 - SRAM: 20KB
- Giao tiếp ngoại vi
 - Tối đa 2 giao diện I²C (hỗ trợ SMBus/PMBus).
 - Tối đa 3 bộ USART (hỗ trợ giao diện ISO 7816, LIN, hồng ngoại IrDA, điều khiển modem).
 - Tối đa 2 giao diện SPI (tốc độ lên đến 18 Mbit/s).
 - Giao diện CAN (chuẩn 2.0B Active).
 - Giao diện USB 2.0 tốc độ tối đa.
- 51 cổng giao tiếp I/O.



2.2.2 Khóa chốt điện từ LY-03

Solenoid Lock LY-03, hoạt động như một ổ khóa cửa sử dụng Solenoid để kích đóng mở bằng dòng điện. Trong đề tài này, nhóm sử dụng biến áp và relay 5V để cấp, ổn áp dòng điện để truyền tín hiệu từ vi điều khiển tới khóa chốt.



Hình 3: Hình ảnh thực tế (trái) và nguyên lý hoạt động (phải) của Solenoid

Thông số	Giá trị
Điện áp sử dụng	12V DC
Dòng điện tiêu thụ	0.8A
Công suất tiêu thụ	9.6W
Loại Solenoid	Solenoid từ
Tốc độ phản ứng	dưới 1s
Thời gian kích liên tục	dưới 10s

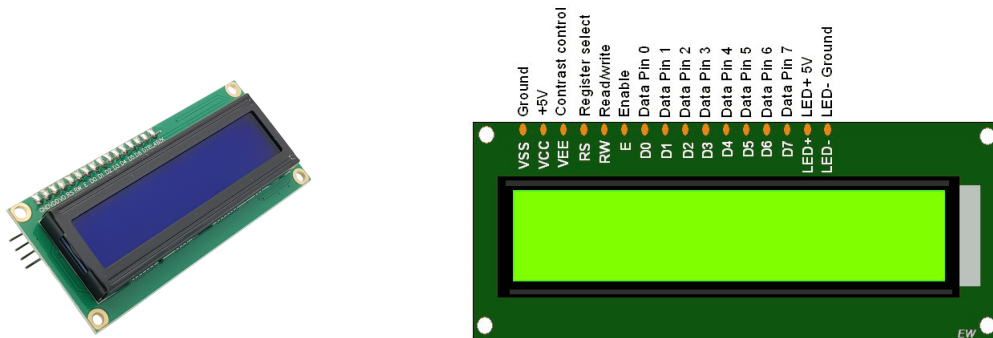
Bảng 1: Thông số khóa điện LY-03

2.2.3 Màn hình LED LCD 16 × 2

Màn hình với nền xanh dương sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự. Kết hợp với mạch chuyển giao tiếp I²C để dễ dàng truyền nhận dữ liệu với bộ xử lý trung tâm.

Thông số	Giá trị
Điện áp sử dụng	5V DC
Kích thước	$80 \times 36 \times 12.5$ mm
Hiển thị	Chữ trắng, nền xanh dương

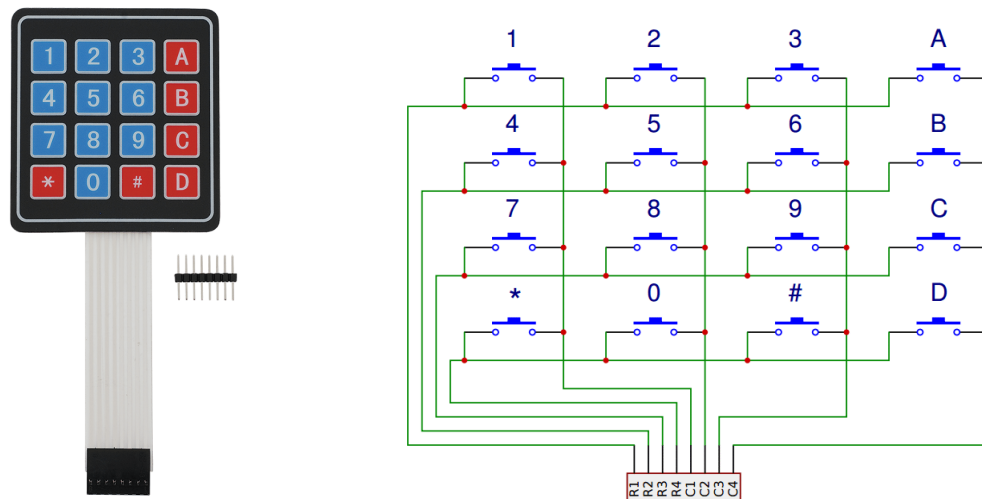
Bảng 2: Thông số của màn hình LED 16×2



Hình 4: Hình ảnh thực tế (trái) và bản đồ chân cắm (phải) của màn hình LED

2.2.4 Bàn phím ma trận 4×4

Bàn phím ma trận (Matrix Keypad) 4×4 là một giải pháp nhập liệu rời rạc phổ biến, cung cấp khả năng nhập dữ liệu số và lệnh điều khiển thông qua mười sáu phím bấm vật lý, được sắp xếp theo cấu trúc lưới không gian hai chiều. Ưu điểm cốt lõi của kiến trúc này là khả năng tối ưu hóa tài nguyên phần cứng (GPIO Optimization), cho phép quản lý mười sáu trạng thái logic độc lập với số đường tín hiệu chỉ bằng một nửa số trạng thái. Các phím bấm không có nguồn điện riêng và chia sẻ đường tín hiệu chung, vì điều khiển không thể đọc trực tiếp trạng thái của tất cả mười sáu phím cùng một lúc. Thay vào đó, hệ thống phải sử dụng kỹ thuật quét lên toàn bộ bàn phím, với mỗi lần quét được thực thi cách nhau một khoảng thời gian rất ngắn cố định để đảm bảo ghi nhận được mọi dữ liệu mà người dùng đưa vào từ bàn phím.



Hình 5: Hình ảnh thực tế và sơ đồ nút (từ trái qua phải) của bàn phím

Thông số	Giá trị
Độ dài dây cắm	88mm
Số đầu nối	8 chân
Kích thước bàn phím	77 × 69 mm
Nhiệt độ hoạt động	0 °C đến 70 °C

Bảng 3: Thông số của bàn phím ma trận

2.2.5 Các thành phần khác

Tên linh kiện	Công dụng
Jumper đực-đực, đực-cái, cái-cái	Kết nối các linh kiện
Relay 5V	Ổn định điện áp cho Solenoid, nhận tín hiệu từ vi điều khiển
Mạch giao tiếp I ² C	Hiện thực giao tiếp giữa màn hình LCD và vi điều khiển

Bảng 4: Các thành phần khác của ổ khóa

2.3 Giải thuật Knutt-Morris-Pratt

Thuật toán **Knuth-Morris-Pratt** (KMP) được phát triển bởi Donald Knuth, Vaughan Pratt, và James Morris vào năm 1977. Nó được ứng dụng rộng rãi trong các công cụ tìm kiếm, trình biên dịch, và trình soạn thảo văn bản. Đây là một thuật toán tìm kiếm chuỗi hiệu quả, được dùng để tìm một mẫu trong một đoạn văn bản. Tiền xử lý – Xây dựng mảng LPS (Longest Prefix Suffix). Thuật toán này sử dụng bước tiền xử lý để xử lý các trường hợp không khớp, nhờ đó đạt được độ phức tạp thời gian tuyến tính $O(n)$.

Thuật toán được triển khai với hai bước như sau:

Bước 1: Tiền xử lý - Xây dựng mảng LPS (Longest Prefix Suffix)

Ta xử lý chuỗi mẫu để tạo ra một mảng LPS. Mảng có ý nghĩa là: “Nếu ký tự không khớp tại vị trí này, ta có thể lùi lại bao nhiêu ký tự trong mẫu mà vẫn không bỏ lỡ khả năng khớp nào?” Nhờ đó, ta không cần phải quay lại đầu mẫu mỗi khi gặp lỗi không khớp. Bước này chỉ được thực hiện một lần duy nhất, trước khi bắt đầu tìm kiếm trong chuỗi văn bản.

Bước 2: So sánh - Tìm chuỗi con trong đoạn chuỗi

Ta sẽ bắt đầu so sánh các ký tự trong chuỗi nhập vào lần lượt một ký tự:

- Nếu ký tự khớp: di chuyển tiếp đến vị trí tiếp theo của cả chuỗi mẫu và chuỗi nhập vào.
- Nếu ký tự không khớp, sẽ có hai trường hợp:
 - Nếu không bắt đầu ở ký tự đầu tiên ở ký tự mẫu, ta dùng giá trị mảng LPS ở vị trí trước để đưa con trỏ chuỗi mẫu về vị trí đó. Điều đó có nghĩa là: nhảy về đến tiền tố dài nhất đồng thời cũng là hậu tố - không phải kiểm tra lại các ký tự đó.
 - Nếu đó là ký tự đầu, ta chỉ cần chuyển tới vị trí tiếp theo của chuỗi vào để kiểm tra.
- Nếu tới cuối chuỗi mẫu, thì ta đã tìm thấy chuỗi đó. Ta sẽ ghi nhận lại vị trí bắt đầu và tiếp tục kiểm tra.