**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**KHOA ĐIỆN – ĐIỆN TỬ**

****

BÁO CÁO TỔNG KẾT

Đề tài nghiên cứu khoa học sinh viên

Năm học 2023 - 2024

**HỆ THỐNG TÍNH CƯỚC PHÍ**

**ĐƯỜNG SẮT METRO SỬ DỤNG RFID**

**GVHD: Ths. Võ Quang Sơn**

SVTH: 1. Nguyễn Quý Tuấn – MSV: 201404133 Khoa: Điện – Điện tử

2. Nguyễn Võ Quốc Khánh – MVS: 201403986 Khoa: Điện – Điện tử

3. Nguyễn Quang Minh – MSV: 201404024 Khoa: Điện – Điện tử

4. Bạch Đăng Ngọc Tuấn – MSV: 201405577 Khoa: Điện – Điện tử

5. Nguyễn Đức Huy – MSV: 201403970 Khoa: Điện – Điện tử

Hà Nội – 05/2024

TRƯỜNG ĐH GIAO THÔNG VẬN TẢI

**KHOA ĐIỆN-ĐIỆN TỬ**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM ĐỘC LẬP - TỰ DO - HẠNH PHÚC**

**----o0o----**

HÀ NỘI, ngày 06 tháng 05 năm 2024

### NHIỆM VỤ NGHIÊN CỨU KHOA HỌC

###### TÊN ĐỀ TÀI: HỆ THỐNG TÍNH CƯỚC PHÍ ĐƯỜNG SẮT METRO SỬ DỤNG RFID

1. NHIỆM VỤ
   1. Các số liệu ban đầu:
      * Các hệ thống sử dụng RFID trong thu thu cước giao thông tự động.
      * Các mô hình ứng dụng RFID trong giao thông ở các nước tiên tiến trên thế giới.
      * Các module có sẵn trên thị trường để có thể mô phỏng mô hình thu phí sử dụng RFID (Arduino Mega, RFID RC522...).
   2. Nội dung thực hiện:
      * Thiết kế và thi công phần cứng:
        + Thiết kế, thi công khối điều khiển.
        + Thiết kế thi công khối hiển thị.
        + Lập trình phần cứng
      * Thiết kế và thi công phần mềm:
        + Thiết kế phần giao diện quản lý.
        + Thiết kế cơ sở dữ liệu lưu trữ.
      * Chạy thử và chỉnh sửa, viết báo cáo luận văn.
      * Báo cáo đề tài tốt nghiệp
2. NGÀY GIAO NHIỆM VỤ: 06/02/2024
3. NGÀY HOÀN THÀNH NHIỆM VỤ: 8/05/2024
4. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: **Ths. Võ Quang Sơn**

### LỜI CẢM ƠN

Trong quá trình làm đề tài nghiên cứu, chúng em đã nhận được sự đóng góp, chỉ bảo chân thành của các thầy cô giáo bộ môn Kỹ thuật điện tử. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc nhất đến thầy **ThS. Võ Quang Sơn**, người đã tận tình chỉ bảo chúng em trong suốt thời gian làm đề tài nghiên cứu.

*Chúng em xin chân thành cảm ơn!*

### MỤC LỤC

[LỜI CẢM ƠN i](#_Toc165972885)

[MỤC LỤC ii](#_Toc165972886)

[LIỆT KÊ HÌNH VẼ iv](#_Toc165972887)

[LIỆT KÊ BẢNG VẼ vi](#_Toc165972888)

[Chương 1: TỔNG QUAN 1](#_Toc165972889)

[1.1 ĐẶT VẤN ĐỀ 1](#_Toc165972890)

[1.2 MỤC TIÊU 2](#_Toc165972891)

[1.3 NỘI DUNG NGHIÊN CỨU 3](#_Toc165972892)

[1.4 GIỚI HẠN 3](#_Toc165972893)

[1.5 BỐ CỤC 3](#_Toc165972894)

[Chương 2: CƠ SỞ LÝ THUYẾT 5](#_Toc165972895)

[2.1 TỔNG QUAN HỆ THỐNG THANH TOÁN CƯỚC PHÍ ĐƯỜNG SẮT METRO 5](#_Toc165972896)

[2.1.1 Tìm hiểu hệ thống đường sắt Metro 5](#_Toc165972897)

[2.1.2 Tìm hiểu về các yêu cầu cơ bản của hệ thống thanh toán cƣớc đƣờng sắt Metro 5](#_Toc165972898)

[2.1.3 Mô hình tổng quan 6](#_Toc165972899)

[2.2 GIỚI THIỆU HỆ THỐNG RFID 7](#_Toc165972900)

[2.2.1 Công nghệ RFID 7](#_Toc165972901)

[2.2.2 Thành phần hệ thống RFID 9](#_Toc165972902)

[2.2.3 Ưu nhược điểm của hệ thống RFID 9](#_Toc165972903)

[2.3 GIỚI THIỆU PHẦN CỨNG 10](#_Toc165972904)

[2.3.1 Module thẻ RFID RC522 10](#_Toc165972905)

[2.3.2 Thẻ RFID 12](#_Toc165972906)

[2.3.3 Bộ vi xử lý Arduino MEGA 2560 14](#_Toc165972907)

[2.3.4 LCD 16x2 17](#_Toc165972908)

[2.3.5 Module LCD I2C 18](#_Toc165972909)

[Chương 3: TÍNH TOÁN VÀ THIẾT KẾ 19](#_Toc165972910)

[3.1 GIỚI THIỆU 19](#_Toc165972911)

[3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG 19](#_Toc165972912)

[3.2.1 Thiết kế sơ đồ khối hệ thống 19](#_Toc165972913)

[3.2.2 Tính toán và thiết kế mạch 20](#_Toc165972914)

[Chƣơng 4: THI CÔNG HỆ THỐNG 23](#_Toc165972915)

[4.1 GIỚI THIỆU 23](#_Toc165972916)

[4.2 THI CÔNG HỆ THỐNG 23](#_Toc165972917)

[4.2.1 Thi công mô hình 23](#_Toc165972918)

[4.2.2 Lắp ráp và kiểm tra 23](#_Toc165972919)

[4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH 25](#_Toc165972920)

[4.3.1 Đóng gói và thi công bộ điều khiển 25](#_Toc165972921)

[4.3.2 Thi công mô hình 26](#_Toc165972922)

[4.4 LẬP TRÌNH HỆ THỐNG 26](#_Toc165972923)

[4.4.1 Lưu đồ giải thuật 26](#_Toc165972924)

[4.4.2 Phần mềm lập trình cho vi điều khiển Arduino IDE 29](#_Toc165972925)

[4.4.3 Phần mềm lập trình cho máy tính 30](#_Toc165972926)

[4.4.4 Phần mềm SQL Server Management Studio 31](#_Toc165972927)

[4.5 TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC CỦA HỆ THỐNG 37](#_Toc165972928)

[4.5.1 Hướng dẫn sử dụng 37](#_Toc165972929)

[4.5.2 Quy trình thao tác 38](#_Toc165972930)

[Chương 5: KẾT QUẢ VÀ NHẬN XÉT 41](#_Toc165972931)

[5.1 KHỐI ĐỌC DỮ LIỆU 41](#_Toc165972932)

[5.2 KHỐI VI ĐIỀU KHIỂN 41](#_Toc165972933)

[5.3 CÁC KHỐI KHÁC 41](#_Toc165972934)

[5.4 KẾT QUẢ THỰC HIỆN 41](#_Toc165972935)

[5.4.1 Kết quả thi công phần cứng 41](#_Toc165972936)

[5.4.2 Hình ảnh mô phỏng 42](#_Toc165972937)

[5.4.3 Đánh giả kết quả mô phỏng 46](#_Toc165972938)

[Chương 6: KẾT LUẬN VÀ PHÁT TRIỂN 47](#_Toc165972939)

[6.1 KẾT LUẬN 47](#_Toc165972940)

[6.2 HƯỚNG PHÁT TRIỂN 47](#_Toc165972941)

[TÀI LIỆU THAM KHẢO 48](#_Toc165972942)

[PHỤ LỤC 49](#_Toc165972943)

# LIỆT KÊ HÌNH VẼ

*Hình 2.1 Mô hình hoạt động 5*

*Hình 2.2 Một số thẻ RFID thông dụng 7*

*Hình 2.3 Giao tiếp giữa thẻ RFID và đầu đọc 7*

*Hình 2.4 Module RFID RC522 9*

*Hình 2.5 Thẻ RFID 11*

*Hình 2.6 Board Arduino Mega 2560 13*

*Hình 2.7 Sơ đồ cấu trúc CPU của atmega32 15*

*Hình 2.8 LCD 16x2 16*

*Hình 2.9 Module LCD I2C 17*

*Hình 3.1 Sơ đồ khối của hệ thống 18*

*Hình 3.2 Module RC522 19*

*Hình 3.3 Sơ đồ kết nối LCD 20*

*Hình 3.4 Sơ đồ nguyên lý 21*

*Hình 4.1 Kiểm tra kết nối Arduino 23*

*Hình 4.2 Khối đọc dữ liệu 23*

*Hình 4.3 Khối hiển thị* *24*

*Hình 4.4 Toàn bộ mô hình phần cứng 25*

*Hình 4.5 Lưu đồ giải thuật vi điều khiển 26*

*Hình 4.6 Lưu đồ giải thuật C# 37*

*Hình 4.7 Phần mềm lập trình cho Arduino 28*

*Hình 4.8 Tạo project trong Visual Studio 29*

*Hình 4.9 Giao diện thiết kế trong Visual Studio 30*

*Hình 4.10 Giao diện lập trình trong Visual Studio 33*

*Hình 4.11 Kết nối Server 31*

*Hình 4.12 Khởi tạo Database mới 32*

*Hình 4.13 Tạo bảng mới 32*

*Hình 4.14 Tạo truy vấn 33*

*Hình 4.15 Bảng DangNhap 33*

*Hình 4.16 Bảng DataHanhKhach 34*

*Hình 4.17 Truy vấn Selec*t 34

*Hình 4.18 Truy vấn Insert 35*

*Hình 4.19 Truy vấn Update 35*

*Hình 4.20 Truy vấn Delete 36*

*Hình 4.21 Quy trình thao tác hệ thống 37*

*Hình 4.22 Giao diện đăng nhập hệ thống 37*

*Hình 4.23 Giao diện chọn ứng dụng 37*

*Hình 4.24 Giao diện quản lý hành khách 38*

*Hình 4.25 Giao diện quản lý hành trình 38*

*Hình 5.1 Phần cứng của hệ thống 40*

*Hình 5.2 Khi thẻ RFID được quét qua đầu đọc, tương ứng tại các ga 40*

*Hình 5.3 Kết quả sau khi thêm thông tin hành khách 41*

*Hình 5.4 Thông tin hành khách sau khi chỉnh sửa 42*

*Hình 5.5 Thông tin hành khách đã không còn sau khi xóa* *42*

*Hình 5.6 Khi thẻ chưa có thông tin* *43*

*Hình 5.7 Hành khách “Nguyễn Quý Tuấn” sau khi thêm thông tin và nạp “10000” vào tài khoản, tiến hành lên ga số 1 43*

*Hình 5.8 Hành khách xuống ga số 2, tiền phí được trừ vào tài khoản 44*

# LIỆT KÊ BẢNG VẼ

*Bảng 2.1 Tập lệnh RFID 11*

*Bảng 2.2 Thông số kỹ thuật board Arduino Mega 2560 14*

*Bảng 2.3 Các chân LCD 16x2 17*

*Bảng 4.1 Danh sách linh kiện 22*

## Chương 1: TỔNG QUAN

#### ĐẶT VẤN ĐỀ

Ngày nay sự phát triển của khoa học công nghệ đang đóng vai trò quan trọng trong đời sống con người. Nhiều công nghệ mới ra đời và được sử dụng trong nhiều lĩnh vực trong đó có ngành vận tải. Một trong những công nghệ đang được ứng dụng ngày càng rộng rãi là RFID. Công nghệ RFID vốn đã được sử dụng rộng rãi trong nhiều ứng dụng như thanh toán siêu thị, quản lý hàng hóa xuất nhập, sử dụng cho bãi giữ xe, các hệ thống bảo mật...

Ở các nước phát triển trên thế giới, công nghệ RFID đã được sử dụng để thanh toán cước đường bộ, xe buýt, tàu điện ngầm, các hệ thống vận tải công cộng... Riêng ở Việt Nam, công nghệ RFID hiện đang được sử dụng rộng rãi trong việc quản lý nhân viên ở các công ty và bãi đổ xe. Sắp tới, tại TP Hà Nội sẽ có tuyến đường sắt metro Hà Nội và các tuyến đường sắt khác sẽ được đưa vào hoạt động sẽ mở ra nhiều cơ hội mới để ứng dụng công nghệ RFID vào quản lý, tính cước di chuyển. Và được kỳ vọng sẽ làm cho việc quản lý kiểm và thanh toán sẽ trở nên thuận tiện và dễ dàng hơn, khắc phục được các nhược điểm cũng như tiết kiệm so với phương thức mua vé bằng vé giấy truyền thống.[5]



Với vốn kiến thức được tích lũy trong quá trình học tập, cũng như mong muốn ứng dụng công nghệ vào việc quản lý hành khách, thành toán tiền vé bắt kịp xu hướng phát triển của xã hội, nhóm chọn đề tài: “HỆ THỐNG THANH TOÁN CƯỚC ĐƯỜNG SẮT METRO SỬ DỤNG RFID”. Đề tài bám sát nhu cầu thực tế của xã hội, mở ra nhiều cơ hội và tiềm năm cho sự phát triển của công nghệ RFID trong lĩn vực giao thông công cộng.

#### MỤC TIÊU

Tìm hiểu về cách quản lý, thanh toán cước phí đang áp dụng ở các nước. Từ đó thiết kế mô hình phần cứng của hệ thống mô phỏng theo mô hình đã tìm hiểu. Thiết kế giao diện quản lý và thanh toán cước cho hành khách. Sau khi đã hoàn thành, đúc kết kiến thức, kinh nghiệm để tìm hướng phát triển với kỳ vọng có thể áp dụng vào thực tế

ở Việt Nam.



#### NỘI DUNG NGHIÊN CỨU

* + - Tìm hiểu, nghiên cứu tổng quan về hệ thống thanh toán cước dùng RFID.
    - Nghiên cứu Arduino MEGA, module RFID RF522.
    - Tìm hiểu ngôn ngữ C#, cơ sở dữ liệu SQL.
    - Thiết kế giao diện, viết code để quản lý hành khách, thanh toán phí.
    - Xây dựng CSDL cho hệ thống.
    - Thiết kế và thi công mô hình phần cứng của hệ thống.
    - Viết code cho vi điều khiển kết nối với các phần cứng.
    - Kiểm tra hiệu chỉnh.
    - Viết báo cáo.

#### GIỚI HẠN

Do đây là đề tài nghiên cứu dựa trên các mô hình hoạt động của các tuyết Metro ở nước ngoài và thông tin ở các dự án Metro ở Việt Nam. Đề tài còn tồn tại một số giới hạn:

* + - Mô hình mô phỏng 2 nhà ga, mỗi ga có một cổng lên xuống.
    - Hệ thống thông báo cho hành khách là 1 LCD 16x2, loa.
    - Cơ sở dữ liệu hoạt động trên 1 máy tính, chưa hoạt động trên nhiều máy

#### BỐ CỤC

* + - Chương 1: Tổng quan

Chương này trình bày đặt vấn đề, mục tiêu, nội dung nghiên cứu, giới hạn và bố cục đề tài.

* + - Chương 2: Cơ sở lý thuyết

Chương này trình bày về quy trình hoạt động của hệ thống. Giới thiệu phần cứng các thiết bị được sử dụng trong đề tài.

* + - Chƣơng 3: Thiết kế tính toàn

Chương này trình bay giới thiệu hệ thống, tính toán thiết kế phù hợp, sơ đồ nguyên lý.

* + - Chương 4: Thi công hệ thống

Chương này trình bày thi công hệ thống, thi công mô hình. Quá trình lập trình hệ thống, lưu đồ giải thuật. Hướng dẫn sử dụng hệ thống.

* + - Chương 5: Kết quả, nhận xét, đánh giá

Chương này trình bày kết quả đạt được, nhận xét các kết quả đạt được từ mô hình, mức độ hoàn thiệt đề tài.

* + - Chương 6: Kết luận, hướng phát triển

Chương này trình bày kết luận sau khi thực hiện đề tài. Nếu được hướng phát triển cho đề tài.

## Chương 2: CƠ SỞ LÝ THUYẾT

#### TỔNG QUAN HỆ THỐNG THANH TOÁN CƯỚC PHÍ ĐƯỜNG SẮT METRO

##### Tìm hiểu hệ thống đường sắt Metro

Metro là hệ thống vận tải trong đô thị thành phố, thường thị chạy trên đường ray và có điểm giống với tàu hỏa, có phần lớn di chuyển dưới lòng đất và cả trên cao. Metro là hệ thống chở khách với tốc độ cao trên đường ray, nhiều lượt, chuyến trong ngày. Hiện nay các thành phố lớn trên thế giới đều có hệ thống metro và trở thành một trong những phương tiện di chuyển nhanh chóng, tiện lợi và văn minh của người dân.

Ở Việt Nam thì các tuyển metro đã được nghiên cứu và lên kế hoạch xây dựng từ khoảng hơn vài năm trước để đáp ứng như cầu về phương tiện công cộng cũng như sự phát triển xã hội ở các thành phố lớn. Sau khi khánh thành và đi vào hoạt động, các tuyến metro ở Hà Nội được kỳ vọng sẽ trở thành phương tiện di chuyển chính, giải quyết các bài toán ùn tắt giao thông, ô nhiễm môi trường do khí thải của các phương tiện cá nhân gây ra như xe máy, xe ô tô. Ngoài ra còn thay đổi bộ mặt của thành phố, khiến thành phố trở nên văn minh, hiện đại hơn, từ đó khai thác các nguồn lực về du lịch. Ở TP. Hà Nộisắp tới sẽ có tuyến đường sắt Metro số 1 Hà Nội được đưa vào hoạt động từ năm 2023.

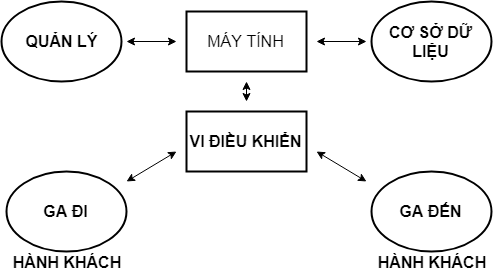
##### Tìm hiểu về các yêu cầu cơ bản của hệ thống thanh toán cƣớc đƣờng sắt Metro

Ở các thành phố trên thế giới, hệ thống Metro là một hệ thống vận tải công cộng có nhu cầu sử dụng rất cao, vì thế nên việc thiết kế hệ thống tính cước phí và quản lý hành khách là một yêu cầu rất quan trọng. Trong nội dung đề tài, nhóm đã thực hiện nghiên cứu khách quan, từ đó mô phỏng một hệ thống tính cước phí đường sắt Metro có một số yêu cầu cơ bản sau đây:

* + - * Đáp ứng được nhu cầu sử dụng của hành khách trong việc sử dụng thẻ để đi lại, cũng như nhà quản lý trong việc thêm, sửa, xóa dữ liệu về hành khách tham gia.
      * Hệ thống phải luôn cập nhật theo thời gian thực, nhằm phục vụ một lưu lượng lớn người sử dụng trong cùng một thời gian.
      * Lưu được thông tin với số lượng lớn.
      * Việc trừ cước phí theo quãng đường di chuyển phải được thực hiện một cách tự động và chính xác nhờ máy tính.

##### Mô hình tổng quan

Sau khi nghiên cứu tổng quan các hệ thống Metro được sử dụng ở các thành phố trên thế giới, nhóm đã đưa ra mô hình mô phỏng cơ bản của hệ thống như sau:



*Hình 2.1 Mô hình hoạt động*

Mô tả hoạt động:

* + - * Để sử dụng được hệ thống Metro, trước tiên hành khách phải đăng ký với cơ quan quản lý để được cấp một thẻ RFID, sau đó tiến hành đăng ký thông tin và nộp tiền vào tài khoản và sử dụng để đi lại.
      * Khi bắt đầu sử dụng, hành khách phải sử dụng thẻ RFID được cấp quét vào đầu đọc thẻ. Sau đó hệ thống sẽ đọc được ID của RFID, tiến hành quét lên hệ thống cơ sở dữ liệu (CSDL) để kiểm tra xem thông tin hành khách có tồn tại, và số dư trong tài khoản có đủ để thực hiện hành trình hay không. Nếu tất cả điều kiện đều thỏa thì hệ thống sẽ thông báo cho phép hành khách lên tàu và bắt đầu hành trình.
      * Sau khi tới ga xuống, hành khách cũng sẽ sử dụng thẻ RFID để quét qua đầu đọc thẻ, sau đó hệ thống trừ tiền vào tài khoản của hành khách.

#### GIỚI THIỆU HỆ THỐNG RFID

##### Công nghệ RFID

Công nghệ RFID (Radio Frequency Identification) là công nghệ nhận dạng đối tượng bằng sóng vô tuyến, cho phép một thiết bị đọc có thể đọc thông tin chưa trong một thiết bị khác ở một khoảng cách gần mà không cần phải có một sự tiếp xúc vật lý nào. Một hệ thống RFID thường bao gốm 2 phần chính là thẻ tag (chip RFID chứa thông tin) và bộ đọc (reader) đọc các thông tin trên chip.

Kỹ thuật RFID sử dụng truyền thông không dây trong tải tần sóng vô tuyến để truyền dữ liệu từ các thẻ tag đến bộ đọc. Bộ đọc dữ liệu của tag và gửi thông tin để hệ thống để xử lý trên cơ sở dữ liệu.

Dạng đơn giản và phổ biến nhất được sử dụng hiện nay là hệ thống RFID bịu động. Trong đó bộ đọc truyền một tín hiệu tần số vô tuyến thông qua anten đến một con chip, sau đó bộ đọc sẽ nhận lại thông tin phản hồi từ chip và gửi đến máy tính để xử lý thông tin. Các con chip từ các thẻ tag này không cần nguồn nuôi, chúng sử dụng năng lượng phát ra từ tín hiệu được gửi bởi bộ đọc. [1]

Thẻ RFID là thiết bị có thể lưu trữ và truyền dữ liệu về bộ đọc bằng sóng vô tuyến. Trong đó các thẻ thường lưu trữ thông tin về các sản phẩm nào đó.

Dữ liệu có thể là một số nhận dạng đơn giản được lưu trữ trong một thẻ chỉ đọc hoặc dữ liệu phức tạp hơn. Các thẻ phức tạp hơn này có thể chứa được các dữ liệu về ngày sản xuất số serial, hoặc thậm chí một số loại đặc biệt còn chưa các cảm biến để theo dõi nhiệt độ trung bình hoặc các loại dữ liệu khác.

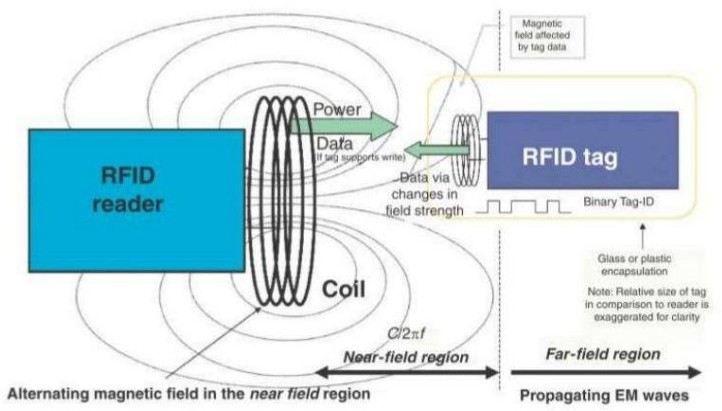
Thẻ RFID gồm chip bán dẫn nhỏ (bộ nhớ của chip có thể chứa từ 96 đến 512 bit dữ liệu, nhiều gấp 64 lần so với mã vạng) và anten được thu nhỏ trong một số hình thức đóng gói. Vài thẻ RFID giống như những nhãn giấy và được ứng dụng để bỏ vào hộp và đóng gói. Một số khác được sản xuất thành các miếng da bao cổ tay. Mỗi thẻ

được lập trình với một nhân jdạng duy nhất cho phép theo dõi không dây đối tượng hoặc con người đang gắn thẻ đó.



*Hình 2.2 Một số thẻ RFID thông dụng*

Khi thẻ đi vào vùng sóng điện từ, nó sẽ phát hiện ra tín hiệu kích hoạt từ đầu đọc và sẽ phát thông tin nhận dạng đến đầu đọc. Đầu đọc giải mã dữ liệu được mã hóa trong chip (sóng vô tuyến phản xạ từ thẻ) và gửi vào hệ thống để xử lý.



*Hình 2.3 Giao tiếp giữa thẻ RFID và đầu đọc*

##### Thành phần hệ thống RFID

Một hệ thống RFID bao gồm các thành phần sau:

* + - * Reader: là thành phần bắt buộc, thường được tích hợp sẵn cả anten.
      * Thẻ tag: là thành phần bắt buộc với mọi hệ thống RFID.
      * Thiết bị xử lý: bao gồm các vi xử lý có khả năng nhận được mã tag được gửi về từ reader, sau đó gửi lên hệ thống.
      * Ngoài ra các hệ thống lớn còn được kết nối với các máy tính, hạ tầng mạng để truyền nhận thông tin của thẻ tag, thực hiện các tác vụ như liên kết tài khoản, thông tin, tiền phí... Cũng như có các cơ cấu chấp hành để thực thi các yêu cầu đặt ra với hệ thống.

##### Ưu nhược điểm của hệ thống RFID

* + - * Ưu điểm:
        + Đọc với tốc độ cao mà không cần tiếp xúc vật lý: nhiều đối tượng có thể được quét tại cùng một thời điểm, có thể lên đến 40 thẻ trong 3 giây. Làm giảm thời gian hoạt động, tăng năng suất của hệ thống.
        + Khả năng đọc và ghi dữ liệu nhiều lần: một số thẻ cho phép đọc và ghi dữ liệu nhiều lần, từ đó làm giảm chi phí hoạt động của hệ thống, cũng như của người sử dụng.
        + Nhỏ gọn, bền: các thẻ RFID hoạt động khá tốt trong môi trường không thuận lợi (nóng ẩm, bụi bẩn...).
        + Một số thẻ RFID, đặc biệt là các thẻ thụ động không cần phải cung cấp nguồn để có thể hoạt động, từ đó nâng cao tính tiện lợi của hệ thống.
        + Việc áp dụng công nghệ RFID vào các lĩnh vực của đời sống làm tăng năng suất lao động, đồng thời tự động hoát các quy trình, sản xuất, thay thế các hoạt động đòi hỏi việc phải lặp đi lặp lại với tần suất cao của con người, từ đó giảm thiểu, triệt tiêu những sai sót có thể xảy ra.
      * Nhược điểm:
        + Khả năng kiểm soát các thiết bị còn hạn chế: thẻ dễ bị nhiều sóng trong môi trường nước và kim loại.
        + Các đầu đọc có thể đọc chồng lên nhau: vì nhiệm vụ của các đầu đọc thẻ là gửi tín hiệu đến các thẻ tag, sau đó nhận tín hiệu gửi về, vì thể trong một số trường hợp có thể xảy ra việc đọc chồng chéo lên nhau.
        + Giá thành của hệ thống RFID hiện nay vẫn còn khá cao, xét đến tính thực tế ở Việt Nam thì vẫn chưa thể ứng dụng rộng rãi trong mọi lĩnh vực.

#### GIỚI THIỆU PHẦN CỨNG

##### Module thẻ RFID RC522

Module RFID sử dụng là RFID RC522.

Module RFID RC522 sử dụng IC MFRC522 của Phillips dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56 Mhz, với mức giá rẻ, thiết kế nhỏ gọn, module này là sự lựa chọn hàng đầu cho các ứng dụng về ghi đọc thẻ RFID

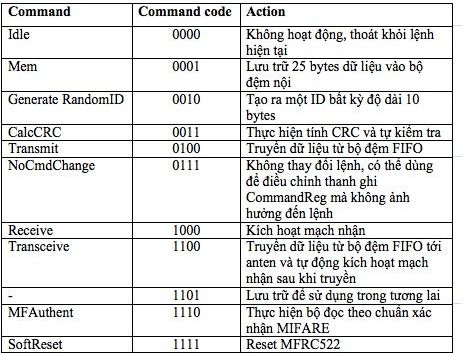


*Hình 2.4 Module RFID RC522*

* + - * Điện áp sử dụng: 3.3 V, 13 - 26 mA.
      * Dòng tiêu thụ ở chế độ standby: 3.3 V, 10 - 13 mA.
      * Sleep mode: <80 µA.
      * Tải tối đa: 30 Ma.
      * Tần số hoạt động: 13.56 Mhz.
      * KHoảng cách đọc: 0 - 60 mm.
      * Giao thức truyền thông: SPI.
      * Tốc độ dữ liệu tối đa: 10 Mbit/s.
      * Kích thước: 40 x 60mm.
      * Nhiệt độ hoạt động: -20 -> 80°C
      * Độ ẩm hoạt động: 5% - 95 %.
      * Hỗ trợ ISO/IEC 14443A/MIFARE.

Chân kết nối RC522:

* + - * SDA (CS): chân lựa chọn chip khi giao tiếp SPI (kích hoạt mức thấp).
      * SCK: chân xung của SPI.
      * MOSI (SDI) Master Data Out - Slave in trong SPI.
      * MISO (SDO) Master Data Out - Slave out trong SPI.
      * IRQ: chân ngắt.
      * GND chân mass.
      * RST chân reset.
      * Nguồn 3.3V.



*Bảng 2.1 Tập lệnh RFID*

##### Thẻ RFID



*Hình 2.5 Thẻ RFID*

Thẻ RFID gồm có 2 phần: Chip lưu trữ một số thứ tự duy nhất, hoặc thông tin khác dựa trên loại thẻ và ăng-ten được gắn với vi mạch truyền thông tin từ chip đến bộ đọc, một ăng-ten càng lớn thì khoảng cách đọc càng xa.

Các loại thẻ RFID được chia làm 3 loại cơ bản và thông dụng là:

* + - * Thẻ thụ động (Passive tag): Thẻ thụ động hay nhãn dán thụ động là thẻ không có nguồn năng lượng. Ngay khi mà dòng điện được gây ra bởi những tín hiệu sóng radio đi vào trong ăng-ten cung cấp đủ năng lượng cho mạch tích hợp CMOS (IC) trong thẻ, mạch bắt đầu hoạt động và thẻ truyền tín hiệu phản hồi lại. Điều này có nghĩa là khi thẻ thụ động đi qua máy đọc, năng lượng của sóng radio phát từ máy đọc sẽ cung cấp năng lượng cho chip và đánh thức nó để thu nhận thôgn tin mà nó lưu giữ. Điều này có nghĩa là ăng-ten phải thiết kế để thu năng lượng từ cả hai tín hiệu đến và tín hiệu phản lại truyền ra. Chính vì nó không có nguồn nuôi bên trong thẻ, nên những thẻ thụ động và nhãn dán thụ động có kích thước khá nhỏ, vì thể nó cũng không có khoảng cách đọc quá xa.
      * Thẻ bán chủ động (Semi-active tag): Thẻ bán chủ động RFID tương đối giống với thẻ thụ động, trừ phần có thêm một pin nhỏ. Pin này cho phép IC của thẻ được cấp nguồn liên tục, giảm bớt sự cần thiết trong thiết kế ăng-ten thu năng lượng từ tín hiệu quay lại. Các thẻ bán thụ động không chủ động truyền tín hiệu vô tuyến về đầu đọc, mà nó nằm im bảo tồn năng lượng cho tới khi nận được tín hiệu vô tuyến từ đầu đọc, nó sẽ kích hoạt hệ thông hoạt động. Thẻ bán chủ động RFID nhanh hơn trong sự phản hồi lại, và vì vậy khỏe hơn trong việc đọc số truyền so với thẻ chủ động. Do đó khoảng cách đọc của nó cũng xa hơn so với thẻ thụ động.
      * Thẻ chủ động (Active tag): Đây là loại thẻ khác so với hai loại trên, thẻ chủ động RFID có nguồn năng lượng trong chính bản thân, được sử dụng để cung cấp nguồn cho tất các các IC và phát ra tín hiệu. Chúng thường được gọi là đèn hiệu bởi vì chúng phát các tín hiệu mà chúng nhận được. Thẻ chủ động có vùng hoạt động rộng hơn, có thể lên tới vài chục mét, trong khi bộ nhớ của nó cũng lớn hơn, cho phép lưu trữ và truyền nhiều dữ liệu hơn.

Trong khuôn khổ đề tài này sử dụng thẻ thụ động (Passive tag).

##### Bộ vi xử lý Arduino MEGA 2560

*Hình 2.6 Board Arduino Mega 2560*

Arduino Mega 2560 là một bo mạch được thiết kế với bộ xử lý trung tâm là vi điều khiển công suất thấp 8 - bit CMOS AVR Atmega2560. Có 54 ngõ vào/ra số, 16 ngõ vào ra analog, thạch anh 16 Mhz, kết nối USB, 1 jack cắm điện, header ICSP, 1 nút reset. Cung cấp mọi thứ cần thiết để hỗ trợ các vi điều khiển, chỉ cần kết nối với máy tính bằng cáp USB hoặc một adapter AC - DC hoặc pin để cấp nguồn.

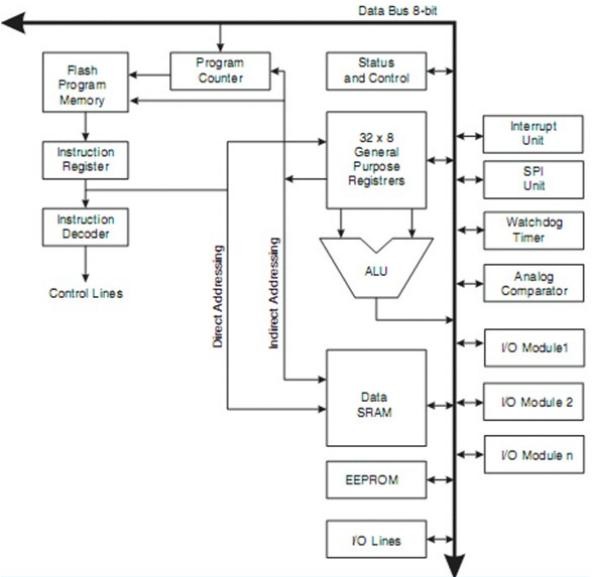
|  |  |
| --- | --- |
| Vi xử lý | Atmega 2560 |
| Điện áp hoạt động | 5V |
| Điện áp đầu vào đề nghị | 7 - 12 V |
| Điện áp vào giới hạn | 6 - 20 V |
| Phạm vi nhiệt độ hoạt động | -40°C đến 80°C |
| Số ngõ vào ra tín hiệu số (digital) | 54 |
| Số ngõ vào ra dạng tương tự (analog) | 16 |
| Dòng điện DC mỗi chân vào ra số (digital) | 40 mA |
| Dòng điện chân nguồn | 50 mA |
| Bộ nhớ | 256 KB |
| Sram | 8 KB |
| EEPROM | 4 KB |
| Xung nhịp | 16 Mhz |

*Bảng 2.2 Thông số kỹ thuật board Arduino Mega 2560*

Arduino là một board mạch vi xử lý, nhằm xây dụng các ứng dụng tương tác nhau hoặc với môi trường được thuật lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tàng vi xử lý AVR Atmel 8 bit hoặc ARM Atmel 32 bit. Kèm theo là một bộ phát triển tích hợp (IDE) cho phép người dùng có thể viết chương trình bằng ngôn ngữ C/C++, cùng với các thư viện được tích hợp sẵn giúp cho việc lập trình được dễ dàng hơn. Ngoài ra, Arduino còn cho phép người dùng kết nối với các module thêm gắn ngoài được gọi là shield. Các board Ardui no sử dụng các shield, các board mạch in mở rộng bằng cách cắm vào các chân cắm của Arduino, một số loại shile thông dụng có thể kết đến như module điều khiển động cơ, LCD, wifi... Ngoài các module được thiết kế sẵn thì chúng ta cũng có thể tự tạo ra các module của riêng mình.

Một vi điều khiển Arduino thông dụng thường được lập trình sẵn với một boot loader cho phép đơn giản hóa việc upload chương trình từ máy tính xuống board mà không cần một bộ mạch nạp bên ngoài, điều này giúp cho việc sử dụng Arduino được thuận tiện và nhanh chóng hơn bằng cách coi máy tính như là một bộ nạp chương trình.

ATmega2560 có bộ nhớ 2560 KB để chứa mã lập trình (trong đó 8KB được sử dụng để nạp khởi động), EEPROM 4KB, SRAM 8KB có thể được đọc và viết với thư viện EEPROM.



*Hình 2.7 Sơ đồ cấu trúc CPU của atmega32*

Arduino được hỗ trợ thông qua kết nối USB với máy tính để nạp chương trình, cổng USB này cũng có thể đóng vai trò như một nguồn cung cấp điện nhỏ cho Arduino khi hoạt động với những kết nối, ứng dụng và các module không tốn quá nhiều năng lượng. Ngoài ra thì với các module, ứng dụng tiêu tốn nhiều năng lượng hơn sẽ cần một nguồn cung cấp ngoài từ 6 đến 20V. Nếu sử dụng lớn hơn 12V, Arduino sẽ bị quá nhiệt và làm hỏng các linh kiện, vì thể phạm vi hoạt động khuyến cáo của Arduino là từ 7 -> 12V.

Ngoài các chân tại vùng cung cấp nguồn như 5V, 3,3V, GND. Arduino Mega

2560 cung cấp các chân tín hiệu sau:

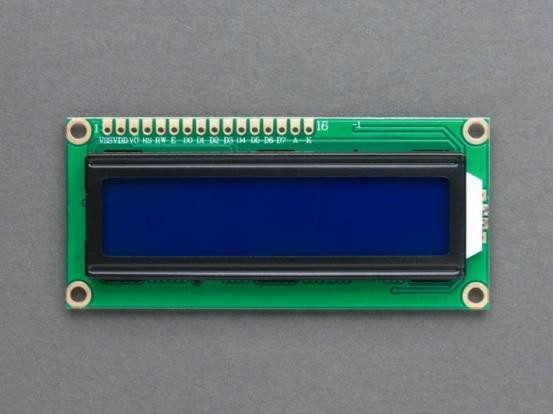
* 54 ngõ vào/ra tín hiệu số sử dụng như một ngõ vào hoặc ngõ ra, trong đó 15 ngõ có thể xuất xung PWM. Mỗi ngõ vào/ra hoạt động ở 5V và có

điện trở kéo lên tích hợp sẵn. Mỗi ngõ vào ra cung cấp hoặc nhận tối đa dòng 40mA.

* PWM: từ chân số 0 đến chân 13, cung cấp 8 bits đầu ra PWM với lệnh analogWrite().
* SPI: Chân 50 (MISO), chân 51 (MOSI), chân 52 SCK, chân 53 (SS) là những chân hỗ trợ chuẩn truyền dữ liệu theo chuẩn SPI thông qua thư viện SPI.h được tích hợp sẵn.
* Serial: 0 (RX) và 1 (TX); Serial 1: 19 (RX) và 18 (TX); Serial 2: 17 (RX) và 16 (TX); Serial 3: 15 (RX) và 14 (TX). Được sử dụng để nhận (RX) và truyền (TX) dữ liệu nối tiếp.
* LED: Chân 13, mặc định mức thấp
* 16 ngõ vào/ra tín hiệu tương tự từ A0 dến A15, cung cấp 10 bit độ phân giải, tức 1024 giá trị khác nhau.
* Arduino được trang bị các cầu chì điện từ resettable polyfuse - một linh kiện điện từ thụ động chống quá dòng để bảo vệ cộng USB máy tính của bạn. Nếu dòng điện lớn 500mA đi vào cổng USB, các cầu chì sẽ tự động ngắt dòng điện.

##### LCD 16x2

LCD có khả năng hiện thị ký tự đa dạng, trực quan (chữ, số và ký tự đồ họa), dễ dàng đưa vào mạch ứng dụng theo nhiều hình thức giao tiếp khác nhau, đồng thời giá thành cũng khá rẻ và tốn ít tài nguyên của hệ thống.

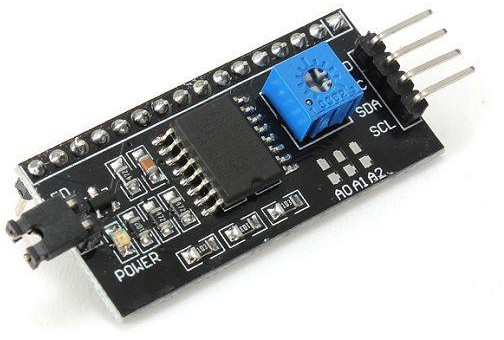


*Hình 2.8 LCD 16x2*

|  |  |  |  |
| --- | --- | --- | --- |
| Chân | Ký hiệu | I/O | Mổ tả |
| 1 | VSS | - | Mass nguồn |
| 2 | VCC | - | Nguồn dương 5V DC |
| 3 | VFE | - | Nguồn điều khiển tương phản |
| 4 | RS | I | RS = 0 chọn thanh ghi lệnh  RS = 1 chọn thanh ghi dữ liệu |
| 5 | R/W | I | R/W = 1 đọc dữ liệu  R/W = 0 ghi dữ liệu |
| 6 | E | I/O | Chân cho phép |
| 7-14 | DB0 - DB7 | I/O | Các bit dữ liệu |
| 15-16 | A - K | - | Điều khiển độ sáng LCD |

*Bảng 2.3 Các chân LCD 16x2*

##### Module LCD I2C



*Hình 2.9 Module LCD I2C*

Thông thường để sử dụng màn hình LCD, chúng ta sẽ mất khá nhiều chân trên Arduino, ngoài ra thì việc sử dụng quá nhiều dây kết nối dễ khiến mô hình trở nên rối và khó xử lý khi có trục trặc xảy ra. Với module LCD I2C, chỉ cần sử dụng 4 chân, trong đó 2 chân cấp nguồn, 2 chân SDA và SCL để giao tiếp với Arduino và đưa thông tin hiển thị ra màn hình LCD.

## Chương 3: TÍNH TOÁN VÀ THIẾT KẾ

#### GIỚI THIỆU

Từ yêu cầu của hệ thống, nhóm thực hiện đưa ra phương án thiết kế hệ thống gồm các khối: đọc dữ liệu, vi xử lý, máy tính để quản lý hệ thống.

#### TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG

##### Thiết kế sơ đồ khối hệ thống

*Hình 3.1 Sơ đồ khối của hệ thống*

Nguyên lý hoạt động:

* + - * Khi có hành khách thực hiện đăng ký, thêm sửa xóa thông tin, mã tag của thẻ sẽ được đọc từ khối đọc dữ liệu, sau đó vi điều khiển có nhiệm vụ kết nối và gửi mã tag nhận được lên phần mêm trên máy tính, máy tính thực hiện các thao tác truy xuất vào cơ sở dữ liệu để thêm, sửa, xóa thông tin cho hành khách.
      * Khi có hành khách quẹt thẻ để lên tàu, mã tag của thẻ sẽ được đọc từ khối đọc dữ liệu, sau đó vi điều khiển nhận được mã tag và gửi thông tin lên phần mêm trên máy tính. Phần mềm trên máy tính tiến hành truy xuất vào cơ sở dữ liệu để kiểm tra các thông tin của hành khách. Nếu các thông tin của hành khách đều hợp lệ sẽ cho phép thực hiện hành trình. Đến khi kết thúc, mã thẻ sẽ được đọc

tại trạm đến, gửi dữ liệu lên phần mềm máy tính để tiến hành trừ phí di chuyển, cập nhật dữ liệu của hành khách. Sau đó, máy tính sẽ gửi tín hiệu về lại vi điều khiển, và thông báo cho hành khách thông qua các ngoại vi.

##### Tính toán và thiết kế mạch

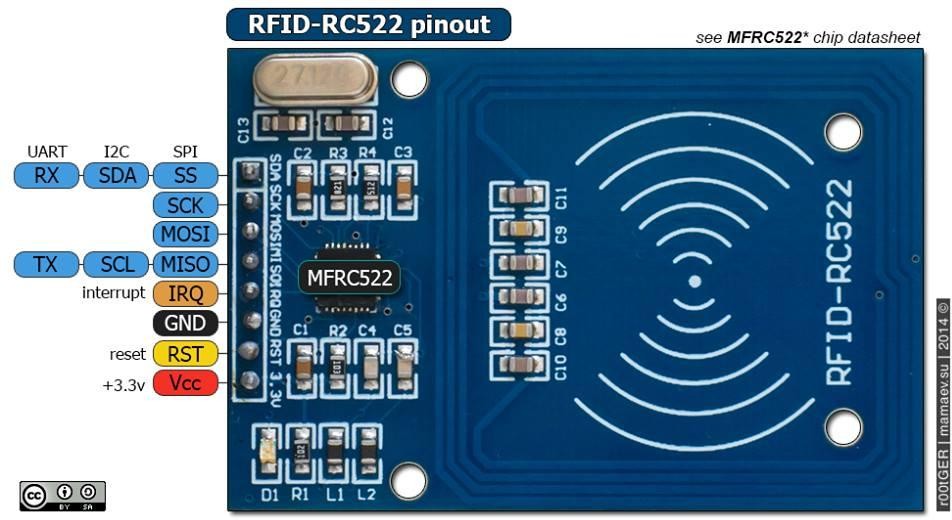
Từ những yêu cầu cảu hệ thống, phần cứng sẽ bao gồm các khối:

1. Khối nguồn:

Sử dụng nguồn DC 9V - 2A để cấp nguồn ngoài cho Arduino, các khối đọc dữ liệu RFID, thông báo sử dụng nguồn 3,3V và 5V trên board Arduino.

1. Khối đọc dữ liệu:

Sử dụng Module RFID RC522 đọc thẻ RFID giao tiếp với các khối vi điều khiển (Arduino Mega 2560) qua giao thức SPI.



*Hình 3.2 Module RC522*

1. Khối vi điều khiển:

Sử dụng Arduino Mega 2560 làm bộ xử lý trung tâm, nhận dữ liệu từ khối đọc dữ liệu gửi về để gửi lên máy tính, nhận dữ liệu từ máy tính để gửi đến khối hiển thị để thông báo cho người sử dụng biết.

1. Khối máy tính:

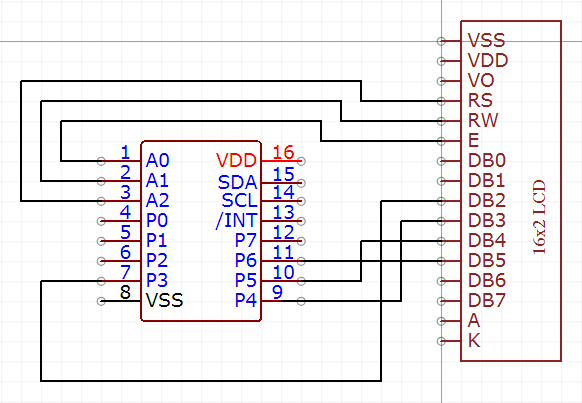
Sử dụng máy tính kết nối trưc tiếp với khối vi điều khiển và tiến hành nhận và gửi dữ liệu về vi điều khiển, thực hiện các chức năng của hệ thống.

Board Arduino và máy tính giao tiếp với nhau thông qua Serial Port, khi có dư xliệu trong output buffer của Arduino sẽ được gửi qua máy tính và lưu ở input buffer.

Máy tính đọc dữ liệu ở input buffer sẽ thực hiện lệnh, đồng tời trên Arduino cũng song song tồn tại phương thức ấy để nhận dữ liệu khi máy tính gửi đi. [5]

1. Khối hiển thị:

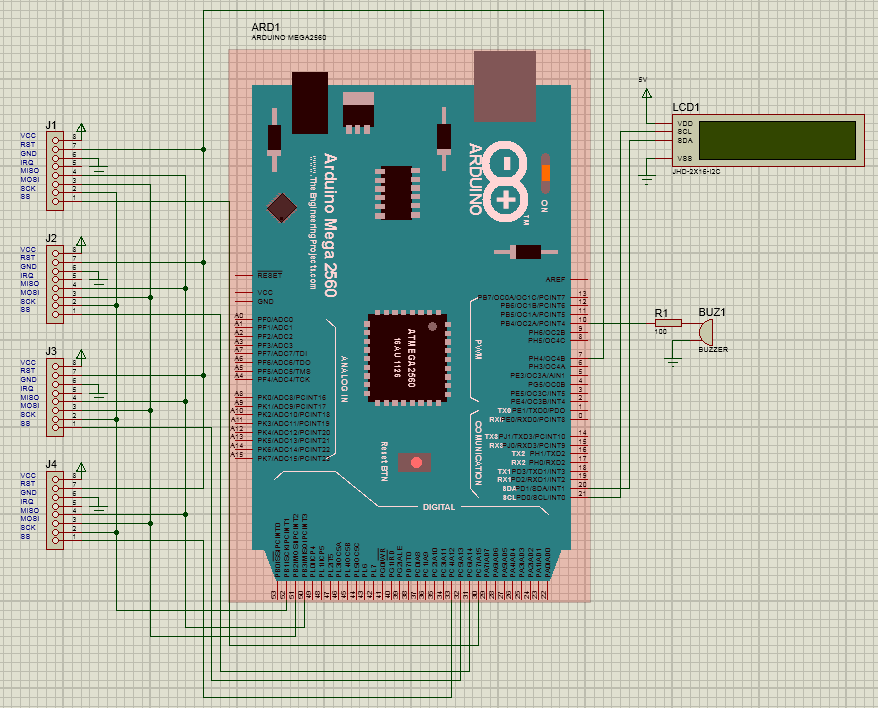
Sử dụng LCD 16x2 kết hợp với module I2C để hiện thị thông tin ra màn hình thông báo cho hành khách.



*Hình 3.3 Sơ đồ kết nối LCD*

* + 1. Sơ đồ nguyên lý của toàn mạch

Sơ đồ nguyên lý của toàn mạch:



*Hình 3.4 Sơ đồ nguyên lý*

* + - * Các module RC522 mô phỏng các nhà ga, mỗi nhà ga có 2 module RC522 đọc thẻ tương ứng với 2 ga lên và xuống.
      * Các module gửi dữ liệu nhận được về Arduino, bắt đầu quá trình xử lý.
      * Khi có thẻ quẹt qua các module RC522, dữ liệu sẽ được gửi về máy tính để xử lý, đồng thời sau khi nhận được dữ liệu gửi về từ máy tính, Arduino sẽ giao tiếp với khối hiển thị thông báo cho hành khách.

## Chƣơng 4: THI CÔNG HỆ THỐNG

#### GIỚI THIỆU

Từ những nghiên cứu trước đó, nhóm thực hiện chia phần thi công hệ thống ra làm 2 phần:

* Thiết kế, thi công phần cứng.
* Thiết kế, viết chương trình cho Arduino, thiết kế phần mềm giao diện quản lý hệ thống và cơ sở dữ liệu.

#### THI CÔNG HỆ THỐNG

##### Thi công mô hình

Các loại linh linh kiện sử dụng trong mô hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên khối | Tên linh kiện | Số lượng | Ghi chú |
| 1 | Nguồn | Nguồn 9V-2A | 1 |  |
| 2 | Vi điều khiển | Arduino Mega 2560 | 1 |  |
| 3 | Đọc dữ liệu | RFID RC522 | 4 | Nối chung các chân SCK, MISO, MOSI  của 4 RFID lại với nhau và nối với Arduino |
| 4 | Hiển thị | LCD 16x2 | 1 | Kết nối theo sơ đồ nguyên lý |
| MOdule I2C | 1 |
| 5 | Thông báo | Loa | 1 |
| Điện trở 100Ω | 1 |

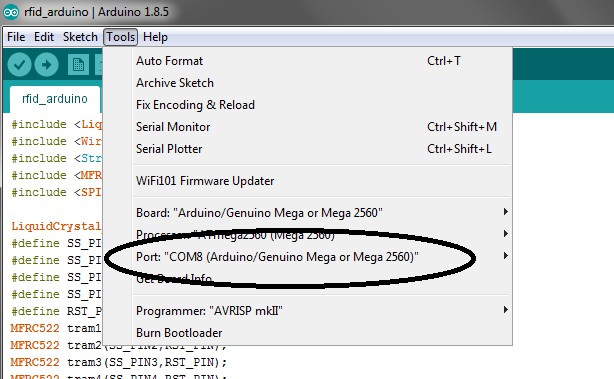
*Bảng 4.1 Danh sách linh kiện*

##### Lắp ráp và kiểm tra

Tiến hành lắp ráp các linh kiện, kết nối các khối với nhau.

* + - 1. Kết nối máy tính:

Kết nối máy tính với Arduino thông qua cổng USB, có thể kiểm tra trực tiếp bằng phần mềm lập trinh cho Arduino là Arduino IDE.



*Hình 4.1 Kiểm tra kết nối Arduino*

Nếu như kết nối thành công, sẽ xuất hiện dòng: “Port: COM8(Arduino/Genuino Mega or Mega2560). Có nghĩa là đã kết nối thành công Arduino với máy tính thông qua cổng COM8.

* + - 1. Kết nối khối đọc dữ liệu RFID:

Các khối được kết nối theo sơ đồ nguyên lý, sau đó cấp nguồn cho Arduino, kiểm tra đèn màu đỏ báo nguồn trên các RFID RC522 xem có hoạt động không, dùng code mẫu và sử dụng Serial Monitor có sẵn của Arduino IDE để kiểm tra xem có hoạt động bình thường hay không.

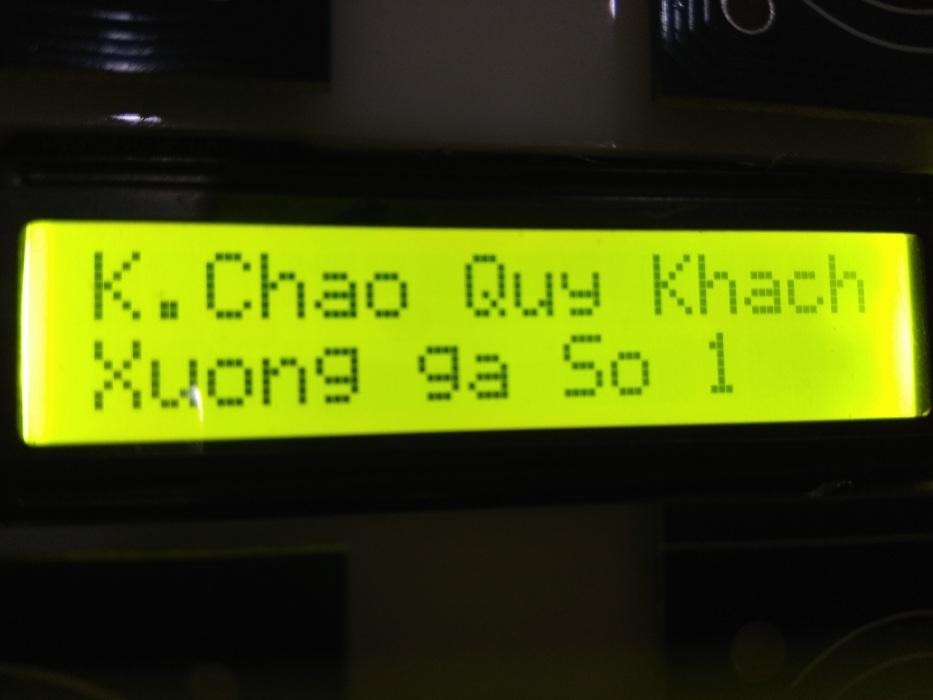


*Hình 4.2 Khối đọc dữ liệu*

lý.

* + - 1. Kết nối khối hiển thị:

LCD sau khi kết nối với module I2C, sẽ kết nối với Arduino theo sơ đồ nguyên



*Hình 4.3 Khối hiển thị*

* + - 1. Khối thông báo:

Sử dụng loa để thông báo khi hành khách quẹt thẻ qua khối đọc dữ liệu

* + - 1. Kết nối nguồn:

Kết nối nguồn vào Arduino, chú ý xem thứ các module kết nối với Arduino có được cấp nguồn ổn định không, có bị chập chờn hay không hoạt động không. Theo dõi thêm vài phút để có thể phát hiện nếu Arduino bị nóng vì quá dòng, nguồn quá cao dẫn đến hư hỏng thiết bị.

#### ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

##### Đóng gói và thi công bộ điều khiển

Sau khi kiểm tra và kết nối các khối, các thiết bị lại với nhau, ta được mô hình

sau:



phỏng.

*Hình 4.4 Toàn bộ mô hình phần cứng*

Mô hình được đóng hộp bằng mica để tăng tính thẫm mỹ cho hệ thống mô

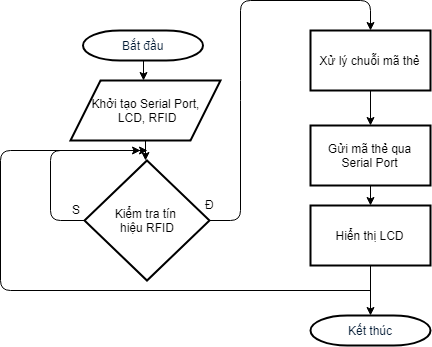
##### Thi công mô hình

Sau khi đã kết nối phần cứng, ta kết nối với máy tính thông qua cổng USB, như vậy ta đã được một mô hình hoàn chỉnh. Sau đó, tiến hành lập trình cho Arduino, xây dựng phần mềm quản lý và cơ sơ dữ liệu.

#### LẬP TRÌNH HỆ THỐNG

##### Lưu đồ giải thuật

* + - 1. Lƣu đồ giải thuật Arduino Mega 2560

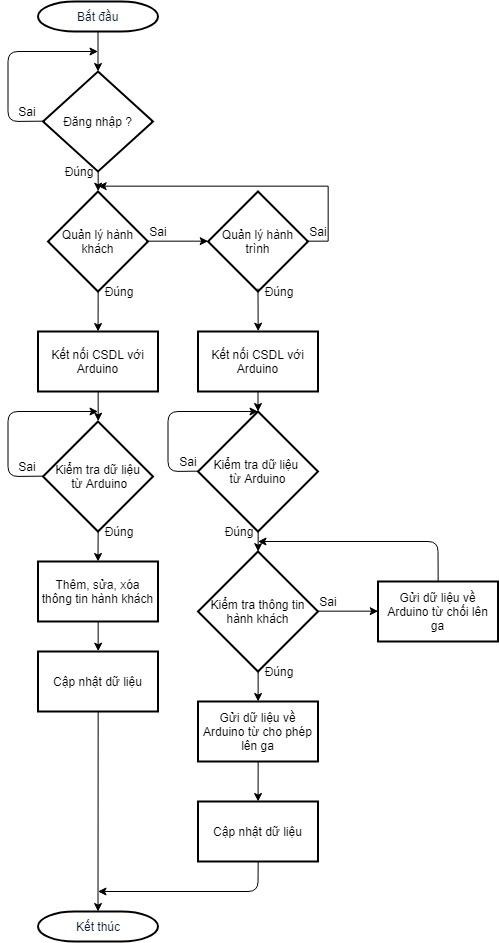


*Hình 4.5 Lưu đồ giải thuật vi điều khiển*

***Giải thích lưu đồ:***

Sau khi khởi tạo các ngoại vi LCD, RFID và Serial Port, kiểm tra liên tục xem thẻ có được đọc hay không, nếu không thì tiếp tục kiểm tra, nếu có thì tiến hành ghép chuỗi và gửi lên máy tính thông qua Serial Port và hiển thị thông tin ở LCD.

* + - 1. Lƣu đồ giải thuật C#:



*Hình 4.6 Lưu đồ giải thuật C#*

##### Phần mềm lập trình cho vi điều khiển Arduino IDE

*Hình 4.7 Phần mềm lập trình cho Arduino*

Môi trường lập trình Arduino IDE có thể chạy trên cả 3 nền tảng phổ biến nhất hiện nay là Windows, MacOS và Linux. Do có tính chất mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Ngôn ngữ lập trình có thể mở rộng thông qua các thư viện C/C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR C vào chương trình nếu muốn.

Để cài đặt Arduino IDE, tùy vào từng hệ điều hành khác nhau mà ta sẽ tải các bản tương thích thông qua trang web chính thức: [http://***www.arduino.cc/***](http://www.arduino.cc/)và tiến hành cài đặt từng bước theo hướng dẫn.

Trước khi thực hiệp lập trình, ta phải lựa chọn loại board mà ta sử dụng, cũng như port được kết nối thông qua menu ***Tools***. Để tạo project mới, ta chọn menu ***File/New***. Sau đó tiến hành lập trình cho Arduino. Trong quá trình viết chương trình, nếu cần thêm các thư viện thì vào menu ***Sketch/Include Library***. Sau khi đã viết xong,

tiến hành biên dịch và kiểm tra lỗi bằng cách chọn menu ***Sketch/Verify/Compile***. Nếu trình biên dịch kiểm tra và không phát hiện lỗi, ta sẽ đổ chương trình xuống board bằng cách chọn menu ***Sketch/Upload***.

##### Phần mềm lập trình cho máy tính

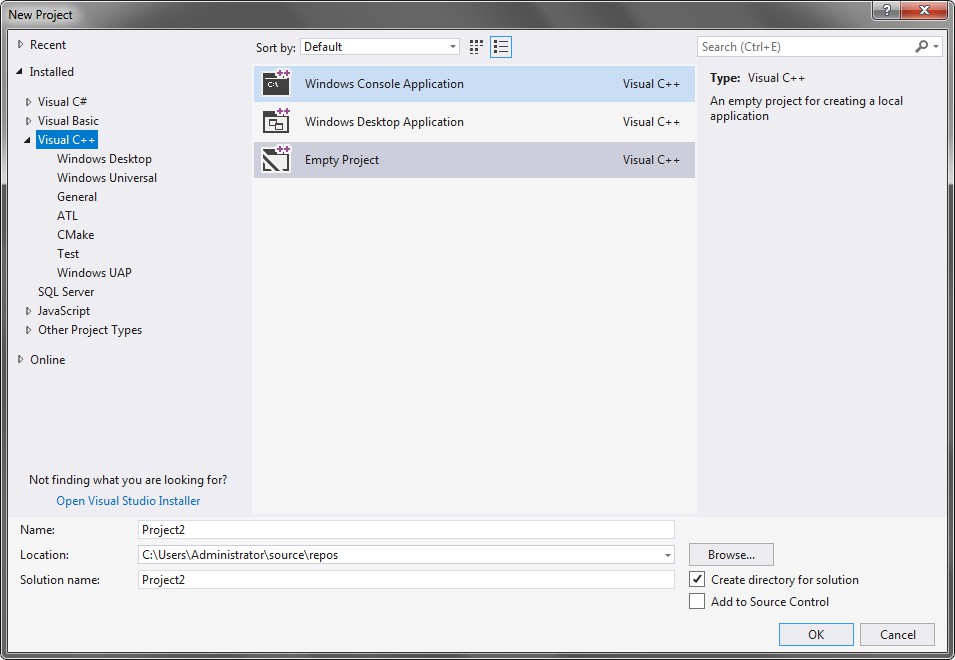
\* Giới thiệu C# và phần mềm lập trình Visual Studio

C# là một ngôn ngữ lập trình hướng đối tượng (object-oriented) và có thể được sử dụng với cho nhiều mục đích khác nhau (general purpose). C# là ngôn ngữ được phát triển bởi Microsoft và là ngôn ngữ được sử dụng phổ biến trong khuôn khổ .NET framework (cùng được phát triển bởi Microsoft).

Sử dụng C# chúng ta có thể xây dựng được các ứng dụng phần mềm chạy trên hệ điều hành Windows, các ứng dụng web service, các ứng dụng mobile, các ứng dụng về database và rất nhiều loại dụng khác nữa.

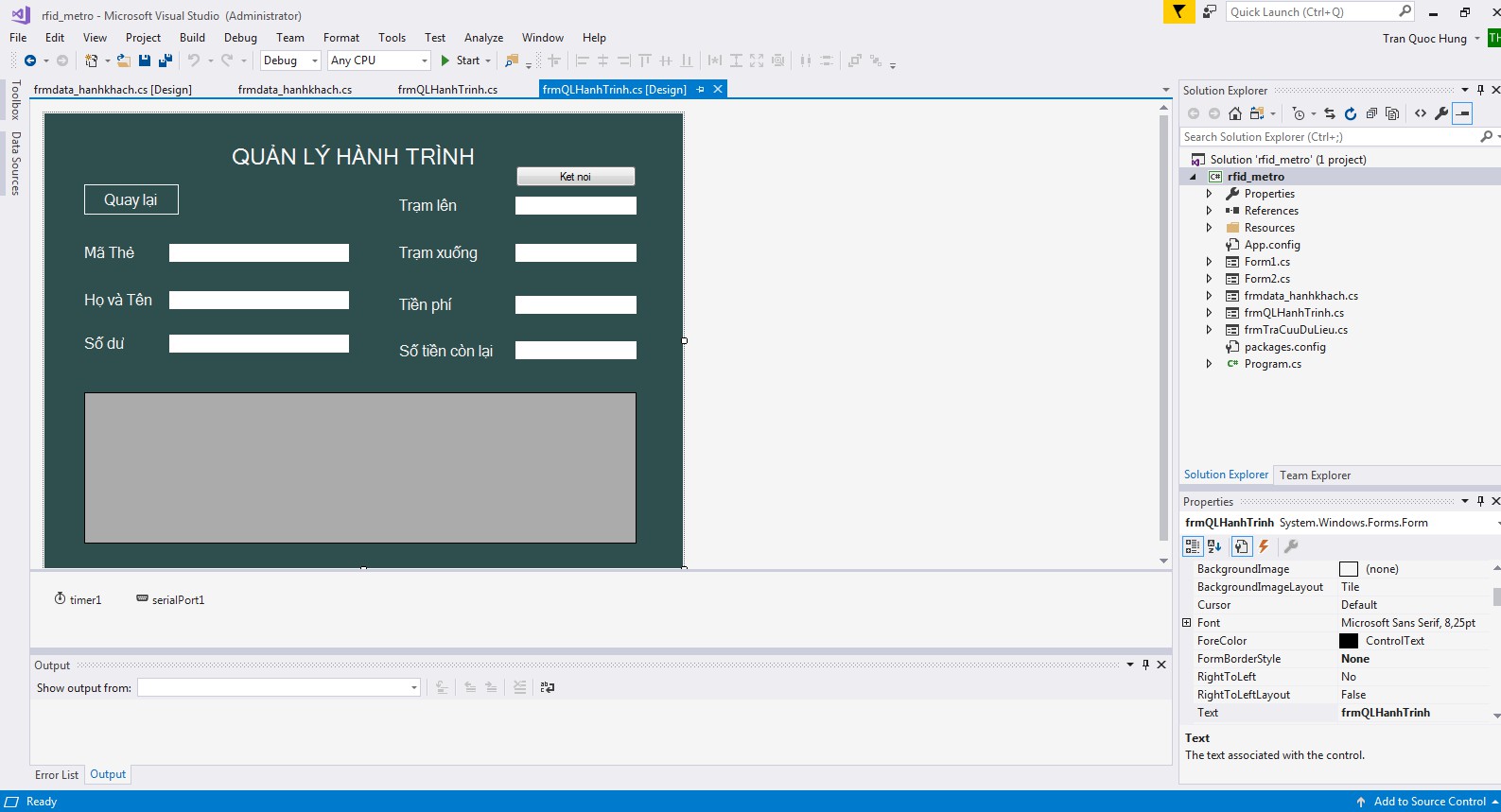
Để cài đặt Visual Studio, ta có thể tải phiên bản Express, hoàn toàn miễn phí và đáp ứng nhu cầu nghiên cứu cũng như công việc.

Để tạo project mới, ta vào ***File/New/Project***, sau đó đặt tên project, nơi lưu trữ và chọn OK để hoàn thành.



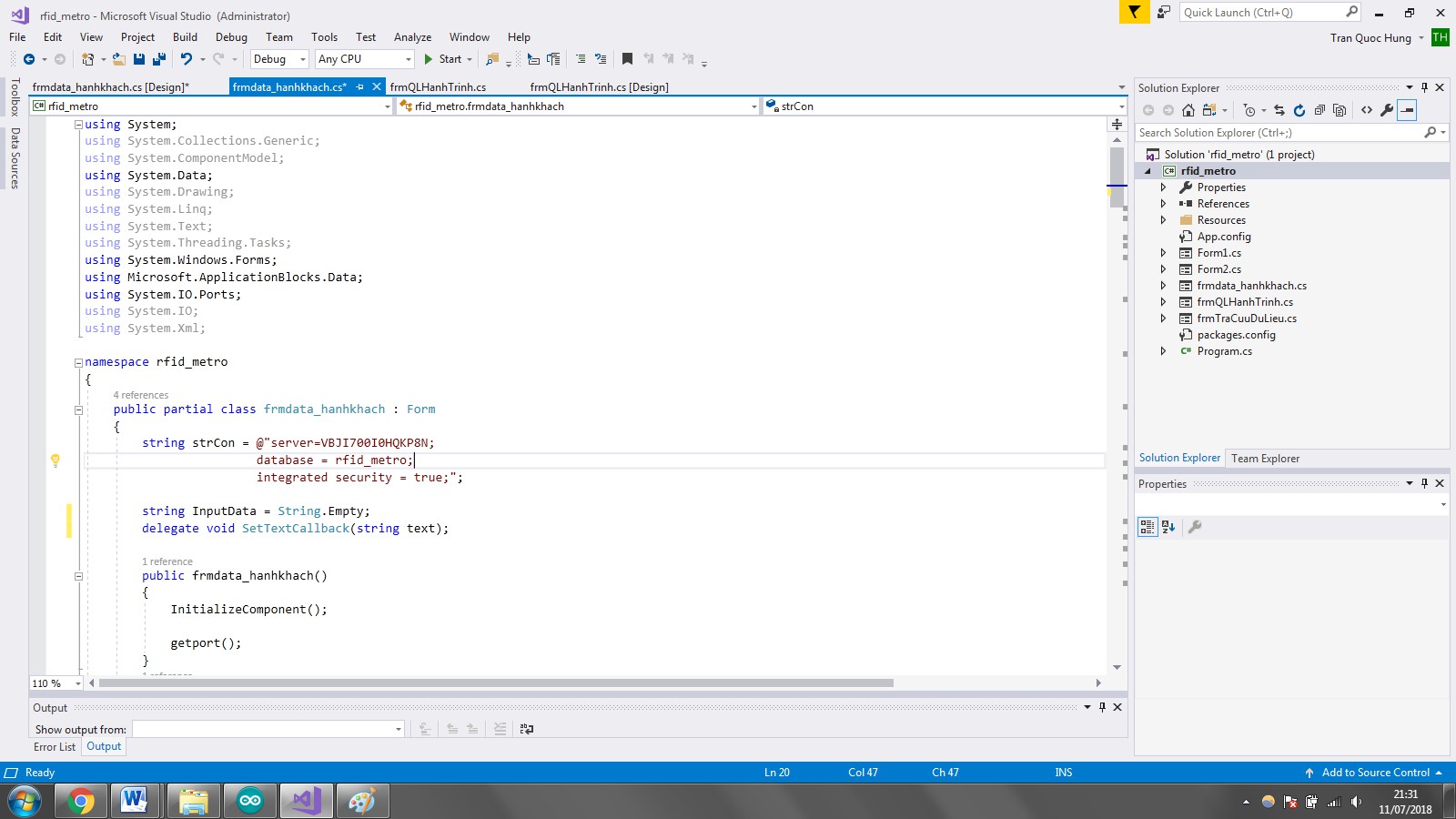
*Hình 4.8 Tạo project trong Visual Studio*

Sau đó, chương trình sẽ xuất hiện giao diện cho phép chúng ta thiết kế giao diện bằng cách sử dụng các đối tượng có sẵn cùng với các thuộc tính tương ứng.



*Hình 4.9 Giao diện thiết kế trong Visual Studio*

Sau khi chọn các sự kiện xảy ra, sẽ mở ra giao diện cho phép chúng ta lập trình như sau:



*Hình 4.10 Giao diện lập trình trong Visual Studio*

Sau khi lập trình xong, chúng ta thực hiện kiểm tra lỗi , sau đó tiến hành Debug bằng cách chọn Start Debugging và chạy chương trình.

##### Phần mềm SQL Server Management Studio

1. Giới thiệu SQL Server Management Studio

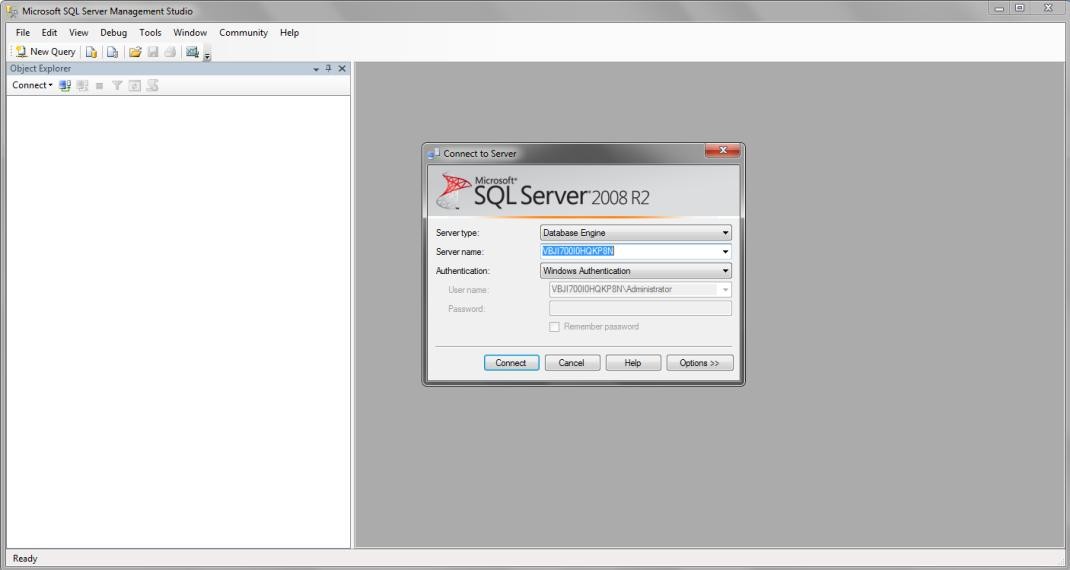
SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (Transact-SQL) để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RDBMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp “ăn ý” với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server….

1. Tạo cơ sơ dữ liệu vào thực hiện các truy vấn

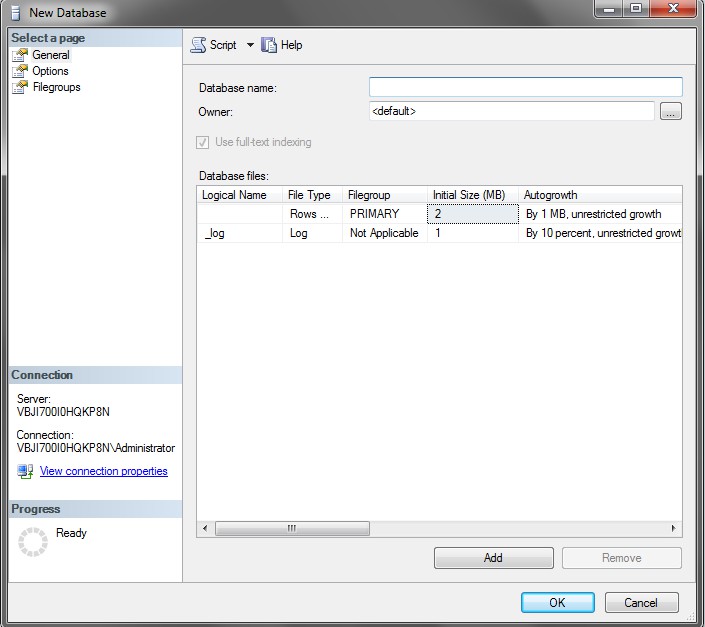
Để cài đặt, tao có thể cài đặt bản Express là phiên bản gọn nhất và hoàn toàn miễn phí.

Kết nối vào Server trong SQL Server 2008



*Hình 4.11 Kết nối Server*

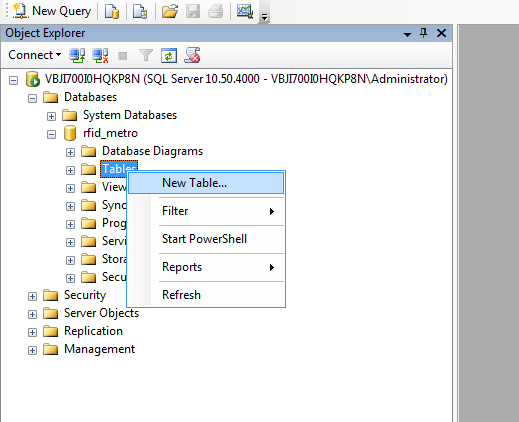
Để tạo CSDL mới, tao chuột phải vào Database/New DataBase, chương trình sẽ xuất hiện ra giao diện, đặt tên và chọn OK để tạo Database.



*Hình 4.12 Khởi tạo Database mới*

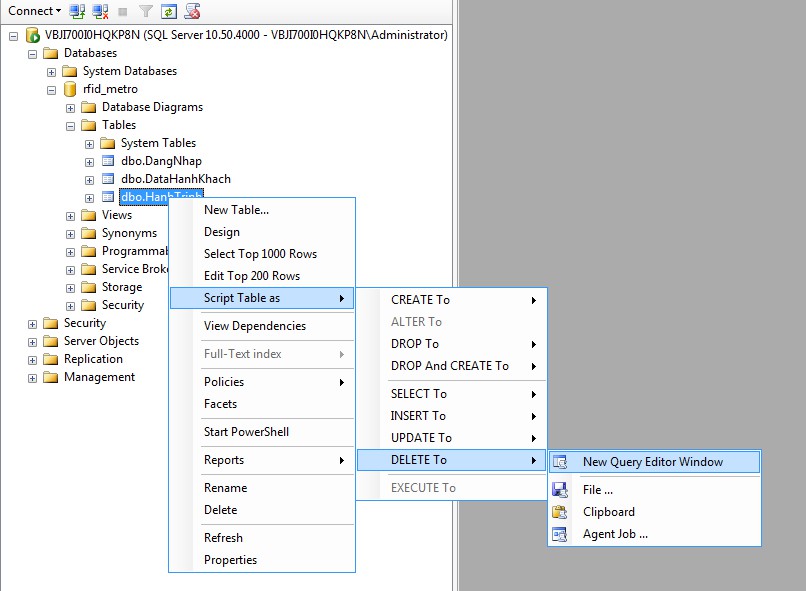
* + Tạo bảng và các truy vấn

Để tạo bảng mới, ta vào Database chọn Database cần tạo, sau đó chuột phải vào Table/New Table.



*Hình 4.13 Tạo bảng mới*

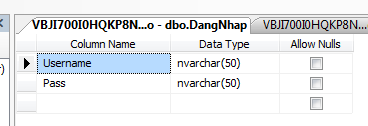
Để tạo truy vấn mới vào các bảng, ta chọn chuột phải ***Script Table As***, chọn truy vấn muốn thực hiện, chọn ***New Query Editor Windows*** và thực hiện.



*Hình 4.14 Tạo truy vấn*

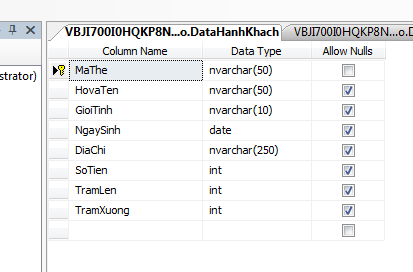
1. Các bảng và truy vấn trong chƣơng trình của hệ thống

Cơ sở dữ liệu SQL rồi 2 bảng và các truy vấn dùng để thực hiện các tác vụ lấy dữ liệu (Select), thêm (Insert), sửa (Update), xóa (Delete).

* + Các bảng trong cơ sở dữ liệu

*Hình 4.15 Bảng DangNhap*

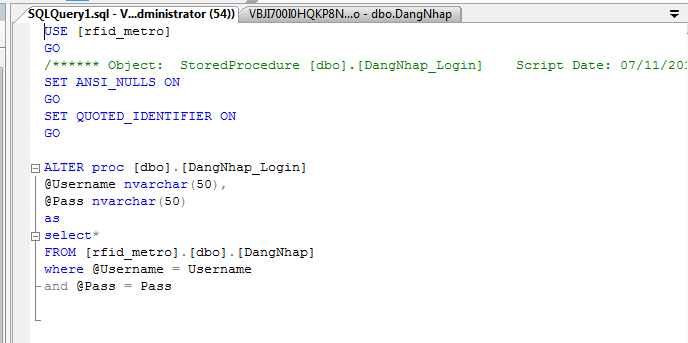
Bảng này lưu các user name và password để có thể đăng nhập vào hệ thống.



*Hình 4.16 Bảng DataHanhKhach*

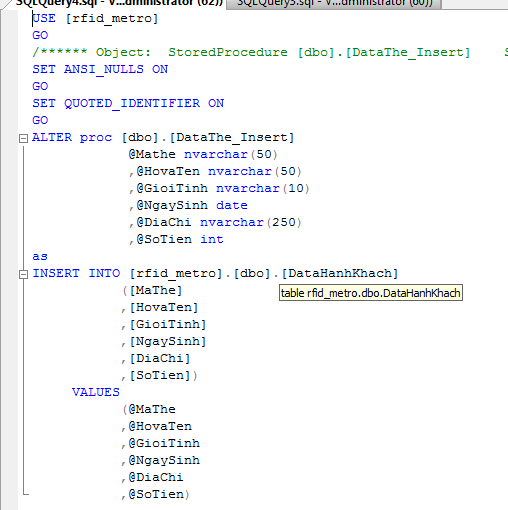
Bảng này có chức năng lưu thông tin của hành khách khi sử dụng hệ thống.

* + Các truy vấn
    1. Truy vấn Select: tuy vấn này được sử dụng để lấy dữ liệu từ bảng “DangNhap” kiểm tra xem User và Pass có đúng không để người quản lý đăng nhập vào hệ thống.



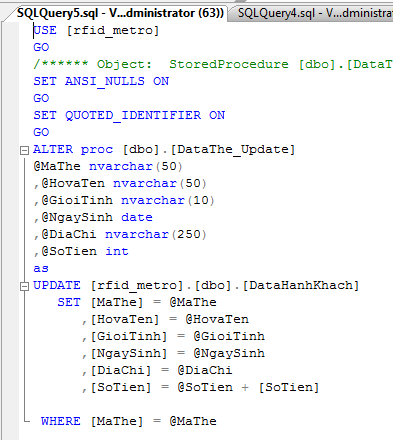
*Hình 4.17 Truy vấn Selec*t

* + 1. Truy vấn Insert: dùng để thêm thông tin hành khách vào bảng “DataHanhKhach”.



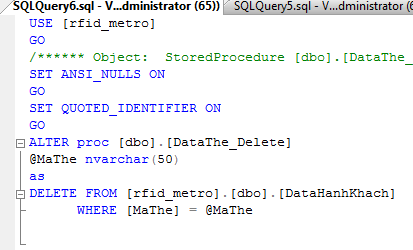
*Hình 4.18 Truy vấn Insert*

* + 1. Truy vấn Update: truy vấn này có tác dụng chỉnh sửa thông tin hành khách khi thay đổi, thường là thay đổi tiền trong tài khoản.



*Hình 4.19 Truy vấn Update*

* + 1. Truy vấn Delete: có tác dụng xóa thông tin hành khách khi người đó không còn sử dụng thẻ nữa.



*Hình 4.20 Truy vấn Delete*

#### TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC CỦA HỆ THỐNG

##### Hướng dẫn sử dụng

Để sử dụng hệ thống thì người quản lý phải thực hiện các thao tác sau đây: Bước 1: Kết nối nguồn cho Arduino, kết nối Arduino với máy tính.

Bước 2: Chạy chương trình SQL Server 2008 và chương trình quản lý, sau đó đăng nhập vào hệ thống.

Bước 3: Giao diện sao khi đăng nhập vào hệ thống bao gồm các phần sau:

* Quản lý hành khách:

Có tác dụng thêm, sửa, xóa dữ liệu hành khách. Ta quét thẻ RFID vào đầu đọc, mã thẻ sẽ tự dộng được điền vào form. Nếu dữ liệu hành khách chưa có, tiến hành điền thông tin vào các ô còn lại và chọn nút ***Thêm***. Còn nếu đã có thông tin rồi, thì các ô khác sẽ được điền đầy đủ khi ta quẹt thẻ qua đầu đọc, sau đó tiến hành ***Sửa*** hoặc ***Xóa*** theo nhu cầu.

* Quản lý hành trình:

Có tác dụng kiểm tra và cho phép/ không cho phép người dùng lên tàu, tiến hành trừ phí di chuyển.

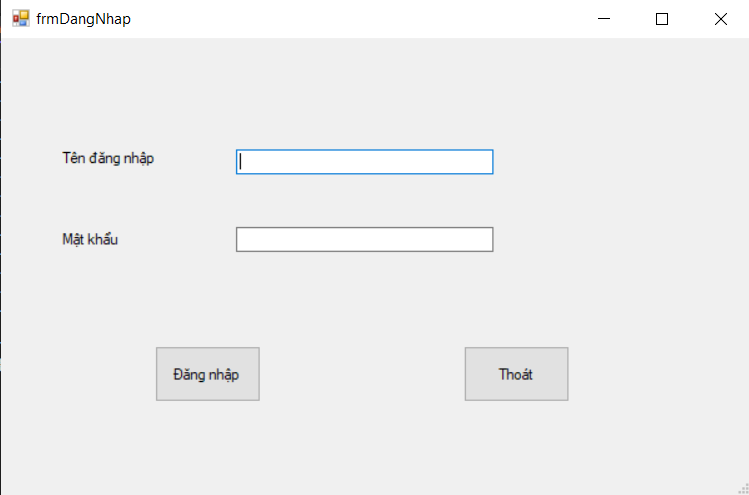
Bước 4: Để kết thúc chương trình, người dùng có thể nhấn nút ***Quay lại*** và

***Thoát.***

##### Quy trình thao tác

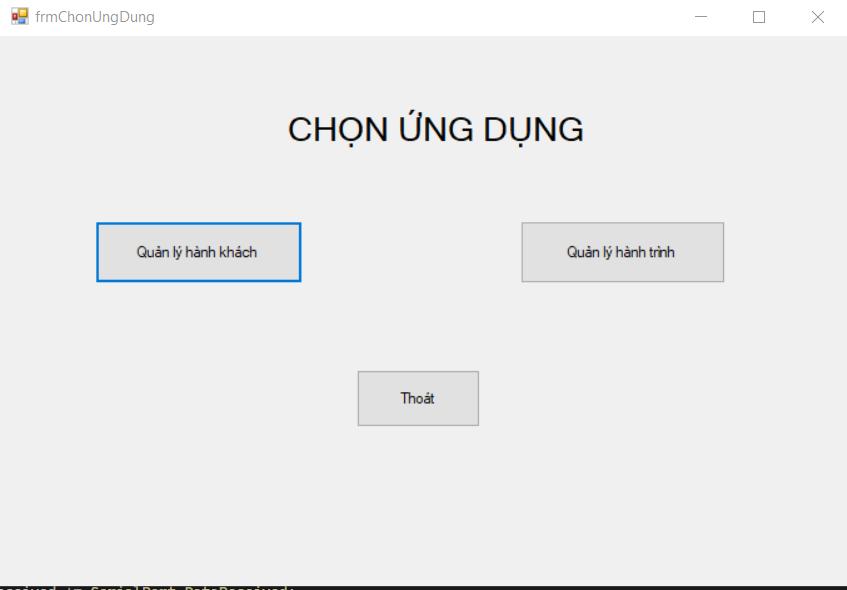
*Hình 4.21 Quy trình thao tác hệ thống*

Đầu tiên, người quản lý sẽ đăng nhập vào hệ thống.



*Hình 4.22 Giao diện đăng nhập hệ thống*

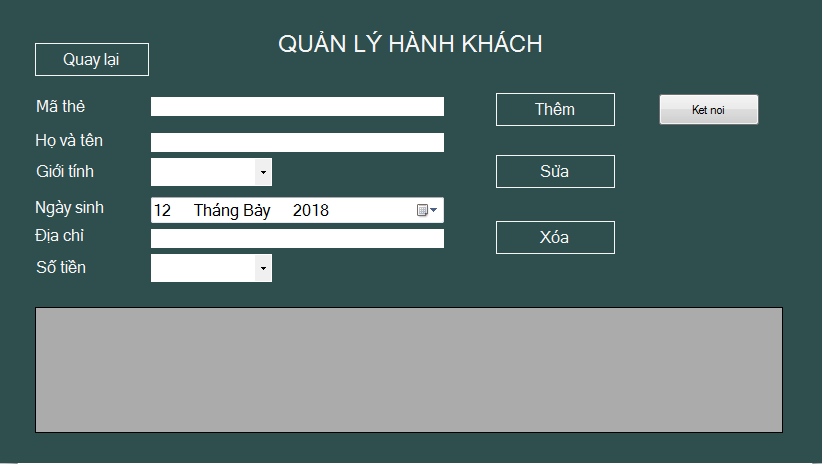
Sau đó, giao diện chọn ứng dụng sẽ hiện ra.



*Hình 4.23 Giao diện chọn ứng dụng*

Tại giao diện này, người dùng chọn 2 ứng dụng :

* Quản lý hành khách: người dùng quét thẻ qua đầu đọc, sau đó mã thẻ sẽ xuất hiện tại ô ***Mã thẻ***, sau đó tiến hành ***thêm***, ***sửa*** hoặc ***xóa*** thông tin hành khách.



*Hình 4.24 Giao diện quản lý hành khách*

* Quản lý hành trình: người dùng quét thẻ qua đầu đọc tại trạm lên và xuống tại các ga. Sau khi thực hiện một hành trình lên và xuống, thì tiền phí sẽ tự động được trừ trong tài khoản của người dùng.



*Hình 4.25 Giao diện quản lý hành trình*

## Chương 5: KẾT QUẢ VÀ NHẬN XÉT

Trong thời gian nghiên cứu đề tài từ ngày 26/03/2018 đến ngày 12/07/2018 (14 tuần) nhóm đã thực hiện được một số nội dung sau:

* Tìm hiểu về hệ thống đường sắt Metro ở những thành phố trên thế giới.
* Thiết kế được hệ thống tính cước phí khi sử dụng Metro hoạt động ổn định.
* Tìm hiểu về cách thức sử dụng của module RFID RC522, Arduino Mega 2560 nói riêng và các dòng Arduino nói chung.
* Tìm hiểu về ngôn ngữ C# và SQL, từ đó tạo được form quản lý và cơ sở dữ liệu trên máy tinh.

Qua quá trình nghiên cứu đề tài gặp một vài khó khăn dẫn đến như nhược điểm vẫn còn tồn tại:

* Phần cứng đôi lúc hoạt động không được trơn tru.
* Phần mềm vẫn còn đơn giản, vì không phải là sinh viên CNTT, những kiến thức về C# và SQL là những kiến thức tìm hiểu trong thời gian ngắn nên vẫn còn đơn giản và đôi lúc vẫn phát sinh lỗi trong quá trình hoạt động.

#### KHỐI ĐỌC DỮ LIỆU

Sử dụng module RFID RC522 để đọc mã thẻ tương đối chính xác, tuy nhiên vẫn còn chậm và chưa thể đáp ứng được nhu cầu khi có một lượng lớn người thực hiện tác vụ quét thẻ.

#### KHỐI VI ĐIỀU KHIỂN

Sử dụng Arduino Mega 2560 kết nối với các module RFID RC522 và LCD 16x2, tốc độ xử lý nhanh và ổn định, biết được cách lập trình trên Arduino Mega 2560 nói riêng và các dòng vi điều khiển Arduino nói chung.

#### CÁC KHỐI KHÁC

Mô hình sử dụng màn hình LCD 16x2 để thông báo thông tin cho người dùng, kết quả đã thực hiện thành công. Tuy nhiên màn hình LCD 16x2 vẫn còn khá nhỏ và thiết tính trực quan, không thân thiện với người dùng.

#### KẾT QUẢ THỰC HIỆN

##### Kết quả thi công phần cứng



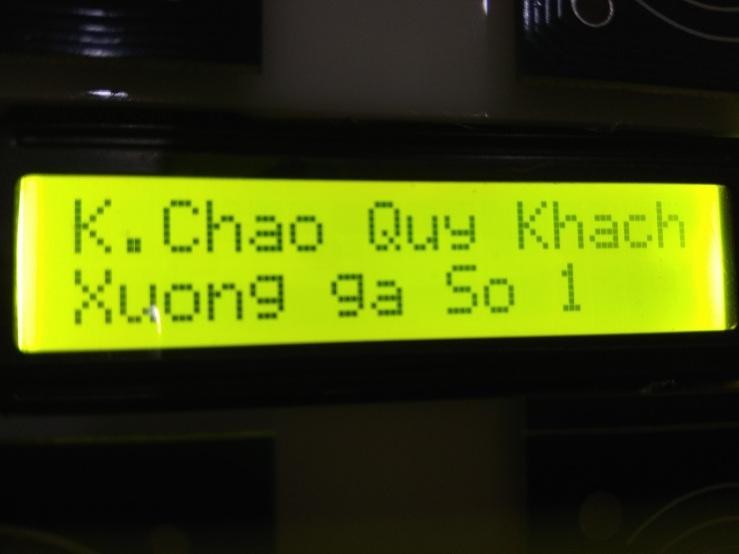
*Hình 5.1 Phần cứng của hệ thống*

##### Hình ảnh mô phỏng

1. Phần cứng

Đã kết nối được các khối và viết chương trình cho Arduino với kết quả như sau:

* + Đã đọc được mã thẻ từ đầu đọc và gửi lên máy tính
  + Hiển thị LCD khi có quét thẻ tại các ga, hiển thị được thông tin gửi ngược về từ máy tính.

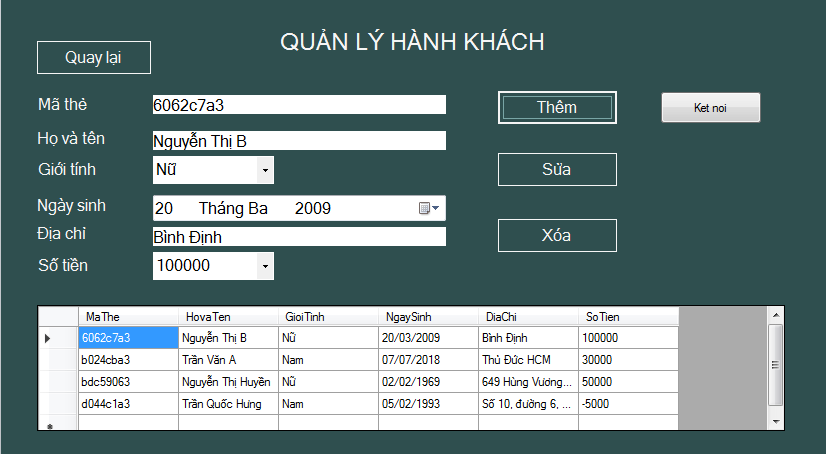


*Hình 5.2 Khi thẻ RFID được quét qua đầu đọc, tương ứng tại các ga*

1. Phần mềm quản lý

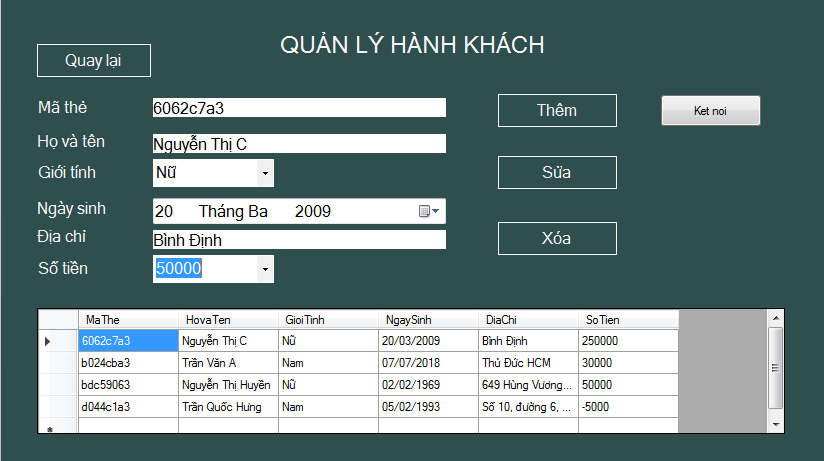
Sao khi đăng nhập vào form quản lý, người dùng chọn chức năng cần thực hiện

* + Quản lý hành khách: Khi ta quét thẻ qua đầu đọc, mã thẻ của RFID sẽ được hiện ở ô ***Mã Thẻ,*** sau đó, ta thêm các thông tin khác như họ và tên, ngày tháng năm sinh, địa chỉ, số tiền để thêm vào cơ sơ dữ liệu. Khi bấm các nút chức năng thì phần mềm sẽ hỏi nếu thực sự muốn thực hiện thao tác đó. Bấm “OK” để đồng ý, còn không thì chọn “Cancel”.

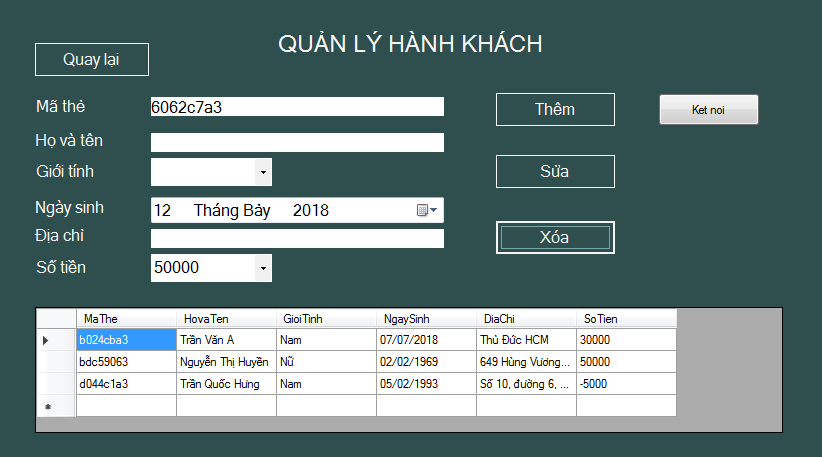


*Hình 5.3 Kết quả sau khi thêm thông tin hành khách*

Sau khi đã thêm thông tin, nếu muốn sửa đổi thông tin, ta quét thẻ qua đầu đọc lần nữa, các thông tin của hành khách sẽ được hiện ra, sau đó ta tiến hành chỉnh sửa thông tin theo ý muốn, và bấm nút ***Sửa***, hoặc bấm nút ***Xóa*** nếu muốn xóa hẳn thông tin trên cơ sở dữ liệu.

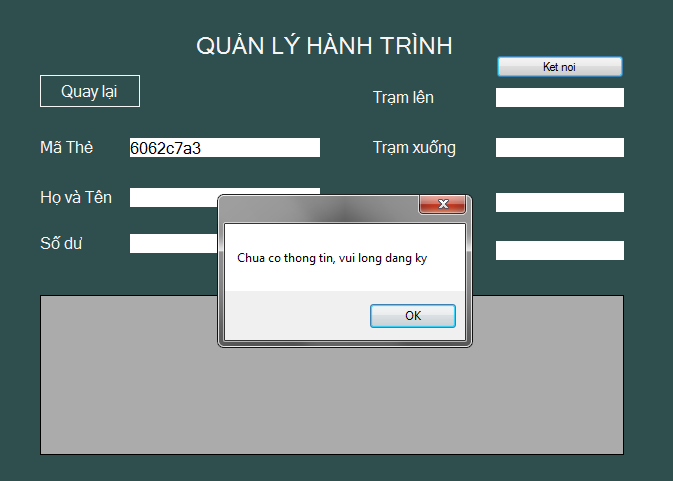


*Hình 5.4 Thông tin hành khách sau khi chỉnh sửa*



*Hình 5.5 Thông tin hành khách đã không còn sau khi xóa*

* + Quản lý hành trình: với ứng dụng này, tất cả đều thực hiện tự động. Người dùng quét thẻ tại các trạm lên xuống ở các ga. Nếu thẻ chưa có thông tin, hệ thống sẽ thông báo để người dùng đăng ký.



*Hình 5.6 Khi thẻ chưa có thông tin*

Khi đó, người dùng phải quay lại form quản lý hành khách để đăng ký thông tin hành khách mới thực hiện được hành trình.

Khi hành khách đã đủ điều kiện để thực hiện chuyến đi. Hành khách thực hiện quét thẻ tại ga lên, và quét thẻ lần 2 ở ga xuống. Hệ thống sẽ tự động cập nhật ga lên, ga xuống của hành khách trên cơ sở dữ liệu và trừ phí di chuyển vào tài khoản.



*Hình 5.7 Hành khách “Nguyễn Thị Đào” sau khi thêm thông tin và nạp “10000” vào tài khoản, tiến hành lên ga số 1*



*Hình 5.8 Hành khách xuống ga số 2, tiền phí được trừ vào tài khoản*

##### Đánh giả kết quả mô phỏng

Dựa trên kết quả đạt được khi mô phỏng, nhóm rút ra được một số kết luận sau

đây:

* + - * Mô hình phần cứng đôi lúc vẫn còn hoạt động chưa ổn định.
      * Phần mềm chạy mô phỏng đôi lúc có một số trường hợp xuất hiện trong thực tế vẫn chưa xử lý được.

## Chương 6: KẾT LUẬN VÀ PHÁT TRIỂN

#### KẾT LUẬN

Sau thời gian nghiên cứu và thi công, nhóm đã thực hiện được công việc như xây dựng phần cứng, viết giao diện quản lý, xây dựng cơ sở dữ liệu cho hệ thống. Hiểu được các lập trình Arduino, hiểu biết cơ bản về cơ chế và cách thức lập trình C# và SQL. Từ những yêu cầu đặt ra khi nghiên cứu, nhóm đã đạt được một số yêu cầu sau:

* + - Hoàn thiện được hoàn chỉnh mô hình, các bộ đọc quét được mã thẻ nhanh chóng.
    - Phần mềm hoàn thiện và hoạt động ổn định.
    - Hiểu biết cơ bản về lập trình C# và SQL.
    - Hiểu biết về Arduino Mega 2560 và module RC522. Ngoài ra còn một số yếu được cần được khắc phục:
    - Giao diện chưa thân thiện với người dùng.
    - Mô hình vẫn còn sơ xài và đôi lúc xảy ra lỗi.
    - Kỹ năng làm việc nhóm vẫn còn chưa tốt.
    - Mô hình còn nhiều điểm chưa giống với thực tế của hệ thống metro ở các nước phát triển

#### HƯỚNG PHÁT TRIỂN

Sau thời gian nghiên cứu và xây dựng mô hình mô phỏng, để có thể áp dụng vào thực tiễn tốt hơn thì cần phải phát triển thêm các yếu tố sau:

* + - Thiết kế cơ sở dữ liệu không chỉ lưu trữ trên một máy tính mà phải lưu trữ trong server với nhiều máy tính kết nối, hoặc thông qua internet, đáp ứng nhu cầu mở rộng mô hình.
    - Thiết kế giao diện và mô hình trực quan hơn, để người dùng có thể trực tiếp tự nạp thêm tiền vào thẻ tại các quầy, các cửa hàng tiện lợi.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Văn Hiệp, “Giáo trình công nghệ nhận dạng bằng sống vô tuyến”, ĐH Sư phạm Kỹ Thuật TP.HCM, 2014.
2. Trần Nguyên Phong, “Giáo trình thực hành SQL”, ĐH Khoa học tự nhiên - ĐH Quốc Gia TP.HCM, 2012.
3. Phạm Trọng Đạt, “Đồ án ứng dụng công nghệ RFID trong hệ thống thu phí vé xe buýt tự động”, ĐH Giao thông vận tải Hà Nội, 2016.
4. Stephen B.Miles, Sanjay E.Sama, John R.Willians, “RFID Technology and Application”, Cambridge University.
5. Lê Ngọc Minh, “Đồ án Hệ thống tính cước phí đường sắt Metro sử dụng RFID”, ĐH Sư phạm Kỹ thuật TP.HCM, 2016.

# PHỤ LỤC

* + - Chương trình chính Arduino #include <LiquidCrystal\_I2C.h> #include <Wire.h>

#include <String.h> #include <MFRC522.h> #include <SPI.h>

LiquidCrystal\_I2C lcd(0x27,16,2); #define SS\_PIN1 30

#define SS\_PIN2 33

#define SS\_PIN3 31

#define SS\_PIN4 32

#define RST\_PIN 5

MFRC522 tram1(SS\_PIN1,RST\_PIN); MFRC522 tram2(SS\_PIN2,RST\_PIN); MFRC522 tram3(SS\_PIN3,RST\_PIN); MFRC522 tram4(SS\_PIN4,RST\_PIN);

//Khoi tao RC522, LCD

void setup() { Serial.begin(9600); SPI.begin(); tram1.PCD\_Init(); tram2.PCD\_Init(); tram3.PCD\_Init();

tram4.PCD\_Init(); lcd.init(); lcd.backlight();

}

void loop() { lenga1();

lenga2(); xuongga1(); xuongga2(); hienthi\_lcd();

}

* + - Chương trình con của Arduino void lenga1()

{

if(!tram1.PICC\_IsNewCardPresent())

{

return;

}

if(!tram1.PICC\_ReadCardSerial())

{

return;

}

String matag=""; String b = "-1-1"; String c = "";

for(byte i=0;i<tram1.uid.size;i++)

matag.concat(String(tram1.uid.uidByte[i],HEX)); c = matag + b;

Serial.println(c); lcd.clear();

lcd.print("K.Chao Quy Khach"); lcd.setCursor(0,1);

lcd.print(" Len ga So 1 "); delay(2000);

lcd.clear();

}

void lenga2()

{

if(!tram3.PICC\_IsNewCardPresent())

{

return;

}

if(!tram3.PICC\_ReadCardSerial())

{

return;

}

String matag2=""; String g = "-2-1"; String h = "";

for(byte i=0;i<tram3.uid.size;i++)

matag2.concat(String(tram3.uid.uidByte[i],HEX)); h = matag2 + g;

Serial.println(h); lcd.clear();

lcd.print("K.Chao Quy Khach"); lcd.setCursor(0,1);

lcd.print(" Len ga So 2 "); delay(2000);

lcd.clear();

}

void xuongga1()

{

if(!tram2.PICC\_IsNewCardPresent())

{

return;

}

if(!tram2.PICC\_ReadCardSerial())

{

return;

}

String matag1=""; String d = "-1-2"; String e = "";

for (byte i=0;i<tram2.uid.size;i++) matag1.concat(String(tram2.uid.uidByte[i],HEX)); e = matag1 + d;

Serial.println(e);

delay(300);

lcd.clear();

lcd.print("K.Chao Quy Khach"); lcd.setCursor(0,1); lcd.print("Xuong ga So 1"); delay(2000);

lcd.clear();

}

void xuongga2()

{

if(!tram4.PICC\_IsNewCardPresent())

{

return;

}

if(!tram4.PICC\_ReadCardSerial())

{

return;

}

String matag3=""; String j = "-2-2"; String k = "";

for(byte i=0;i<tram4.uid.size;i++) matag3.concat(String(tram4.uid.uidByte[i],HEX)); k = matag3 + j;

Serial.println(k); lcd.clear();

lcd.print("K.Chao Quy Khach");

lcd.setCursor(0,1); lcd.print("Xuong ga So 2"); delay(2000);

lcd.clear();

}

void hienthi\_lcd()

{

if(Serial.available())

{

delay(50); lcd.clear();

lcd.print("K.Chao quy khach"); lcd.setCursor(0,1);

while (Serial.available() > 0)

{

lcd.write(Serial.read());

}

}

}

* Đoạn code giao diện C# a.Form Đăng nhập

using System;

using System.Collections.Generic; using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Microsoft.ApplicationBlocks.Data;

namespace rfid\_metro

{

public partial class Form1 : Form

{

string strCon = @"server=VBJI700I0HQKP8N; database = rfid\_metro;

integrated security = true;"; public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string user = txtUsername.Text.Trim(); string pass = txtPass.Text.Trim();

DataTable dt = SqlHelper.ExecuteDataset(strCon, "DangNhap\_Login", user, pass).Tables[0];

if (dt.Rows.Count > 0)

{

MessageBox.Show("Dang nhap thanh cong !"); Form2 frm = new Form2();

frm.Show();

Hide();

}

else MessageBox.Show("That bai !!");

}

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void txtPass\_TextChanged(object sender, EventArgs e)

{

txtPass.PasswordChar = '\*';

}

b. Form Quản lý hành khách

using System.Data; using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

using Microsoft.ApplicationBlocks.Data; using System.IO.Ports;

using System.IO; using System.Xml;

namespace rfid\_metro

{

public partial class frmdata\_hanhkhach : Form

{

string strCon = @"server=VBJI700I0HQKP8N; database = rfid\_metro;

integrated security = true;";

string InputData = String.Empty;

delegate void SetTextCallback(string text); public frmdata\_hanhkhach()

{

InitializeComponent();

getport();

}

void getport()

{

String[] ports = SerialPort.GetPortNames();

}

public void setValue(int index)

{

DataGridViewRow row = dgDataThe.Rows[index]; txtMaThe.Text = row.Cells[1].Value.ToString(); txtHovaTen.Text = row.Cells[2].Value.ToString(); cbbGioiTinh.Text = row.Cells[3].Value.ToString(); dtpNgaySinh.Text = row.Cells[4].Value.ToString(); txtDiaChi.Text = row.Cells[5].Value.ToString(); cbbSoDuTK.Text = row.Cells[6].Value.ToString();

}

public void loadData()

{

dgDataThe.DataSource = SqlHelper.ExecuteDataset(strCon, "DataThe\_SelectAll").Tables[0];

}

public void frmdata\_hanhkhach\_Load(object sender, EventArgs e)

{

serialPort1.DataReceived += new SerialDataReceivedEventHandler(serialPort1\_DataReceive);

loadData();

}

private void serialPort1\_DataReceive(object obj, SerialDataReceivedEventArgs

e)

{

InputData = serialPort1.ReadLine(); if (InputData != String.Empty)

{

this.Invoke(new EventHandler(DisplayText));

}

}

private void DisplayText(object sender, EventArgs e)

{

clear();

string DispString = serialPort1.ReadLine().Replace("/r", ""); string[] MaThe = DispString.Split('-');

txtMaThe.Text = MaThe[0]; Find(txtMaThe.Text);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

string mathe = txtMaThe.Text.Trim(); string hovaten = txtHovaTen.Text.Trim(); string gioitinh = cbbGioiTinh.Text;

DateTime ns = DateTime.Parse(dtpNgaySinh.Value.ToString()); string dc = txtDiaChi.Text.Trim();

string sotien = cbbSoDuTK.Text;

SqlHelper.ExecuteNonQuery(strCon, "DataThe\_Insert", mathe, hovaten, gioitinh, ns, dc, sotien);

DialogResult DR = MessageBox.Show("Ban co muon them hanh khach nay khong ?", "Thong bao", MessageBoxButtons.OKCancel);

if (DialogResult.OK==DR)

{

loadData(); clear();

}

if (DialogResult.Cancel==DR)

{

frmdata\_hanhkhach\_Load(null, null);

}

}

catch(Exception)

{

}

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

string mathe = txtMaThe.Text.Trim(); string hovaten = txtHovaTen.Text.Trim(); string gioitinh = cbbGioiTinh.Text;

DateTime ns = DateTime.Parse(dtpNgaySinh.Value.ToString()); string dc = txtDiaChi.Text.Trim();

string sotien = cbbSoDuTK.Text;

SqlHelper.ExecuteNonQuery(strCon, "DataThe\_Update", mathe, hovaten, gioitinh, ns, dc, sotien);

DialogResult DR = MessageBox.Show("Ban co muon chinh sua thong tin hanh khach nay khong ?", "Thong bao", MessageBoxButtons.OKCancel);

if (DialogResult.OK == DR)

{

loadData(); clear();

}

if (DialogResult.Cancel == DR)

{

;

}

}

catch(Exception)

{

}

}

private void button3\_Click(object sender, EventArgs e)

{

try

{

string mathe = txtMaThe.Text.Trim(); SqlHelper.ExecuteNonQuery(strCon, "DataThe\_Delete", mathe);

DialogResult DR = MessageBox.Show("Ban co muon xoa hanh khach nay khong?", "Thong Bao", MessageBoxButtons.OKCancel);

if (DialogResult.OK == DR)

{

loadData(); clear();

}

if (DialogResult.Cancel == DR)

{

;

}

}

catch (Exception)

{

}

}

private void frmdata\_hanhkhach\_FormClosing(object sender, FormClosingEventArgs e)

{

Form2 frm = new Form2(); frm.Show();

Hide(); serialPort1.Close();

}

private void dgDataThe\_RowPrePaint(object sender, DataGridViewRowPrePaintEventArgs e)

{

for (int i = 0; i < dgDataThe.RowCount; i++)

{

dgDataThe[0, i].Value = i + 1;

}

}

private void dgDataThe\_CellClick(object sender, DataGridViewCellEventArgs e)

{

setValue(e.RowIndex);

}

private bool Find(string maThe)

{

var hanhkhachs = SqlHelper.ExecuteDataset(strCon, "DataThe\_SelectAll").Tables[0].Rows;

foreach(DataRow hanhkhach in hanhkhachs)

{

var a = hanhkhach[0];

string b = Convert.ToString(a); if (b == maThe)

{

txtMaThe.Text = Convert.ToString(hanhkhach[0]);

txtHovaTen.Text = Convert.ToString(hanhkhach[1]); cbbGioiTinh.Text = Convert.ToString(hanhkhach[2]); dtpNgaySinh.Text = Convert.ToString(hanhkhach[3]); txtDiaChi.Text = Convert.ToString(hanhkhach[4]);

}

}

return true;

}

private void clear()

{

txtMaThe.Text = ""; txtHovaTen.Text = ""; cbbGioiTinh.Text = ""; dtpNgaySinh.Text = ""; txtDiaChi.Text = "";

}

private void txtMaThe\_TextChanged(object sender, EventArgs e)

{

if (txtMaThe.Text.Length > 0) Find(txtMaThe.Text);

}

private void button4\_Click(object sender, EventArgs e)

{

serialPort1.Close();

Form2 form2 = new rfid\_metro.Form2(); form2.Show();

this.Hide();

}

private void button6\_Click(object sender, EventArgs e)

{

if (!serialPort1.IsOpen)

{

serialPort1.PortName = "COM8"; serialPort1.BaudRate = 9600; serialPort1.Open();

}

}

private void button5\_Click(object sender, EventArgs e)

{

serialPort1.Close();

}

}

}

b. Quản lý hành trình using System;

using System.Collections.Generic;

using System.ComponentModel; using System.Data;

using System.Drawing; using System.Linq; using System.Text;

using System.Threading.Tasks; using System.Windows.Forms;

using Microsoft.ApplicationBlocks.Data; using System.IO.Ports;

using System.IO; using System.Xml;

namespace rfid\_metro

{

public partial class frmQLHanhTrinh : Form

{

string strCon = @"server=VBJI700I0HQKP8N; database = rfid\_metro;

integrated security = true;";

string InputData = String.Empty;

delegate void SetTextCallback(string text); public frmQLHanhTrinh()

{

InitializeComponent(); getport();

}

public void loadData()

{

dtgQLHanhTrinh.DataSource = SqlHelper.ExecuteDataset(strCon, "Select\_DLHanhTrinh").Tables[0];

}

void getport()

{

String[] ports = SerialPort.GetPortNames();

}

public void frmQLHanhTrinh\_Load(object sender, EventArgs e)

{

serialPort1.DataReceived += new SerialDataReceivedEventHandler(serialPort1\_DataReceive);

loadData();

}

private void serialPort1\_DataReceive(object obj, SerialDataReceivedEventArgs

e)

{

InputData = serialPort1.ReadLine(); if (InputData != String.Empty)

{

this.Invoke(new EventHandler(DisplayText));

}

}

private void SetText(string text)

{

if (this.txtMaThe.InvokeRequired)

{

SetTextCallback d = new SetTextCallback(SetText); this.Invoke(d, new object[] { text });

}

else this.txtMaThe.Text += text;

}

private void DisplayText(object sender,EventArgs e)

{

string DispString = serialPort1.ReadLine().Replace("\r", ""); string[] MaThe = DispString.Split('-');

txtMaThe.Text = MaThe[0];

if (Convert.ToInt32(MaThe[2]) == 1) txtTramLen.Text = MaThe[1];

else

txtTramXuong.Text = MaThe[1]; Find(txtMaThe.Text);

}

private bool Find(string maThe)

{

var hanhTrinhs = SqlHelper.ExecuteDataset(strCon, "Select\_DLHanhTrinh").Tables[0].Rows;

foreach (DataRow hanhTrinh in hanhTrinhs)

{

var a = hanhTrinh[0];

string b = Convert.ToString(a); if (b == maThe)

{

txtMaThe.Text = Convert.ToString(hanhTrinh[0]);

txtHovaTen.Text = Convert.ToString(hanhTrinh[1]); txtSoDuTK.Text = Convert.ToString(hanhTrinh[2]); txtTramLen.Text = Convert.ToString(hanhTrinh[3]);

}

}

return true;

}

private void frmQLHanhTrinh\_FormClosing(object sender, FormClosingEventArgs e)

{

Form2 frm = new rfid\_metro.Form2(); frm.Show();

Hide(); serialPort1.Close();

}

private void txtMaThe\_TextChanged(object sender, EventArgs e)

{

if(txtMaThe.Text.Length > 0)

{

string a = txtMaThe.Text.Trim();

DataTable dt = SqlHelper.ExecuteDataset(strCon, "timmathe", a).Tables[0]; if(dt.Rows.Count > 0)

{

timer1.Interval = 500; timer1.Enabled = true;

Find(txtMaThe.Text);

int b = Convert.ToInt32(txtSoDuTK.Text); if (b <= 0)

{

serialPort1.Write("So du TK khong du !!"); MessageBox.Show("khong du tien");

}

else if (b > 0)

{

serialPort1.Write("So du TK: " + txtSoDuTK.Text);

}

}

else

{

MessageBox.Show("Chua co thong tin, vui long dang ky"); serialPort1.Write("Vui long DK the");

clear();

}

}

}

private void clear()

{

loadData(); txtMaThe.Text = ""; txtHovaTen.Text = ""; txtSoDuTK.Text = ""; txtTramLen.Text = ""; txtTramXuong.Text = "";

txtSoTienConLai.Text = ""; txtTienPhi.Text = "";

}

private void txtTramXuong\_TextChanged(object sender, EventArgs e)

{

if((txtTramXuong.Text.Length > 0) && (txtMaThe.Text.Length > 0))

{

int a = Convert.ToInt32(txtTramLen.Text); int b = Convert.ToInt32(txtTramXuong.Text); long c;

c = (Math.Abs(b - a)) \* 2000;

long d = Convert.ToInt32(txtSoDuTK.Text); long f = d - c;

txtTienPhi.Text = Convert.ToString(c); txtSoTienConLai.Text = Convert.ToString(f); try

{

string mathe = txtMaThe.Text.Trim();

string sotien = txtSoTienConLai.Text.Trim(); string tramlen = txtTramLen.Text.Trim(); string tramxuong = txtTramXuong.Text.Trim();

SqlHelper.ExecuteNonQuery(strCon, "Tienphi\_Update", mathe, sotien, tramlen, tramxuong);

loadData();

serialPort1.Write("TK con: " + Convert.ToString(f));

}

catch(Exception)

{

}

}

}

private void txtTramlen\_TextChanged(object sender, EventArgs e)

{

if(txtTramLen.Text.Length > 0)

{

try

{

string mathe = txtMaThe.Text.Trim(); string tramlen = txtTramLen.Text.Trim();

tramlen);

}

SqlHelper.ExecuteNonQuery(strCon, "DLHanhTrinh\_Update", mathe, loadData();

catch(Exception)

{

}

}

}

private void txtSoTienConLai\_TextChanged(object sender, EventArgs e)

{

if(txtSoTienConLai.Text.Length > 0)

{

try

{

string Ngaydi = Convert.ToString(DateTime.Now); string mathe = txtMaThe.Text.Trim();

string hovaten = txtHovaTen.Text.Trim(); string sotien = txtSoDuTK.Text.Trim(); string tramlen = txtTramLen.Text.Trim();

string tramxuong = txtTramXuong.Text.Trim(); string tienve = txtTienPhi.Text.Trim();

string sotienconlai = txtSoTienConLai.Text.Trim(); SqlHelper.ExecuteNonQuery(strCon, "LuuDuLieuDiChuyen", Ngaydi,

mathe, hovaten, sotien, tramlen, tramxuong, tienve, sotienconlai);

}

catch(Exception)

{

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

serialPort1.Close();

Form2 form2 = new rfid\_metro.Form2(); form2.Show();

this.Hide();

}

private void txtSoDuTK\_TextChanged(object sender, EventArgs e)

{

}

private void frmQLHanhTrinh\_Load\_1(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

if (!serialPort1.IsOpen)

{

serialPort1.PortName = "COM8"; serialPort1.BaudRate = 9600; serialPort1.Open();

}

}

private void button3\_Click(object sender, EventArgs e)

{

serialPort1.Close();

}

}

}

* Code SQL Server
  1. Truy vấn DangNhap\_login

USE [rfid\_metro] GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DangNhap\_Login] Script Date: 07/12/2018 23:53:38 \*\*\*\*\*\*/

SET ANSI\_NULLS ON GO

SET QUOTED\_IDENTIFIER ON GO

ALTER proc [dbo].[DangNhap\_Login] @Username nvarchar(50),

@Pass nvarchar(50) as

select\*

FROM [rfid\_metro].[dbo].[DangNhap] where @Username = Username

and @Pass = Pass

* 1. Truy vấn DataThe\_SelectAll

USE [rfid\_metro] GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DataThe\_SelectAll] Script Date: 07/12/2018 23:54:14 \*\*\*\*\*\*/

SET ANSI\_NULLS ON GO

SET QUOTED\_IDENTIFIER ON GO

ALTER proc [dbo].[DataThe\_SelectAll] as

SELECT [MaThe]

,[HovaTen]

,[GioiTinh]

,[NgaySinh]

,[DiaChi]

,[SoTien]

FROM [rfid\_metro].[dbo].[DataHanhKhach]

* 1. Truy vấn DataThe\_Insert USE [rfid\_metro]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DataThe\_Insert] Script Date: 07/12/2018 23:55:03 \*\*\*\*\*\*/

SET ANSI\_NULLS ON GO

SET QUOTED\_IDENTIFIER ON GO

ALTER proc [dbo].[DataThe\_Insert]

@Mathe nvarchar(50)

,@HovaTen nvarchar(50)

,@GioiTinh nvarchar(10)

,@NgaySinh date

,@DiaChi nvarchar(250)

,@SoTien int

as

INSERT INTO [rfid\_metro].[dbo].[DataHanhKhach] ([MaThe]

,[HovaTen]

,[GioiTinh]

,[NgaySinh]

,[DiaChi]

,[SoTien]) VALUES

(@MaThe

,@HovaTen

,@GioiTinh

,@NgaySinh

,@DiaChi

,@SoTien)

* 1. Truy vấn DataThe\_Update

USE [rfid\_metro] GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DataThe\_Update] Script Date: 07/12/2018 23:55:30 \*\*\*\*\*\*/

SET ANSI\_NULLS ON GO

SET QUOTED\_IDENTIFIER ON GO

ALTER proc [dbo].[DataThe\_Update] @MaThe nvarchar(50)

,@HovaTen nvarchar(50)

,@GioiTinh nvarchar(10)

,@NgaySinh date

,@DiaChi nvarchar(250)

,@SoTien int as

UPDATE [rfid\_metro].[dbo].[DataHanhKhach] SET [MaThe] = @MaThe

,[HovaTen] = @HovaTen

,[GioiTinh] = @GioiTinh

,[NgaySinh] = @NgaySinh

,[DiaChi] = @DiaChi

,[SoTien] = @SoTien + [SoTien] WHERE [MaThe] = @MaThe

* 1. Truy vấn DataThe\_Delete

USE [rfid\_metro] GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[DataThe\_Delete] Script Date: 07/12/2018 23:56:18 \*\*\*\*\*\*/

SET ANSI\_NULLS ON GO

SET QUOTED\_IDENTIFIER ON GO

ALTER proc [dbo].[DataThe\_Delete] @MaThe nvarchar(50)

as

DELETE FROM [rfid\_metro].[dbo].[DataHanhKhach] WHERE [MaThe] = @MaThe