|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  BÁO CÁO BÀI TẬP LỚN  **HỆ THỐNG NHÚNG VÀ THIẾT KẾ**  **GIAO TIẾP NHÚNG**  **Đề tài:**  **Thiết kế máy giao dịch giải quyết thủ tục**  **cho sinh viên**  Sinh viên thực hiện: Trần Khương Duy - 20172514    Giảng viên hướng dẫn: TS. Phạm Văn Tiến  Hà Nội, tháng 12 năm 2021 |

**MỤC LỤC**

[DANH MỤC HÌNH VẼ 1](#_Toc91413208)

[CHƯƠNG 1. YÊU CẦU CỦA HỆ THỐNG 3](#_Toc91413209)

[1.1 Hoạt động nghiệp vụ 3](#_Toc91413210)

[1.2 Xác lập yêu cầu kĩ thuật 3](#_Toc91413211)

[1.2.1 Yêu cầu chức năng 3](#_Toc91413212)

[1.2.2 Yêu cầu phi chức năng 4](#_Toc91413213)

[CHƯƠNG 2. MÔ HÌNH HÓA HỆ THỐNG BẰNG UML 5](#_Toc91413214)

[2.1 Biểu đồ usecase của hệ thống 5](#_Toc91413215)

[2.2 Biểu đồ hoạt động của hệ thống 6](#_Toc91413216)

[2.3 Biểu đồ tuần tự của hệ thống 7](#_Toc91413217)

[2.3.1 Biểu đồ tuần tự cho chức năng Đăng nhập 7](#_Toc91413218)

[2.3.2 Biểu đồ tuần tự cho chức năng Đăng ký mở rộng lớp 7](#_Toc91413219)

[2.3.3 Biểu đồ tuần tự cho chức năng Cấp giấy chứng nhận 8](#_Toc91413220)

[2.4 Biểu đồ trạng thái của hệ thống 8](#_Toc91413221)

[CHƯƠNG 3. THIẾT KẾ PHẦN CỨNG HỆ THỐNG 10](#_Toc91413222)

[3.1 Giới thiệu về phần mềm Proteus và STM32CubeMX 10](#_Toc91413223)

[3.1.1 Phần mềm Proteus 10](#_Toc91413224)

[3.1.2 Phần mềm STM32CubeMX 10](#_Toc91413225)

[3.2 Các phần tử trong sơ đồ mạch 10](#_Toc91413226)

[3.2.1 Màn hình LCD LM044L 10](#_Toc91413227)

[3.2.2 Nút bấm 11](#_Toc91413228)

[3.2.3 Vi điều khiển STM32F103C6 12](#_Toc91413229)

[3.2.4 Keypad tự tạo 13](#_Toc91413230)

[3.2.5 Motor và mạch điều khiển L293D 13](#_Toc91413231)

[3.3 Sơ đồ mạch hoàn chỉnh 14](#_Toc91413232)

[3.3.1 Kết nối chân giữa LCD và STM32F103C6 15](#_Toc91413233)

[3.3.2 Kết nối chân giữa Keypad và STM32F103C6 15](#_Toc91413234)

[3.3.3 Kết nối chân giữa L293D và STM32F103C6 16](#_Toc91413235)

[3.3.4 Kết nối chân giữa L293D và Motor 16](#_Toc91413236)

[CHƯƠNG 4. PHÁT TRIỂN PHẦN MỀM HỆ THỐNG 17](#_Toc91413237)

[4.1 Giới thiệu về phần mềm KeilC uVision5 17](#_Toc91413238)

[4.2 Lưu đồ thuật toán trong lập trình hệ thống 17](#_Toc91413239)

[4.3 Các hàm chính sử dụng trong lập trình hệ thống 19](#_Toc91413240)

[CHƯƠNG 5. KẾT QUẢ THỬ NGHIỆM 22](#_Toc91413241)

[5.1 Kết quả mô phỏng chức năng Đăng nhập 22](#_Toc91413242)

[5.2 Kết quả mô phỏng của chức năng Cấp giấy chứng nhận 23](#_Toc91413243)

[5.3 Kết quả mô phỏng của chức năng Đăng ký mở rộng lớp 25](#_Toc91413244)

[5.3.1 Chức năng Đăng ký lớp mới 25](#_Toc91413245)

[5.3.2 Chức năng Xem lớp đã đăng ký 27](#_Toc91413246)

[CHƯƠNG 6. ĐÁNH GIÁ HỆ THỐNG 28](#_Toc91413247)

[6.1 Đánh giá về sơ đồ mạch 28](#_Toc91413248)

[6.2 Đánh giá về quá trình mô phỏng 28](#_Toc91413249)

[PHỤ LỤC 29](#_Toc91413250)

DANH MỤC HÌNH VẼ

[Hình 2.1 Biểu đồ usecase của hệ thống 5](#_Toc91412866)

[Hình 2.2 Biểu đồ hoạt động của hệ thống 6](#_Toc91412867)

[Hình 2.3 Biểu đồ tuần tự của usecase Đăng nhập 7](#_Toc91412868)

[Hình 2.4 Biểu đồ tuần tự của usecase Đăng ký mở rộng lớp 7](#_Toc91412869)

[Hình 2.5 Biểu đồ tuần tự của usecase Cấp giấy chứng nhận 8](#_Toc91412870)

[Hình 2.6 Biểu đồ trạng thái của hệ thống 9](#_Toc91412871)

[Hình 3.1 Hình ảnh của LCD LM044L trong Proteus 11](#_Toc91412872)

[Hình 3.2 Nút bấm trong Proteus 11](#_Toc91412873)

[Hình 3.3 Hình ảnh của STM32F103C6 trong Proteus 12](#_Toc91412874)

[Hình 3.4 Cấu hình chân của STM32F103C6 trong STM32CubeMX 12](#_Toc91412875)

[Hình 3.5 Hình ảnh Keypad tự tạo trong Proteus 13](#_Toc91412876)

[Hình 3.6 Hình ảnh Motor trong Proteus 13](#_Toc91412877)

[Hình 3.7 Hình ảnh của L293D trong Proteus 14](#_Toc91412878)

[Hình 3.8 Sơ đồ mạch hoàn chỉnh trong Proteus 14](#_Toc91412879)

[Hình 4.1 Lưu đồ thuật toán phần 1 17](#_Toc91412880)

[Hình 4.2 Lưu đồ thuật toán phần 2 18](#_Toc91412881)

[Hình 4.3 Lưu đồ thuật toán phần 3 18](#_Toc91412882)

[Hình 4.4 Lưu đồ thuật toán phần cuối 19](#_Toc91412883)

[Hình 5.1 Giao diện bắt đầu 22](#_Toc91412884)

[Hình 5.2 Giao diện đăng nhập 22](#_Toc91412885)

[Hình 5.3 Giao diện lỗi đăng nhập 22](#_Toc91412886)

[Hình 5.4 Giao diện chính của máy giao dịch 23](#_Toc91412887)

[Hình 5.5 Giao diện chọn mục đích xin giấy chứng nhận 23](#_Toc91412888)

[Hình 5.8 Giao diện thông tin giấy chứng nhận 23](#_Toc91412889)

[Hình 5.9 Giao diện xác nhận in giấy chứng nhận 24](#_Toc91412890)

[Hình 5.10 Giao diện đang in giấy 24](#_Toc91412891)

[Hình 5.11 Giao diện thông báo in thành công 24](#_Toc91412892)

[Hình 5.12 Giao diện tiếp tục giao dịch hoặc thoát 24](#_Toc91412893)

[Hình 5.13 Giao diện chọn chức năng con 25](#_Toc91412894)

[Hình 5.14 Giao diện nhập mã lớp 25](#_Toc91412895)

[Hình 5.15 Giao diện lỗi mã lớp 1 25](#_Toc91412896)

[Hình 5.16 Giao diện lỗi mã lớp 2 25](#_Toc91412897)

[Hình 5.17 Giao diện tên lớp đăng ký 26](#_Toc91412898)

[Hình 5.18 Giao diện chọn lý do đăng ký lớp 26](#_Toc91412899)

[Hình 5.19 Giao diện đăng ký lớp thành công 26](#_Toc91412900)

[Hình 5.20 Giao diện tiếp tục giao dịch hoặc thoát 26](#_Toc91412901)

[Hình 5.21 Giao diện chưa đăng ký lớp 27](#_Toc91412902)

[Hình 5.22 Giao diện các lớp đã đăng ký 27](#_Toc91412903)

[Hình 5.23 Giao diện thông tin lớp đăng ký 27](#_Toc91412904)

# YÊU CẦU CỦA HỆ THỐNG

Chương 1 mô tả quy trình hoạt động của một máy giao dịch giải quyết thủ tục sinh viên. Chương này còn đưa ra các yêu cầu về chức năng và phi chức năng của hệ thống. Các yêu cầu này là cơ sở cần thiết để xây dựng hệ thống máy giao dịch giải quyết thủ tục sinh viên.

## Hoạt động nghiệp vụ

* Sinh viên sử dụng tài khoản và mật khẩu đã được cấp để đăng nhập vào hệ thống.
* Máy định danh sinh viên, hiển thị thông tin sinh viên và các chức năng lên màn hình.
* Tiếp theo, sinh viên có thể in ra giấy chứng nhận sinh viên hoặc đăng ký mở rộng lớp cho các môn của học kì hiện tại.
* Ở bất kỳ thời gian nào, hệ thống chỉ cho phép một giao dịch.
* Máy thông báo thành công hoặc thất bại khi kết thúc giao dịch.
* Sau khi hoàn thành giao dịch, sinh viên có thể thực hiện giao dịch khác.
* Sinh viên chọn thoát để thoát khỏi hệ thống nếu không thực hiện giao dịch nào nữa.

## Xác lập yêu cầu kĩ thuật

### Yêu cầu chức năng

* Giúp định danh sinh viên bằng cách đăng nhập, hiển thị thông tin sinh viên lên màn hình.
* Giúp in ra giấy chứng nhận sinh viên.
* Hỗ trợ sinh viên đăng ký mở rộng lớp cho học kì hiện tại, giúp sinh viên xem thông tin của các lớp đã đăng ký.

### Yêu cầu phi chức năng

* Sử dụng một màn hình để hiển thị thông tin và các chức năng của máy.
* Sử dụng một keypad để nhập dữ liệu và chọn các chức năng.
* Sử dụng nút bấm để xác nhận giao dịch.
* Sử dụng vi điều khiển STM32 để điều khiển các chức năng của hệ thống.
* Sử dụng một máy in để in giấy tờ.
* Tốc độ xử lý các thao tác chức năng phải nhanh, không quá 5s.
* Sử dụng phần mềm Rational Rose để vẽ mô hình UML cho hệ thống.
* Sử dụng phần mềm Proteus để thiết kế kiến trúc phần cứng và thực hiện mô phỏng hệ thống.
* Sử dụng phần mềm STM32CubeMX để cấu hình các chân GPIO cho vi điểu khiển STM32.
* Sử dụng phần mềm KeilC uVision5 để lập trình firmware cho phần cứng của hệ thống.

# MÔ HÌNH HÓA HỆ THỐNG BẰNG UML

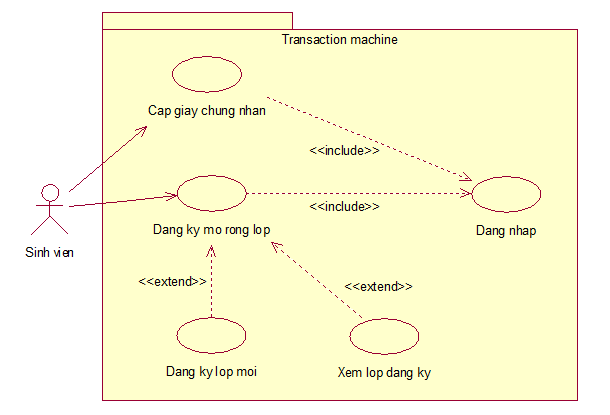
Chương 2 mô tả hệ thống bằng bốn biểu đồ UML là: Biểu đồ usecase, biểu đồ hoạt động, biểu đồ tuần tự, biểu đồ trạng thái. Biểu đồ usecase biểu diễn các chức năng của hệ thống. Biểu đồ hoạt động biểu diễn quy trình hoạt động của hệ thống. Biểu đồ tuần tự biểu diễn tương tác giữa các đối tượng trong hệ thống. Biểu đồ trạng thái biểu diễn trạng thái tại một thời điểm của hệ thống.

## Biểu đồ usecase của hệ thống

Hệ thống máy giao dịch giải quyết thủ tục sinh viên bao gồm năm usercase đại diện cho ba chức năng chính là:

* Đăng nhập: chức năng này kiểm tra xem tài khoản sinh viên có hợp lệ không và nếu hợp lệ thì cho phép sinh viên giao dịch với hệ thống.
* Cấp giấy chứng nhận: chức năng này giúp in ra giấy chứng nhận sinh viên dựa trên thông tin của sinh viên đã đăng nhập.
* Đăng ký mở rộng lớp: chức năng này giúp sinh viên đăng ký vào một lớp đã đầy của một học phần và xem thông tin các lớp đã đăng ký.

Dưới đây là biểu đồ usecase của máy giao dịch giải quyết thủ tục sinh viên:



Hình . Biểu đồ usecase của hệ thống

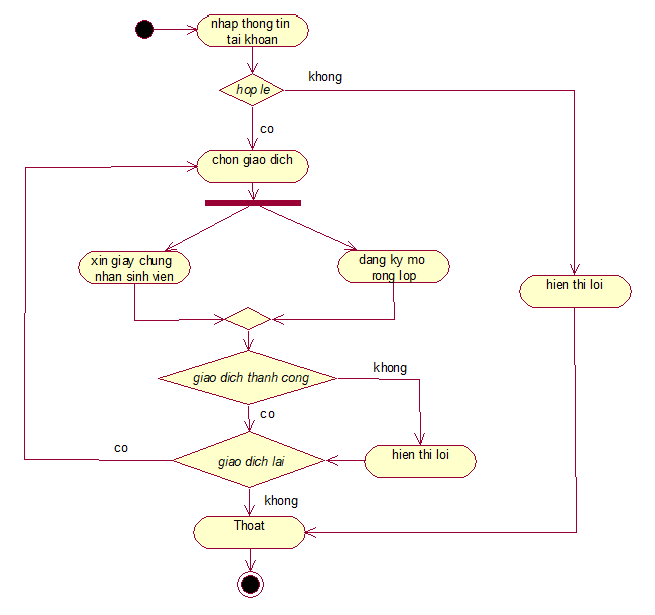
* **Giải thích về <<include>> và <<extend>>**
* <<include>> : chỉ ra rằng muốn sử dụng được chức năng này thì phải sử dụng chức năng kia trước.
* <<extend>> : chỉ ra rằng chức năng này là một chức năng con cụ thể của chức năng lớn kia.

## Biểu đồ hoạt động của hệ thống

Biểu đồ hoạt động biểu diễn các hoạt động tương tác giữa sinh viên và máy giao dịch theo một quá trình cụ thể từ khi sinh viên bắt đầu mở máy đến khi sinh viên thoát khỏi hệ thống.

Các hoạt động chính bao gồm: nhập thông tin tài khoản, chọn giao dịch, hiển thị lỗi, xin giấy chứng nhận sinh viên, đăng ký mở rộng lớp, thoát.

Dưới đây là biểu đồ hoạt động của máy giao dịch giải quyết thủ tục sinh viên:

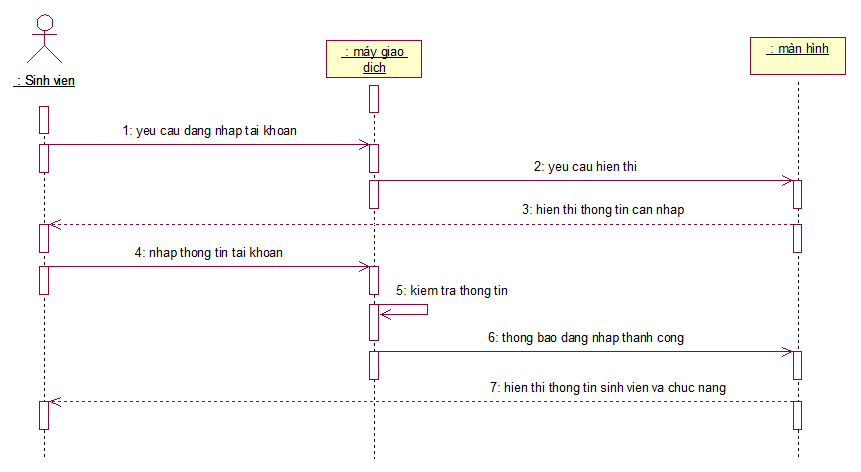


Hình . Biểu đồ hoạt động của hệ thống

## Biểu đồ tuần tự của hệ thống

### Biểu đồ tuần tự cho chức năng Đăng nhập

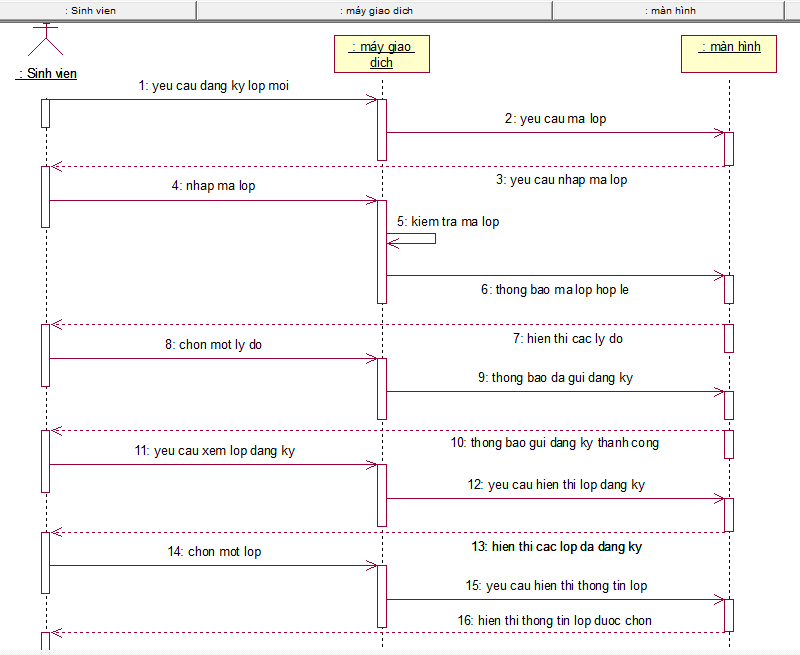
Dưới đây là biểu đồ tuần tự của chức năng Đăng nhập:



Hình . Biểu đồ tuần tự của usecase Đăng nhập

### Biểu đồ tuần tự cho chức năng Đăng ký mở rộng lớp

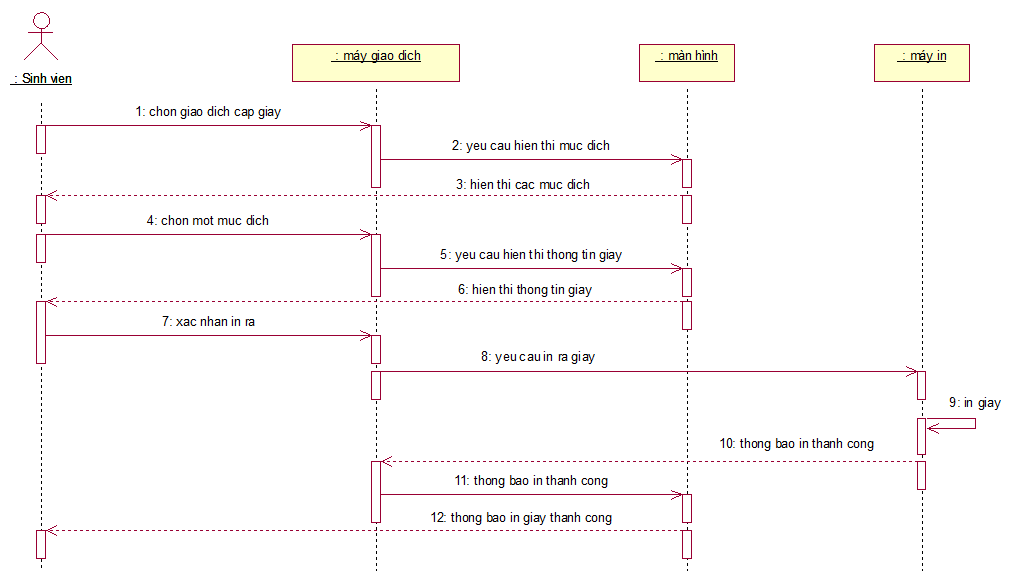
Dưới đây là biểu đồ tuần tự của chức năng Đăng ký mở rộng lớp:



Hình . Biểu đồ tuần tự của usecase Đăng ký mở rộng lớp

### Biểu đồ tuần tự cho chức năng Cấp giấy chứng nhận

Dưới đây là biểu đồ tuần tự của chức năng Cấp giấy chứng nhận:

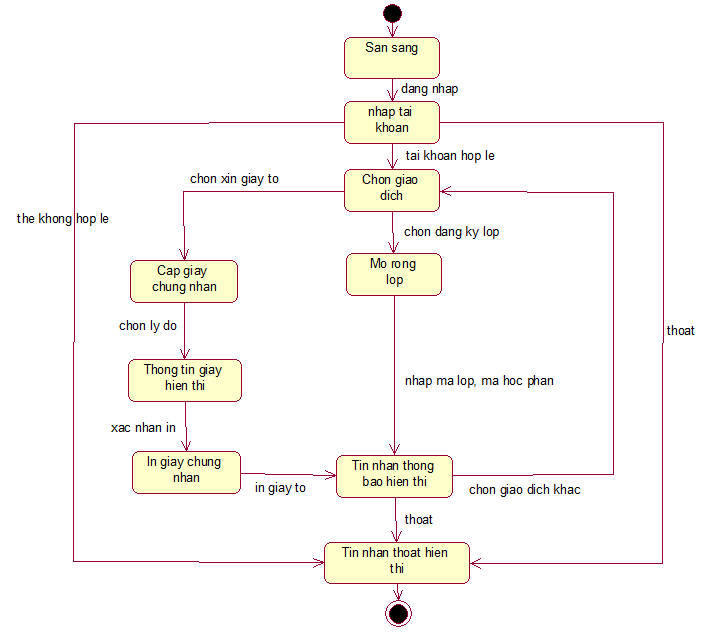


Hình . Biểu đồ tuần tự của usecase Cấp giấy chứng nhận

## Biểu đồ trạng thái của hệ thống

* **Các trạng thái của hệ thống gồm:**
* Sẵn sàng: sẵn sàng hoạt động.
* Nhập tài khoản: đợi cho sinh viên nhập thông tin tài khoản.
* Chọn giao dịch: đợi cho sinh viên chọn một giao dịch.
* Mở rộng lớp: đợi cho sinh viên chọn lý do và nhập mã lớp.
* Cấp giấy chứng nhận: đợi cho sinh viên chọn mục đích.
* Thông tin giấy hiển thị: hiển thị thông tin giấy chứng nhận sau khi sinh viên chọn một mục đích.
* In giấy chứng nhận: đợi cho máy in in ra giấy chứng nhận.
* Tin nhắn thông báo hiển thị: hiển thị thông báo thành công hoặc thất bại sau khi giao dịch.
* Tin nhắn thoát hiển thị: hiển thị thông báo thoát khi sinh viên thoát hệ thống.

Dưới đây là biểu đồ trạng thái của máy giao dịch giải quyết thủ tục sinh viên:



Hình . Biểu đồ trạng thái của hệ thống

# THIẾT KẾ PHẦN CỨNG HỆ THỐNG

Chương 3 trình bày về các linh kiện thành phần được sử dụng trong việc thiết kế máy giao dịch giải quyết thủ tục sinh viên. Các linh kiện này được mô tả dựa trên các linh kiện mô phỏng của phần mềm Proteus. Chương này cũng đưa ra sơ đồ mạch thiết kế của máy giao dịch giải quyết thủ tục sinh viên và mô tả cách kết nối giữa các phần cứng của hệ thống.

## Giới thiệu về phần mềm Proteus và STM32CubeMX

### Phần mềm Proteus

Proteus là phần mềm mô phỏng mạch điện tử của Labcenter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho cả các MCU (Micro Controller Unit) như PIC, 8051, AVR, Motorola. Phần mềm Proteus cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển như MCS-51, PIC, AVR, STM32, …

### Phần mềm STM32CubeMX

STM32CubeMX là một công cụ hỗ trợ cấu hình và sinh code cho vi điều khiển STM32. Tất cả các công việc cấu hình, nâng cấp đều được thực hiện qua giao diện đồ họa. Việc này giúp cho việc lập trình trên STM32 dễ dàng hơn, rút ngắn được thời gian nghiên cứu cũng như thời gian phát triển.

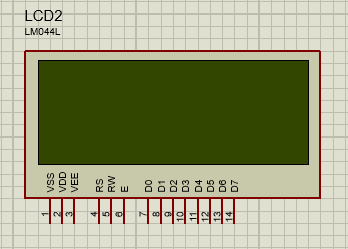
STM32CubeMX tạo code và copy các driver cần thiết vào project. Sau đó, nó tạo project trên các công cụ lập trình như Keil C.

## Các phần tử trong sơ đồ mạch

### Màn hình LCD LM044L

Màn hình LCD đại diện cho màn hình của máy giao dịch giải quyết thủ tục sinh viên. Màn hình này chỉ dùng để hiển thị thông tin và chức năng hệ thống. Màn hình LCD LM044L có thể hiển thị được tối đa 4 dòng chữ và mỗi dòng hiển thị được tối đa là 20 kí tự.

Dưới đây là hình ảnh của LCD LM044L trong phần mềm Proteus:



Hình . Hình ảnh của LCD LM044L trong Proteus

* **Sơ đồ chân của LCD**

1. VSS: tương đương với GND – cực âm.
2. VDD: tương đương với VCC – cực dương.
3. VEE (Constrast Voltage): điều khiển độ sáng màn hình.
4. Register Select (RS): điều khiển địa chỉ nào sẽ được ghi dữ liệu.
5. Read/Write (RW): chế độ đọc (read mode) hoặc ghi (write mode) dữ liệu, phụ thuộc vào việc gửi giá trị gì vào.
6. Enable pin (E): cho phép ghi dữ liệu vào LCD.
7. D0 – D7: 8 chân dữ liệu, mỗi chân sẽ có giá trị 1 hoặc 0 nếu đang ở chế độ đọc (read mode) và sẽ nhận giá trị 1 hoặc 0 nếu đang ở chế độ ghi (write mode).

### Nút bấm

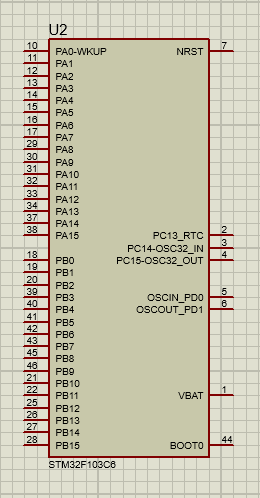
Nút bấm này hoạt động giống như một nút bấm thông thường trong thực tế. Khi người dùng nhấn vào nút thì hai đầu dây ở bên nút được nối liền với nhau và cho phép tín hiệu điện đi qua. Dưới đây là hình ảnh nút bấm trong phần mềm Proteus:



Hình . Nút bấm trong Proteus

### Vi điều khiển STM32F103C6

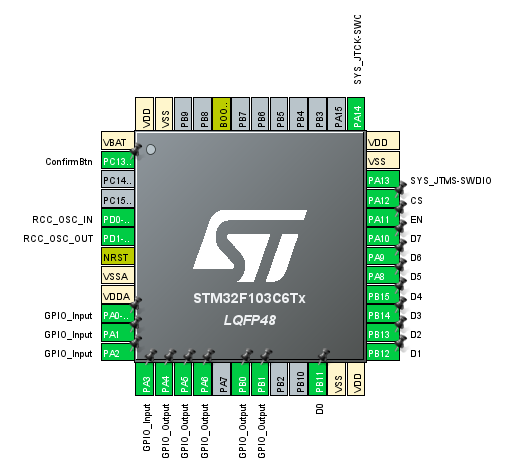
Vi điều khiển STM32F103C6 đóng vai trò như một CPU điều khiển toàn bộ các chức năng của máy giao dịch giải quyết thủ tục sinh viên. Các chân của nó sẽ được ghép nối với các chân của những phần tử khác. Dưới đây là hình ảnh của vi điều khiển STM32F103C6 trong Proteus:



Hình . Hình ảnh của STM32F103C6 trong Proteus

* **Cấu hình chân cho vi điều khiển STM32F103C6**

Dưới đây là sơ đồ chân của STM32F103C6:

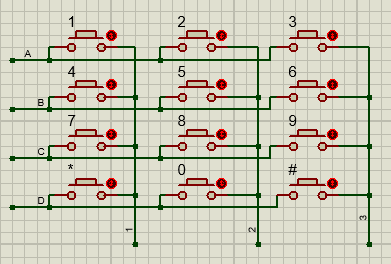


Hình . Cấu hình chân của STM32F103C6 trong STM32CubeMX

* ­­­Các chân PA4, PA5, PA6, PA8, PA9, PA10, PA11, PA12, PB0, PB1, PB11, PB12, PB13, PB14, PB15 là các chân GPIO output có giá trị ban đầu bằng 0. Vi điều khiển STM32F103C6 sẽ xuất tín hiệu điều khiển ra các chân này.
* Các chân PC13, PA0, PA1, PA2, PA3 là các chân GPIO input. Vi điều khiển STM32F103C6 sẽ nhận tín hiệu điện từ các chân này.

### Keypad tự tạo

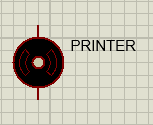
Ghép nối 12 nút bấm để tạo thành một Keypad 4x3. Các nút bấm trong Keypad này tương ứng với các nút của một Keypad 4x3 trong thực tế. Kí tự đại diện cho 12 nút gồm 10 kí tự từ 0 đến 9 và hai kí tự ‘\*’, ‘#’. Sinh viên sẽ sử dụng Keypad này để nhập các thông tin cần thiết và lựa chọn các chức năng của máy giao dịch giải quyết thủ tục sinh viên. Dưới đây là hình ảnh của Keypad tự tạo trong Proteus:



Hình . Hình ảnh Keypad tự tạo trong Proteus

### Motor và mạch điều khiển L293D

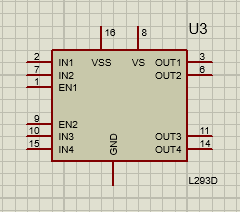
Hai phần tử này được dùng để đại diện cho một máy in do không có linh kiện máy in trong Proteus. Motor quay đại diện cho việc máy in đang in ra giấy. Dưới đây là hình ảnh Motor trong phần mềm Proteus:



Hình . Hình ảnh Motor trong Proteus

Linh kiện L293D là một mạch điều khiển động cơ dùng cho các Motor theo kiểu có thể quay tròn.

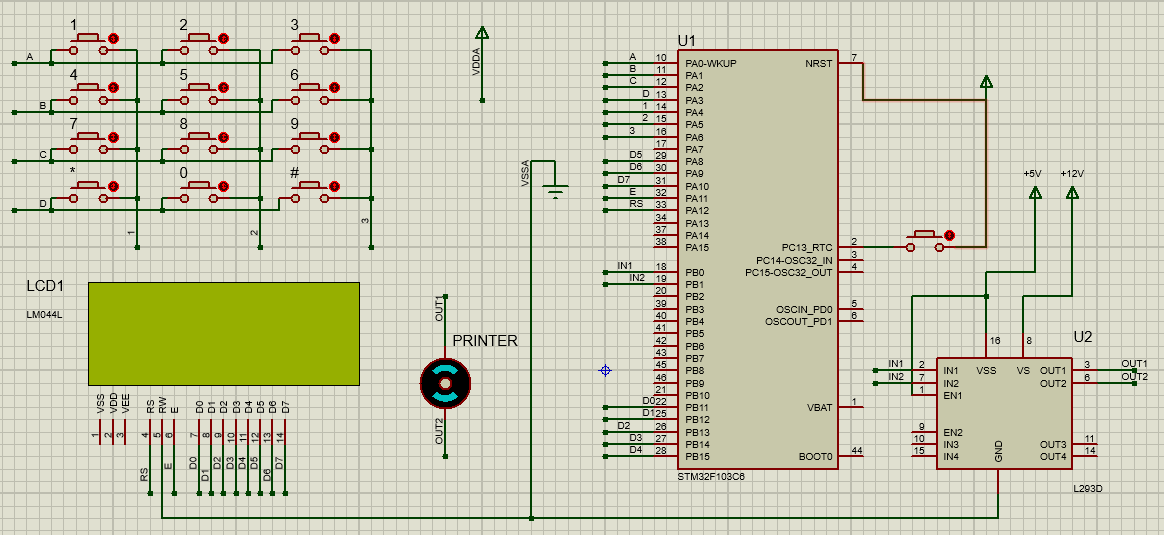
Dưới đây là hình ảnh L293D trong phần mềm Proteus:



Hình . Hình ảnh của L293D trong Proteus

## Sơ đồ mạch hoàn chỉnh

Dưới đây là sơ đồ mạch hoàn chỉnh của máy giao dịch giải quyết thủ tục sinh viên:



Hình . Sơ đồ mạch hoàn chỉnh trong Proteus

### Kết nối chân giữa LCD và STM32F103C6

* Các chân từ D0 đến D7 của LCD ở chế độ input được ghép nối với các chân PB11 đến PB15, PA8 đến PA10 của STM32F103C6 tương ứng ở chế độ output. Khi vi điều khiển muốn LCD xuất kí tự lên màn hình, nó sẽ gửi tín hiệu điện (0 hoặc 1) ra các chân PB11 đến PB15 và PA8 đến PA10. Các chân từ D0 đến D7 của LCD sẽ nhận tín hiệu và được đọc dưới dạng một số nhị phân 8 bit có dạng D7D6D5D4D3D2D1D0. Giá trị của số nhị phân này tương ứng với một kí tự trong bảng mã ASCII. LCD sẽ ghi giá trị này vào bộ nhớ và xuất kí tự lên màn hình.
* Chân RS của LCD ở chế độ input được ghép nối với chân PA12 của STM32F103C6 ở chế độ output. Khi vi điều khiển muốn gửi Lệnh điều khiển LCD thì xuất ra chân PA12 mức 0 (RS = 0), nếu muốn ghi dữ liệu vào bộ nhớ LCD thì xuất ra chân PA12 mức 1 (RS = 1).
* Chân E của LCD ở chế độ input được ghép nối với chân PA11 của STM32F103C6 ở chế độ output. Vi điều khiển chốt dữ liệu vào LCD bằng cách xuất ra chân PA11 mức 1 (E = 1), sau đó lại nhả về 0 (E = 0).
* Do khi lập trình hệ thống chỉ ghi dữ liệu lên LCD nên chân RW của LCD luôn được nối với đất hay bằng 0. Chân VSS nối đất và chân VDD nối nguồn.

### Kết nối chân giữa Keypad và STM32F103C6

* Các chân 1, 2, 3 của Keypad ở chế độ input được ghép nối với các chân PA4, PB5, PA6 của STM32F103C6 tương ứng ở chế độ output. Các chân A, B, C, D của LCD ở chế độ output được ghép nối với các chân từ PA0 đến PA3 của STM32F103C6 tương ứng ở chế độ input.
* Vi điều khiển kiểm tra xem nút nào trên Keypad được nhấn như sau:

+ Ban đầu, vi điều khiển xuất ra các chân PA4, PA5, PA6 mức 1.

+ Tiếp theo, vi điều khiển lần lượt xuất ra các chân trên mức 0.

+ Nếu có nút được nhấn thì một trong các chân từ PA0 đến PA3 sẽ nhận được tín hiệu mức 0 vì khi nhấn nút, hai chân của vi điều khiển sẽ nối liền với nhau.

+ Kiểm tra xem cặp chân nào cùng có mức 0, vi điều khiển sẽ xác định nút được nhấn tương ứng. Quá trình kiểm tra phải xảy ra rất nhanh, để có thể kiểm tra được tất cả các nút trong thời gian nhấn và thả ra.

### Kết nối chân giữa L293D và STM32F103C6

* Hai chân IN1, IN2 của L293D ở chế độ input được ghép nối với hai chân PB0, PB1 của STM32F103C6 tương ứng ở chế độ output. Tín hiệu từ hai chân input IN1, IN2 sẽ được truyền đến hai chân output OUT1, OUT2 của L293D tương ứng. Khi vi điều khiển muốn làm Motor quay, nó sẽ xuất ra chân PB0 mức 1, PB1 mức 0 hoặc ngược lại.
* Chân EN1 của L293D nối với nguồn để cho phép hai chân IN1, IN2 nhận tín hiệu vào. Chân GND nối đất, chân VSS và VS nối nguồn.

### Kết nối chân giữa L293D và Motor

* Hai chân OUT1, OUT2 của L293D ở chế độ output được ghép nối với hai chân input của Motor. Motor sẽ quay khi hai chân đầu vào của Motor khác mức điện áp. Lúc đó, chân OUT1 ở mức 1, chân OUT2 ở mức 0 hoặc ngược lại.

# PHÁT TRIỂN PHẦN MỀM HỆ THỐNG

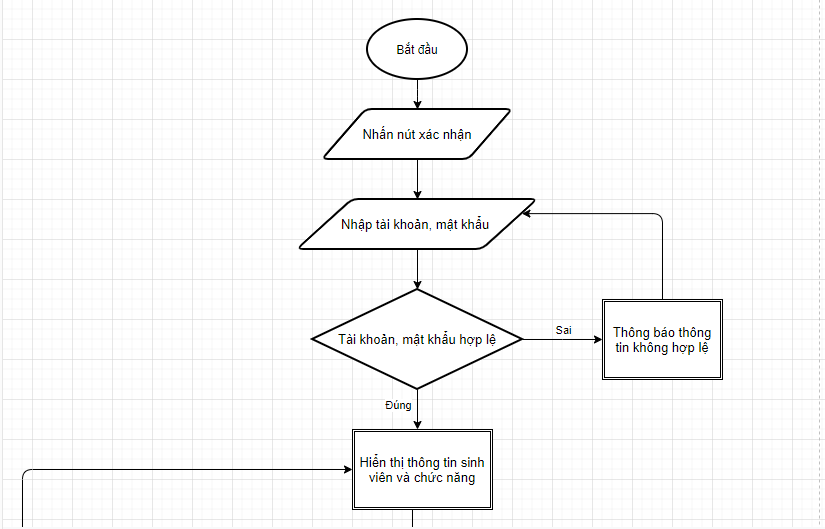
Để hệ thống hoạt động được thì hệ thống cần có một chương trình điều khiển và chương trình đó phải được nạp vào vi điều khiển. Chương trình điều khiển máy giao dịch giải quyết thủ tục sinh viên được viết trên phần mềm KeilC uVision5 và được biên dịch ra file có đuôi .hex. File này được dùng để nạp cho vi điều khiển STM32. Chương 4 trình bày về thuật toán sử dụng trong lập trình chương trình điều khiển hệ thống và giải thích chức năng của các hàm chính được sử dụng trong chương trình. Mã nguồn của hệ thống được lưu trữ trên Github, đường dẫn đến mã nguồn được trích dẫn trong phần Phụ Lục.

## Giới thiệu về phần mềm KeilC uVision5

KeilC là một phần mềm hỗ trợ cho người dùng trong việc lập trình cho vi điều khiển các dòng khác nhau (Atmel, AVR, ST, ...). KeilC giúp người dùng soạn thảo và biên dịch chương trình C hay cả assembly thành ngôn ngữ máy để nạp vào vi điều khiển giúp tương tác giữa vi điều khiển và người lập trình.

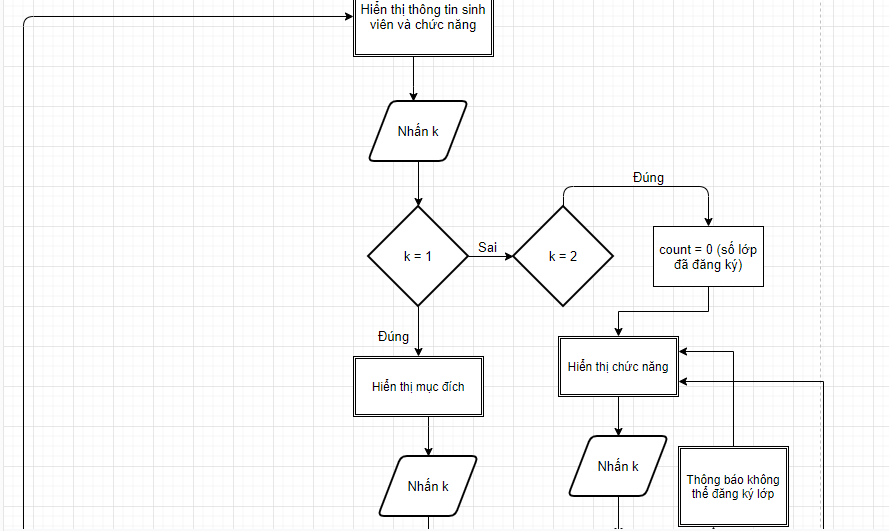
## Lưu đồ thuật toán trong lập trình hệ thống

Dưới đây là phần đầu tiên của lưu đồ thuật toán:



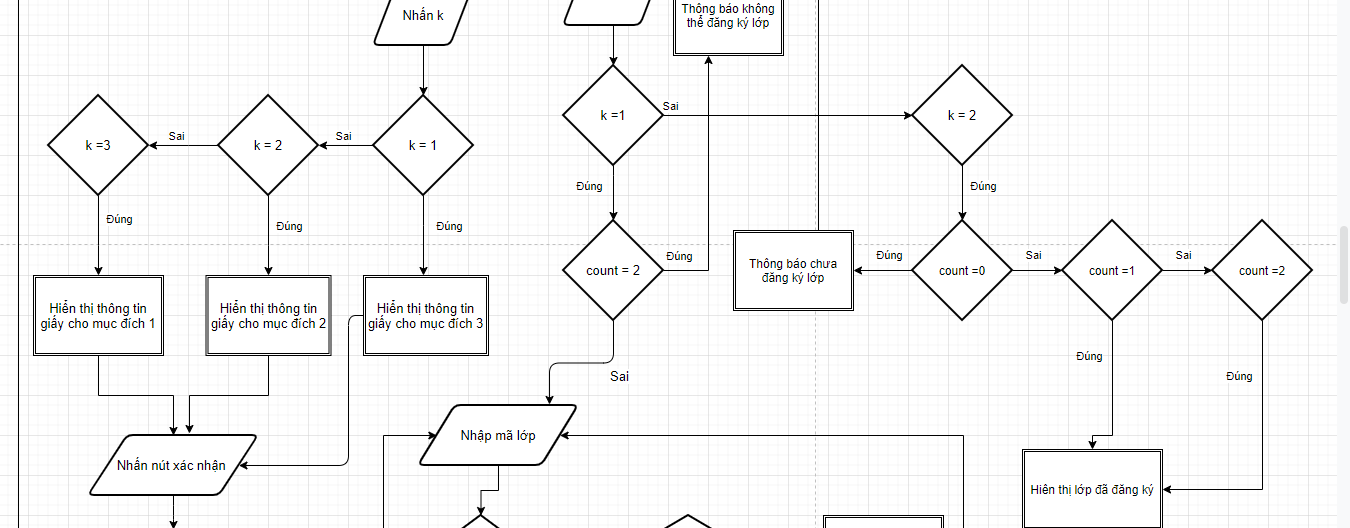
Hình . Lưu đồ thuật toán phần 1

Dưới đây là phần thứ hai của lưu đồ thuật toán:



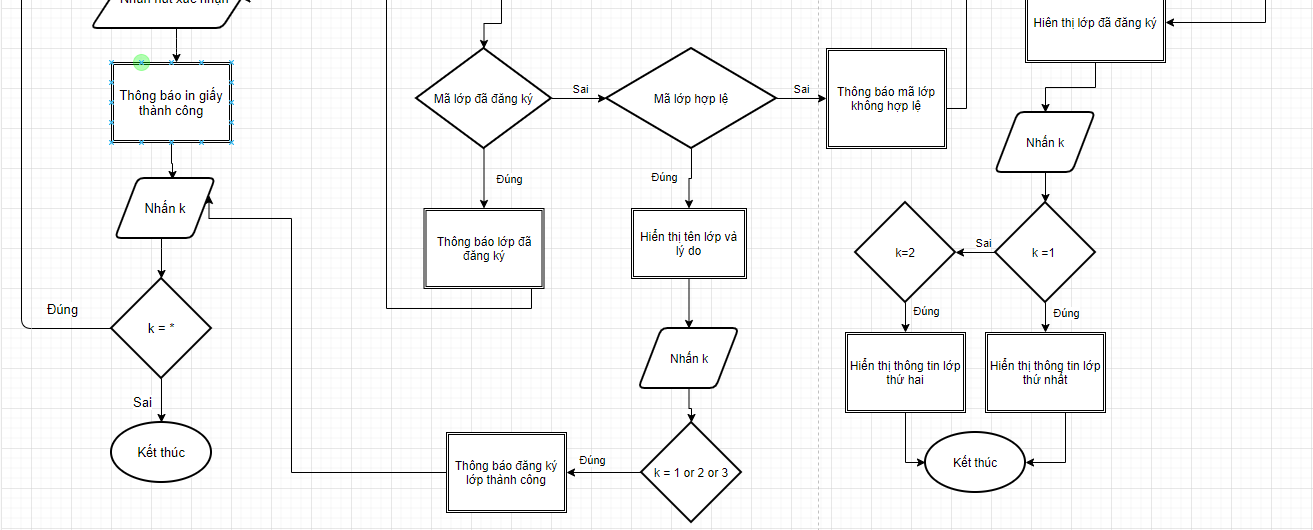
Hình . Lưu đồ thuật toán phần 2

Dưới đây là phần thứ ba của lưu đồ thuật toán:



Hình . Lưu đồ thuật toán phần 3

Dưới đây là phần cuối của lưu đồ thuật toán:



Hình . Lưu đồ thuật toán phần cuối

* **Giải thích về hai biến k và count**
* k là một giá trị trong bàn phím Keypad. Mỗi giá trị k tương ứng với một chức năng, một lý do, một mục đích, … mà sinh viên được phép lựa chọn trong hệ thống bằng cách nhấn phím của Keypad.
* count là số lớp mà sinh viên đã đăng ký trên hệ thống. Mỗi khi sinh viên đăng ký một lớp thành công thì count sẽ tăng lên 1.

## Các hàm chính sử dụng trong lập trình hệ thống

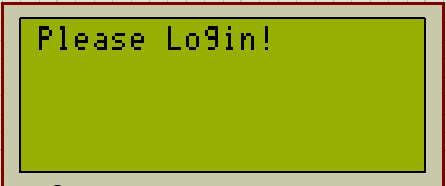
* **Các hàm được xây dựng sẵn**
* void HAL\_Delay(uint32\_t Delay): Hàm này làm trễ việc thực thi chương trình trong một khoảng thời gian nhất định. Nó chứa tham số Delay chỉ thời gian cần làm trễ, thời gian được tính theo đơn vị ms.
* GPIO\_PinState HAL\_GPIO\_ReadPin(GPIO\_TypeDef \*GPIOx, uint16\_t GPIO\_Pin): Hàm này trả về trạng thái (mức cao hoặc mức thấp) tại một chân của vi điều khiển. Tham số GPIOx chỉ ra chân được kiểm tra thuộc nhóm chân vào ra nào của vi điều khiển (x có thể là A, B hoặc C). Tham số GPIO\_Pin chỉ một chân vào ra cụ thể của vi điều khiển phải thuộc nhóm chân ở trên.
* void HAL\_GPIO\_WritePin(GPIO\_TypeDef \*GPIOx, uint16\_t GPIO\_Pin, GPIO\_PinState PinState): Hàm này thiết lập trạng thái (cao hoặc thấp) cho một chân của vi điều khiển. Tham số GPIOx chỉ ra chân được kiểm tra thuộc nhóm chân vào ra nào của vi điều khiển (x có thể là A, B hoặc C). Tham số GPIO\_Pin chỉ một chân vào ra cụ thể của vi điều khiển phải thuộc nhóm chân ở trên. Tham số PinState chỉ trạng thái cần thiết lập cho chân vi điều khiển.
* int strcmp(char \* str1, char \* str2): Hàm này được sử dụng để so sánh 2 chuỗi là str1 và str2. Nếu hàm strcmp(str1, str2) trả về 0 thì 2 chuỗi bằng nhau, trả về lớn hơn 0 thì chuỗi str1 lớn hơn chuỗi str2 và trả về nhỏ hơn 0 thì chuỗi str1 nhỏ hơn chuỗi str2.
* **Các hàm tự tạo**
* void CLCD\_SetCursor(CLCD\_Name\* LCD, uint8\_t Xpos, uint8\_t YPos): Hàm này dùng để đặt vị trí của một kí tự hoặc vị trí bắt đầu của một chuỗi trên màn hình LCD. Tham số Xpos là vị trí theo cột có giá trị từ 0 đến 19, tham số Ypos là vị trí theo hàng có giá trị từ 0 đến 3. Tham số LCD đại diện cho một màn hình LCD, nó chỉ ra màn hình LCD nào đang được sử dụng và chứa các thuộc tính của màn hình LCD đó.
* void CLCD\_WriteChar(CLCD\_Name\* LCD, char character): Hàm này dùng để hiển thị một kí tự lên màn hình LCD. Tham số character đại diện cho một kí tự bất kì. Tham số LCD tương tự như trong hàm CLCD\_SetCursor().
* void CLCD\_WriteString(CLCD\_Name\* LCD, char \*String): Hàm này dùng để hiển thị một chuỗi kí tự lên màn hình LCD. Tham số String đại diện cho một chuỗi kí tự. Tham số LCD tương tự như trong hàm CLCD\_SetCursor().
* void CLCD\_Clear(CLCD\_Name\* LCD): Hàm này được dùng để xóa toàn bộ kí tự trên màn hình LCD. Tham số LCD tương tự như trong hàm CLCD\_SetCursor().
* void CLCD\_CursorOn(CLCD\_Name\* LCD): Hàm này được dùng để hiển thị dấu gạch dưới trên màn hình LCD, nó đại diện cho vị trí của kí tự tiếp theo sẽ được hiển thị. Tham số LCD tương tự như trong hàm CLCD\_SetCursor().
* void CLCD\_CursorOff(CLCD\_Name\* LCD): Hàm này được dùng để ẩn dấu gạch dưới trên màn hình LCD. Tham số LCD tương tự như trong hàm CLCD\_SetCursor().
* char KEYPAD3X4\_Readkey(KEYPAD\_Name\* KEYPAD): Hàm này được dùng để đọc một giá trị tương ứng với nút mà người dùng nhấn trên bàn phím Keypad. Tham số KEYPAD đại diện cho một bàn phím Keypad, nó chỉ ra bàn phím Keypad nào đang được sử dụng và chứa các thuộc tính của bàn phím Keypad đó.

# KẾT QUẢ THỬ NGHIỆM

Chương 5 trình bày kết quả mô phỏng máy giao dịch giải quyết thủ tục sinh viên dưới dạng hình ảnh. Các hình ảnh mô phỏng được chia ra theo từng chức năng trong hệ thống. Kết quả mô phỏng này được lấy từ việc mô phỏng máy giao dịch giải quyết thủ tục sinh viên trên phần mềm Proteus.

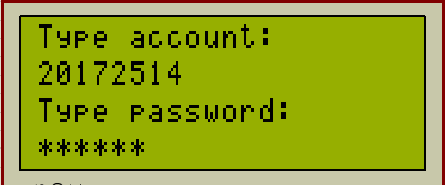
## Kết quả mô phỏng chức năng Đăng nhập

* Giao diện bắt đầu giữ nguyên cho đến khi nút xác nhận được nhấn.



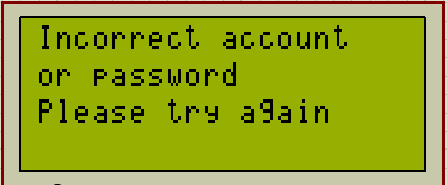
Hình . Giao diện bắt đầu

* Giao diện đăng nhập cho phép nhập tài khoản và mật khẩu.



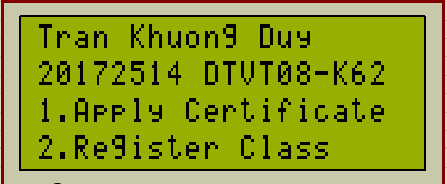
Hình . Giao diện đăng nhập

* Giao diện thông báo đăng nhập không thành công.



Hình . Giao diện lỗi đăng nhập

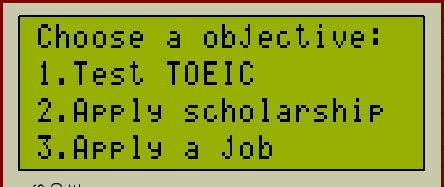
* Giao diện chính của hệ thống sau khi đăng nhập hiển thị thông tin sinh viên và các chức năng (giao dịch) chính.



Hình . Giao diện chính của máy giao dịch

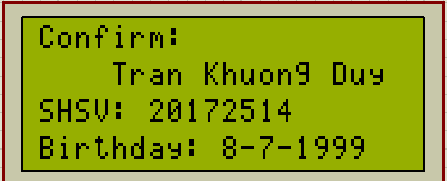
## Kết quả mô phỏng của chức năng Cấp giấy chứng nhận

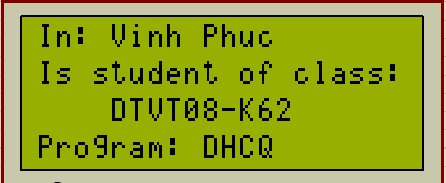
* Giao diện hiển thị các mục đích xin giấy chứng nhận sinh viên.

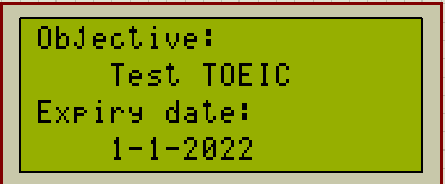


Hình . Giao diện chọn mục đích xin giấy chứng nhận

* Giao diện hiển thị thông tin giấy chứng nhận.

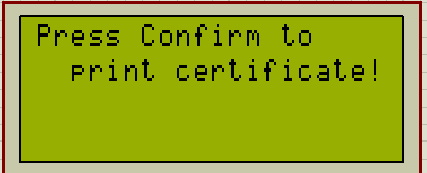






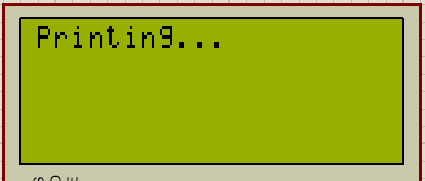
Hình . Giao diện thông tin giấy chứng nhận

* Giao diện xác nhận in ra giấy chứng nhận sinh viên.



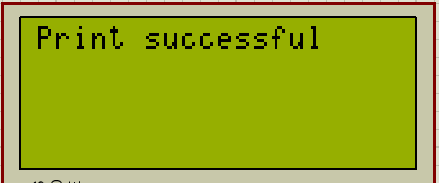
Hình . Giao diện xác nhận in giấy chứng nhận

* Giao diện thông báo đang in giấy chứng nhận sinh viên.



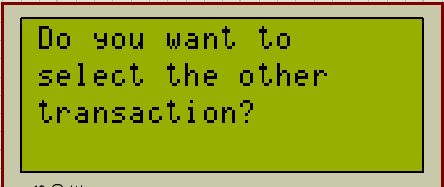
Hình . Giao diện đang in giấy

* Giao diện thông báo in giấy chứng nhận thành công.



Hình . Giao diện thông báo in thành công

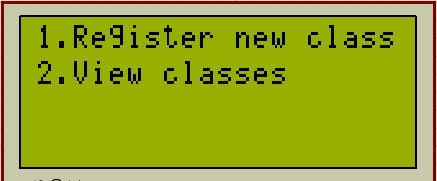
* Giao diện xác nhận có muốn tiếp tục giao dịch hay không.



Hình . Giao diện tiếp tục giao dịch hoặc thoát

## Kết quả mô phỏng của chức năng Đăng ký mở rộng lớp

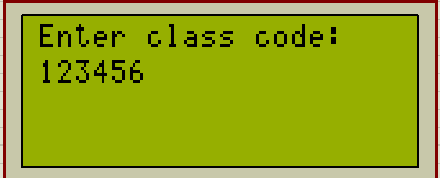
* Giao diện hiển thị các chức năng con.



Hình . Giao diện chọn chức năng con

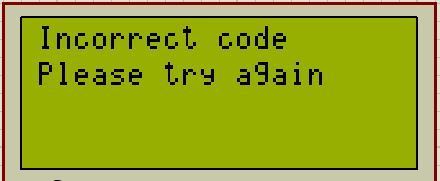
### Chức năng Đăng ký lớp mới

* Giao diện cho phép nhập mã lớp học phần.



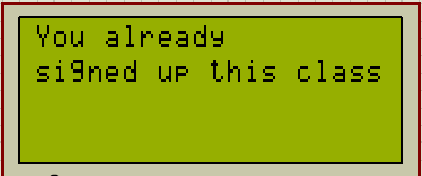
Hình . Giao diện nhập mã lớp

* Giao diện thông báo nhập sai mã lớp.



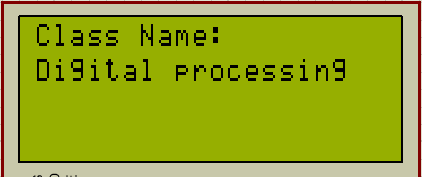
Hình . Giao diện lỗi mã lớp 1

* Giao diện thông báo nhập mã lớp đã đăng ký rồi.



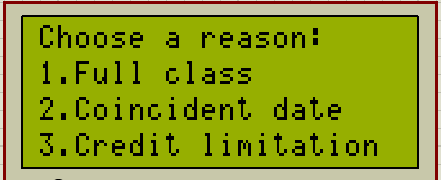
Hình . Giao diện lỗi mã lớp 2

* Giao diện hiển thị tên lớp đăng ký.



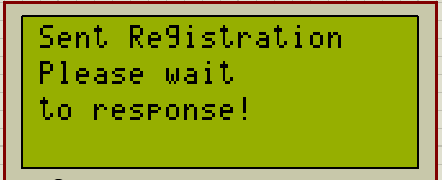
Hình . Giao diện tên lớp đăng ký

* Giao diện hiển thị lý do đăng ký lớp học.



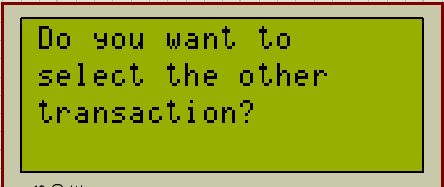
Hình . Giao diện chọn lý do đăng ký lớp

* Giao diện thông báo gửi đăng ký lớp thành công.



Hình . Giao diện đăng ký lớp thành công

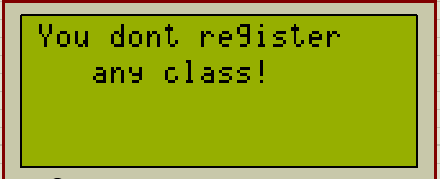
* Giao diện xác nhận có muốn tiếp tục giao dịch hay không.



Hình . Giao diện tiếp tục giao dịch hoặc thoát

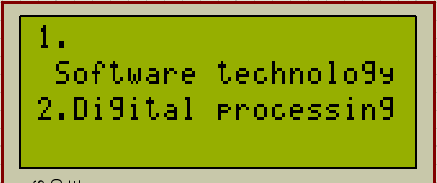
### Chức năng Xem lớp đã đăng ký

* Giao diện thông báo chưa đăng ký lớp học nào.



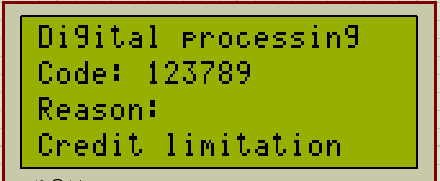
Hình . Giao diện chưa đăng ký lớp

* Giao diện hiển thị các lớp đã đăng ký.



Hình . Giao diện các lớp đã đăng ký

* Giao diện hiển thị thông tin của một lớp đã đăng ký.



Hình . Giao diện thông tin lớp đăng ký

# ĐÁNH GIÁ HỆ THỐNG

Chương 6 đưa ra một vài nhận xét về thành phần trong hệ thống.

## Đánh giá về sơ đồ mạch

* Sơ đồ mạch tương đối đơn giản dùng để phục vụ việc mô phỏng máy giao dịch giải quyết thủ tục sinh viên.
* Trong sơ đồ mạch hệ thống, các phần tử được kết nối trực tiếp với nhau điều này làm tốn nhiều chân của vi điều khiển STM32.
* Trong tương lai, có thể giải quyết vấn đề trên bằng cách sử dụng các mạch tích hợp cho phép giao tiếp I2C hay SPI. Các mạch tích hợp này giúp làm giảm số chân cần sử dụng của vi điều khiển STM32.

## Đánh giá về quá trình mô phỏng

* Tốc độ hiển thị dữ liệu lên màn hình khá nhanh, đúng như yêu cầu đặt ra.
* Đôi lúc mô phỏng bị dừng đột ngột do phần mềm Proteus bị quá tải.
* Các nút bấm của Keypad trên mô phỏng hoạt động đúng. Đôi khi nhấn nút tiếp theo, vi điều khiển bị nhận sai nút do phần mềm Proteus chưa tải kịp.

PHỤ LỤC

Mã nguồn, báo cáo, sơ đồ mạch, lưu đồ thuật toán của hệ thống và các hình ảnh trong báo cáo đều có sẵn tại đường dẫn sau:

<https://github.com/duycu99zxcv123/transaction_machine.git>, (link Github).