

Mục lục

Mục lục	i
Lời mở đầu.....	i
Lời cảm ơn.....	ii
Danh sách hình	iii
Danh sách bảng.....	v
Danh sách các từ viết tắt.....	vi
Chương 1. Tổng quan.....	1
1.1. Đặt vấn đề	1
1.2. Tình hình nghiên cứu hiện nay	1
1.2.1. Tình hình nghiên cứu trong nước	1
1.2.2. Tình hình nghiên cứu ngoài nước	2
1.3. Tính cấp thiết của đề tài	2
1.4. Mục đích của đề tài	3
1.5. Phương pháp nghiên cứu	3
1.6. Bố cục đồ án.....	4
Chương 2. Cơ sở lý thuyết.....	5
2.1. Tổng quan về các chuẩn truyền	5
2.1.1. Tổng quan về UART	5
2.1.2. Tổng quan về chuẩn giao tiếp SPI.....	7
2.1.3. Tổng quan về I2C.....	9
2.2. Tổng quan về công nghệ RFID	10
2.3. Tổng quan về ngôn ngữ lập trình Web	11
2.3.1. HTML	11
2.3.2. CSS	15
2.3.3. PHP	20
2.4. Tổng quan về thư viện xử lý hình ảnh	28

2.5. Tổng quan về cơ sở dữ liệu (Database)	29
2.5.1. SQL Server	29
2.5.2. MySQL	30
2.6. Tổng quan về thuật toán xử lý nhận diện ảnh	31
2.6.1. Thuật toán OCR	31
2.6.2. Thuật toán Canny	32
2.6.3. Quy trình xử lý nhận diện ảnh	33
Chương 3. Thiết kế và xây dựng hệ thống	36
3.1. Giới thiệu và tóm tắt	36
3.1.1. Yêu cầu chung của hệ thống	36
3.1.2. Phương án thiết kế	36
3.2. Sơ đồ khối	37
3.3. Thiết kế phần cứng	38
3.3.1. Khối động cơ Servo	38
3.3.2. Khối nút nhấn	39
3.3.3. Khối cảm biến vật cản	40
3.3.4. Khối hiển thị	41
3.3.5. Khối RFID	43
3.3.6. Khối xử lý trung tâm	44
3.3.7. Khối nguồn	45
3.3.8. Sơ đồ nguyên lý toàn mạch	46
3.4. Thiết kế phần mềm	46
3.4.1. Thiết kế phần mềm cho PC	46
3.4.1.1. Lưu đồ giải thuật chương trình chính	47
3.4.1.2. Lưu đồ giải thuật của chương trình con	48
3.4.2. Thiết kế trang Web đặt chỗ	50
3.4.3. Chương trình cho RFID-Reader	50
Chương 4. Kết quả thực hiện	52
4.1. Phần cứng	52

4.1.1.	Các công cụ sử dụng	52
4.1.2.	Vẽ mạch in mạch đã thiết kế.....	52
4.1.3.	Board mạch đã hoàn thiện	53
4.1.4.	Thi công mô hình bãi xe	53
4.2.	Phần mềm	54
4.2.1.	Phần mềm quản lý bãi giữ xe trên PC	54
4.2.2.	Web đặt chỗ.....	56
Chương 5: Kết quả so sánh, thực nghiệm, phân tích, tổng hợp		59
Chương 6. Kết luận và hướng phát triển		64
6.1.	Kết luận	64
6.2.	Hướng phát triển.....	64
6.3.	Ứng dụng trong tương lai gần.....	65
Tài liệu tham khảo		66
Phụ Lục.....		67

Lời mở đầu

Theo sau sự phát triển và biến động như vũ bão của nền kinh tế thế giới, cùng đó là tốc độ phát triển chóng mặt của Khoa học – Kỹ thuật, ngày nay ở các trung tâm thành phố lớn sự phát triển mật độ dân cư và xe cộ ngày càng đông đúc. Đặc biệt là sự gia tăng về số lượng xe ô tô ngày càng nhiều và điều này phần nào cũng phản ánh sự phát triển của một quốc gia. Từ đó dẫn đến vấn đề môi trường, ùn tắc giao thông và thiếu bãi đậu, đỗ xe cần được cấp thiết giải quyết.

Điện tử đang trở thành một ngành công nghiệp đa nhiệm. Điện tử đã đang đáp ứng những đòi hỏi không ngừng từ các lĩnh vực công, nông, lâm, ngư nghiệp cho đến các nhu cầu thiết bị trong đời sống hàng ngày và nhiều hơn thế nữa. Các thiết bị điện tử đã, đang và sẽ tiếp tục được ứng dụng ngày càng rộng rãi và mang lại hiệu quả trong hầu hết các lĩnh vực khoa học kỹ thuật cũng như trong đời sống xã hội đi đôi với sự phát triển của khoa học và công nghệ.

Với trình độ khoa học kỹ thuật ngày càng phát triển. Nhiều vấn đề đã được giải quyết nhanh gọn với công nghệ điện tử và tự động hóa. Các bãi giữ xe truyền thống đã không còn phù hợp với hầu hết các tòa nhà, chung cư hay bệnh viện... vì những rắc rối mà nó mang lại. Với hệ thống máy giữ xe bằng thẻ từ (máy giữ xe quét thẻ) các vấn đề nan giải muôn thủa như vấn đề ùn tắc, tình trạng mất xe, mất vé gửi... dường như đã được giải quyết triệt để. Xuất phát từ các vấn đề thiết thực đó đề tài **“Bãi giữ xe ứng dụng công nghệ RFID kết hợp nhận diện biển số”** đã được chọn cho quá trình nghiên cứu.

Lời cảm ơn

Trong thời gian thực hiện đề tài này, nhóm đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Nhóm thực hiện đề tài xin gửi lời cảm ơn chân thành đến thầy Ths.Nguyễn Ngô Lâm giảng viên Bộ môn Điện tử công nghiệp - trường Đại học Sư Phạm Kỹ Thuật Tp.HCM người đã tận tình hướng dẫn, chỉ bảo nhóm trong suốt quá trình làm đề tài.

Nhóm thực hiện đề tài cũng xin chân thành cảm ơn các thầy cô giáo trong trường Đại học Sư Phạm Kỹ Thuật Tp.HCM nói chung, các thầy cô trong Khoa Đào Tạo Chất Lượng Cao và Bộ môn Điện tử công nghiệp nói riêng đã cho nhóm em kiến thức về các môn đại cương cũng như các môn học chuyên ngành, giúp nhóm em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ nhóm em trong suốt quá trình học tập.

Tp. Hồ Chí Minh, Tháng 7 năm 2018

Nhóm sinh viên thực hiện

Vũ Tiến Trình - Lê Vũ Khanh

Danh sách hình

Hình 2.1. Tín hiệu tương đương của UART và RS232.....	5
Hình 2.2. Giao diện SPI.....	8
Hình 2.3. Truyền dữ liệu SPI.....	8
Hình 2.4. Giao tiếp I2C.	9
Hình 2.5. Hệ thống RFID cơ bản.....	11
Hình 2.6. Phương thức hoạt động của ngôn ngữ PHP.....	20
Hình 2.7. Ví dụ về tổ chức dạng cây của đường biên.	35
Hình 2.8. Mô hình lưu trữ Counters dạng cây.....	35
Hình 3.1. Sơ đồ khối của hệ thống.	37
Hình 3.2. Sơ đồ kết nối Servo.	38
Hình 3.3. Sơ đồ kết nối nút nhấn.	39
Hình 3.4. Sơ đồ kết nối module hồng ngoại FC-51.....	41
Hình 3.5. Sơ đồ kết nối LCD.....	42
Hình 3.6. Sơ đồ kết nối khối RFID.....	44
Hình 3.7. Lưu đồ phần mềm quản lý trên PC.....	47
Hình 3.8. Lưu đồ giải thuật servo và nút nhấn.....	48
Hình 3.9. Lưu đồ giải thuật LCD.....	49
Hình 3.10. Lưu đồ giải thuật RFID Reader.....	50
Hình 4.1. Mạch in toàn mạch.....	52
Hình 4.2. Board mạch hoàn thiện.....	53
Hình 4.3. Mô hình bãi giữ xe.....	53
Hình 4.4. Giao diện chính phần mềm quản lý bãi xe.	55
Hình 4.5. Giao diện SET THẺ trên phần mềm quản lý bãi xe.....	55
Hình 4.6. Giao diện phần mềm chế độ Fullscreen.	56
Hình 4.7. Sơ đồ hoạt động của Web.....	56
Hình 4.8. Trang chính của Web.....	57
Hình 4.9. Trang booking.....	57
Hình 4.10. Trang chọn chỗ Chooseat.	58
Hình 5.1. Trang điền thông tin.	60
Hình 5.2. Trang chọn chỗ khi chưa chọn chỗ A5.....	60
Hình 5.3. Đặt chỗ thành công.....	61

Hình 5.4. Giao diện khi book chỗ thành công.....	61
Hình 5.5. Giao diện Log In vào hệ thống.....	62
Hình 5.6. Giao diện Set thẻ.....	62
Hình 5.7. Giao diện chụp hình khi xe đúng biển số.....	63
Hình 5.8. Giao diện chụp hình khi xe sai biển số.....	63

Danh sách bảng

Bảng 2.1. Thuộc tính của văn bản trong HTLM.	13
Bảng 2.2. Bảng màu cơ bản trong HTML.	14
Bảng 2.3. Các thuộc tính cơ bản của CSS.	16
Bảng 2.4. Danh sách các quyền trong PHP.	27
Bảng 2.5. Ưu & Nhược điểm của CSDL.	29
Bảng 2.6. Quy trình xử lý nhận diện ảnh.	33
Bảng 3.1. Bảng kết nối chân giữa LCD và Adruino.	42
Bảng 3.2. Sơ đồ kết nối RC522 với Arduino	43
Bảng 3.3. Dòng và áp quy định các thiết bị trong đồ án.	45

Danh sách các từ viết tắt

VAMA	Vietnam Automobile Manufacturers Association	Hiệp hội các nhà sản xuất ô tô Việt Nam
RFID	Radio Frequency Identification	Nhận dạng qua tần số vô tuyến
IoT	Internet of Things	Mạng lưới vạn vật kết nối Internet
UART	Universal Asynchronous Receiver – Transmitter	Truyền dữ liệu nối tiếp bất đồng bộ
TTL	Transistor-Transistor Logic	Một lớp mạch kỹ thuật số được xây dựng từ các transistor lưỡng cực
SPI	Serial Peripheral Interface	Chuẩn truyền thông nối tiếp
ECMA	European Computer Manufacturers Association	Hiệp hội các nhà sản xuất máy tính châu Âu
ISO	International Standards Organization	Tổ chức tiêu chuẩn hóa quốc tế
CLI	Common Language Infrastructure	
HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
PHP	Hypertext Preprocessor	Ngôn ngữ lập trình kịch bản
OpenCV	Open Source Computer Vision	
EmguCV	Emgu Computer Vision	
OCR	Optical Character Recognition	Thuật toán nhận diện ký tự bằng quang học

LCD	Liquid Crystal Display	Màn hình tinh thể lỏng
MISO	Master Input Slave Output	Chân mang dữ liệu từ các thiết bị SPI về vi điều khiển
MOSI	Master Output Slave Input	Chân mang dữ liệu từ vi điều khiển đến các thiết bị SPI
SCK	Serial Clock	Chân giữ xung nhịp trong giao tiếp SPI
SDA	Serial Data Line	Dây truyền dữ liệu
I2C	Inter-Integrated Circuit	Giao tiếp giữa các IC

Chương 1. Tổng quan

1.1. Đặt vấn đề

Dựa trên số liệu thống kê mới nhất từ VAMA (Hiệp hội các nhà sản xuất ô tô Việt Nam), tổng lượng bán ô tô mới trong tháng 3/2017 là 26.872 xe, tăng 52% so với tháng 2 trước đó và tăng 8% so với cùng kỳ năm ngoái. Gộp chung lại trong Quý 1/2017, thị trường đã tiêu thụ 41.600 xe, tăng 8% so cùng kỳ năm trước. Có thể thấy lượng xe ô tô ngày một tăng, song song với vấn đề đó, người ta đặt vấn đề là xây dựng những bãi giữ xe để phục vụ cho người dân trong công việc cũng như trong việc đi lại của họ.

Yêu cầu đặt ra là cần phải áp dụng các tiến bộ về khoa học kỹ thuật cho các bãi giữ xe tiến đến tối ưu hóa tự động và thông minh. Các công nghệ mới tiên tiến đã ra đời như công nghệ RFID, các công nghệ điều khiển và giám sát từ xa, công nghệ IoT (Internet of Thing) ... đã và luôn được ưu tiên ứng dụng vào các hệ thống hiện đại và ứng dụng nó cho các bãi giữ xe thông minh là một thành công điển hình.

Có thể thấy được ở hình thức giữ xe truyền thống đã mang lại nhiều vấn đề tồn đọng và cần được thay thế. Ví dụ như mỗi vé giữ xe chỉ sử dụng một lần, hết lượt xe của khách ra khỏi bãi là nó trở thành rác và không thể tái sử dụng. Ngoài ra, mỗi khi trời mưa vé xe dễ bị thấm nước, bị mờ số xe và nhân viên không thể nhận diện được, đôi khi khách hàng còn làm rách. Ở hệ thống bãi giữ xe thông minh thì tất cả những điều đó đều được giải quyết một cách triệt để, thẻ xe là thẻ điện từ, chống thấm nước khi trời mưa, tiện và gọn khách hàng và chủ đầu tư dễ dàng bảo quản và thời gian sử dụng của thẻ điện từ rất lâu khoảng 10 năm hoặc có thể hơn tùy vào mức độ bảo quản, mức độ an toàn cũng như an ninh được nâng cao hơn khi hạn chế được các hành vi giả mạo.

1.2. Tình hình nghiên cứu hiện nay

1.2.1. Tình hình nghiên cứu trong nước

Hiện nay tình trạng bãi giữ xe dần trở nên là một vấn đề nan giải ở các thành phố lớn trong nước ta vào các dịp lễ Tết cũng như ngày thường. Về vấn đề các bãi xe tự phát không đảm bảo về an ninh, chất lượng quản lý, cũng như việc tính tiền đội giá đã vấy lên nhiều vấn đề lo ngại. Vì thế ta có thể thấy nhu cầu về các bãi giữ xe thông minh, an toàn là cực kì lớn.

Gần đây 4 bãi giữ xe thông minh được đề xuất xây ở Sài Gòn, các bãi đậu cao tầng, lắp ghép, được đề xuất xây tại 4 vị trí khác nhau ở trung tâm TP HCM nhằm giải quyết cấp bách nhu cầu đỗ xe của người dân. Trình bày với lãnh đạo UBND TP HCM chiều 29/3, một doanh nghiệp đưa ra các giải pháp tổ chức bãi giữ xe thông minh ở trung tâm thành

phổ. Công ty này đề xuất xây bãi giữ xe thông minh cao tầng, lắp ghép, tại Công trường Lam Sơn (rộng 1.410 m²); Công viên 23/9 (4.048 m²); Công viên Lê Văn Tám (1.416 m²) và Công viên Tao Đàn (570 m²). Khi hoàn thành, các bãi có thể giữ khoảng 500 ô tô cùng lúc. Với công nghệ robot tự động xoay vòng, khi ô tô vào vị trí đậu, hệ thống tự động chuyển xe lên vị trí cao hơn theo thứ tự. Nếu chủ xe muốn lấy, hệ thống tự động đưa ô tô về vị trí ban đầu để chuyển sang trạng thái giao thông động.

Cho tới thời điểm này hầu hết các trung tâm thương mại, siêu thị lớn đều đã tích hợp các bãi xe thông minh sử dụng công nghệ RFID và đạt bước thành công đột phá. Tuy nhiên vấn đề kết hợp công nghệ IoT vào bãi xe vẫn còn đang là một hướng nghiên cứu mới khi Internet vạn vật đang dần thâm nhập vào hầu hết các quy trình công nghệ.

1.2.2. Tình hình nghiên cứu ngoài nước

Ngày nay trên các nước tiên tiến trên thế giới như Nhật Bản, Hàn Quốc... ở những thành phố chật hẹp, người ta xây dựng hệ thống bãi giữ xe ô tô tự động được trang bị thiết bị nâng để di chuyển ô tô từ mặt đất lên điểm đỗ trên cao (hệ thống nổi hoặc di chuyển xe xuống điểm đỗ dưới lòng đất (hệ thống ngầm). Đây là những giải pháp giúp tăng hơn 100 lần số lượng xe trên một diện tích truyền thống, cho phép giải quyết tình trạng thiếu mặt bằng xây dựng.

Ở các nước phát triển các công nghệ mới như RFID và IoT đã và đang được ứng dụng rộng rãi trong hầu hết các lĩnh vực. Việc ứng dụng các công nghệ mới đã góp phần phát triển kinh tế và bảo vệ môi trường. Nạn kẹt xe hay thiếu chỗ và vấn đề về bảo mật, an ninh, sự không hài lòng về chất lượng quản lý ở các bãi giữ xe đã không còn nữa vì các bãi giữ xe truyền thống đã dần biến mất.

1.3. Tính cấp thiết của đề tài

Như vấn đề đã đặt ra thì nhu cầu sử dụng bãi giữ xe thông minh trên thị trường Việt Nam ngày càng tăng cao. Nắm bắt được tình hình đó nhiều công ty công nghệ đã không ngừng phát triển các hệ thống bãi giữ xe thông minh nhằm đáp ứng nhu cầu thực tế mang lại nhiều lợi nhuận.

Ở các hình thức giữ xe truyền thống, các doanh nghiệp đã gặp phải rất nhiều bất cập và gây khó khăn cho quá trình quản lý cũng như cho nhân viên an ninh. Đồng thời các nhân viên cũng dễ dàng làm thất thoát tài chính, dễ lấy tiền của chủ đầu tư. Ngoài ra, ở các bãi giữ xe thông thường như: xé vé tay, ghi phần lên xe... các hình thức giữ xe truyền thống rất tốn kém, không an toàn, gây lãng phí và ô nhiễm môi trường do lượng giấy thải ra ngoài môi trường.

Để khắc phục vấn đề đó đề tài sau nghiên cứu sẽ phần nào góp phần vào việc giải quyết nhu cầu về bãi xe, cũng như tích hợp thêm công nghệ mới IoT vào để cải thiện hiệu quả và tính linh hoạt, thông minh của các hệ thống hiện có.

So với các đề tài bãi giữ xe nhóm đã tham khảo thì đề tài nhóm đang nghiên cứu được phát triển hơn bởi công nghệ nhận dạng biển số và thiết kế web đặt chỗ giúp cho công việc được thuận tiện và ngày càng đa dạng, thông minh dần thay thế sức lao động của con người trong tương lai.

1.4. Mục đích của đề tài

Thông qua đề tài này nhóm muốn thiết kế và thi công một hệ thống bãi giữ xe thông minh gồm mô hình bãi xe tự động và phần mềm quản lý trên PC ứng dụng công nghệ RFID kết hợp nhận dạng biển số xe.

Ứng dụng công nghệ IoT vào bãi xe để giải quyết các vấn đề về đặt chỗ và giám sát thông minh, mang lại nhiều sự tiện lợi cũng như tính mới cho hệ thống.

1.5. Phương pháp nghiên cứu

Trong đề tài này, nhóm đã sử dụng các phương pháp nghiên cứu sau:

- Phương pháp tham khảo tài liệu: bằng cách thu thập thông tin từ sách, tạp chí về điện tử, viễn thông, truy cập từ mạng internet, các đồ án của khóa trước.
- Phương pháp quan sát: khảo sát một số mạch điện từ mạng internet, khảo sát các bãi giữ xe thông minh hiện hành để chọn lựa phương án thiết kế sau này.
- Phương pháp khảo sát và thực nghiệm: từ những ý tưởng và kiến thức của nhóm, kết hợp sự hướng dẫn của giáo viên, nhóm đã lắp ráp thử nghiệm nhiều dạng mạch khác nhau để từ đó chọn lọc những mạch điện tối ưu.

1.6. Bố cục đề án

Chương 1: Tổng quan

Giới thiệu sơ lược về tình hình nghiên cứu hiện nay cũng như tính cấp thiết của đề tài.

Chương 2: Cơ sở lý thuyết

Nêu các lý thuyết cần thiết để sử dụng trong đề tài.

Chương 3: Thiết kế và xây dựng hệ thống

Trình bày sơ đồ hệ thống và giải thích hoạt động của hệ thống.

Đưa ra các phương pháp lựa chọn phần cứng và xác định lựa chọn phù hợp với yêu cầu của đề tài.

Chương 4: Kết quả thực hiện

Tính toán đưa ra giải thuật, thuật toán phần mềm.

Trình bày kết quả đã thực hiện về phần cứng và phần mềm.

Chương 5: Kết quả so sánh, thực nghiệm, phân tích, tổng hợp

Đưa ra các kết quả thực nghiệm và đánh giá, nhận xét hệ thống.

Chương 6: Kết luận và hướng phát triển

Nêu các ưu điểm và các điểm cần cải thiện của đề tài, hướng khắc phục và hướng phát triển trong tương lai.

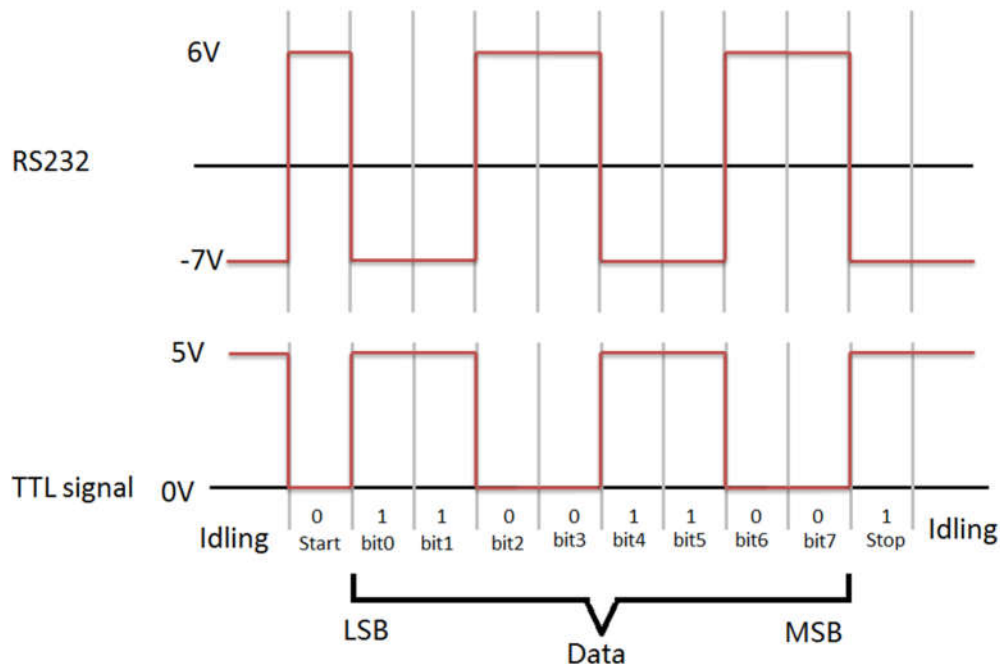
Chương 2. Cơ sở lý thuyết

Để tiến hành thiết kế phần cứng cũng như phần mềm nhóm đã tìm hiểu một số khái niệm sau

2.1. Tổng quan về các chuẩn truyền

2.1.1. Tổng quan về UART

Khái niệm USART (hay UART nếu chỉ nói đến bộ truyền nhận không đồng bộ) thường để chỉ thiết bị phần cứng (device, hardware), không phải chỉ một chuẩn giao tiếp. USART hay UART cần phải kết hợp với một thiết bị chuyển đổi mức điện áp để tạo ra một chuẩn giao tiếp nào đó. Tín hiệu theo chuẩn RS232 trên máy tính cá nhân thường là -12V cho mức logic high và +12V cho mức logic low. Các giải thích trong tài liệu này theo mức logic TTL của USART không theo RS232.



Hình 2.1. Tín hiệu tương đương của UART và RS232.

Giả sử khi ta đang xây dựng một ứng dụng phức tạp cần sử dụng nhiều vi điều khiển (hoặc vi điều khiển và máy tính) kết nối với nhau. Trong quá trình làm việc các vi điều khiển cần trao đổi dữ liệu cho nhau, ví dụ tình huống Master truyền lệnh cho Slaver hoặc Slaver gửi tín hiệu thu thập được về Master xử lý... Giả sử dữ liệu cần trao đổi là các mã có chiều dài 8 bits, ta có thể sẽ nghĩ đến cách kết nối đơn giản nhất là kết nối 1 PORT (8 bit) của mỗi vi điều khiển với nhau, mỗi line trên PORT sẽ chịu trách nhiệm truyền/nhận 1-bit dữ liệu. Đây gọi là cách giao tiếp song song, cách này là cách đơn giản nhất vì dữ liệu được

xuất và nhận trực tiếp không thông qua bất kỳ một giải thuật biến đổi nào và vì thế tốc độ truyền cũng rất nhanh. Nhược điểm của cách truyền này là số đường truyền quá nhiều, tưởng tượng nếu dữ liệu của ta có giá trị càng lớn thì số đường truyền cũng sẽ nhiều thêm. Hệ thống truyền thông song song thường rất cồng kềnh và vì thế kém hiệu quả. Truyền thông nối tiếp sẽ giải quyết vấn đề này, trong truyền thông nối tiếp dữ liệu được truyền từng bit trên 1 (hoặc một ít) đường truyền. Vì lý do này, cho dù dữ liệu của ta có lớn đến đâu ta cũng chỉ dùng rất ít đường truyền.

Khái niệm “đồng bộ” để chỉ sự “báo trước” trong quá trình truyền. Lấy ví dụ thiết bị 1 kết với thiết bị 2 bởi 2 đường, một đường dữ liệu và 1 đường xung nhịp. Cứ mỗi lần thiết bị 1 muốn send 1-bit dữ liệu, thiết bị 1 điều khiển đường xung nhịp chuyển từ mức thấp lên mức cao báo cho thiết bị 2 sẵn sàng nhận một bit. Bằng cách “báo trước” này tất cả các bit dữ liệu có thể truyền/nhận dễ dàng với ít “rủi ro” trong quá trình truyền. Tuy nhiên, cách truyền này đòi hỏi ít nhất 2 đường truyền cho 1 quá trình (send or receive). Giao tiếp giữa máy tính và các bàn phím (trừ bàn phím kết nối theo chuẩn USB) là một ví dụ của cách truyền thông nối tiếp đồng bộ.

Khác với cách truyền đồng bộ, truyền thông “không đồng bộ” chỉ cần một đường truyền cho một quá trình. “Khung dữ liệu” đã được chuẩn hóa bởi các thiết bị nên không cần đường xung nhịp báo trước dữ liệu đến. Ví dụ 2 thiết bị đang giao tiếp với nhau theo phương pháp này, chúng đã được thỏa thuận với nhau rằng cứ 1ms thì sẽ có 1 bit dữ liệu truyền đến, như thế thiết bị nhận chỉ cần kiểm tra và đọc đường truyền mỗi mili-giây để đọc các bit dữ liệu và sau đó kết hợp chúng lại thành dữ liệu có ý nghĩa. Truyền thông nối tiếp không đồng bộ vì thế hiệu quả hơn truyền thông đồng bộ (không cần nhiều lines truyền). Tuy nhiên, để quá trình truyền thành công thì việc tuân thủ các tiêu chuẩn truyền là hết sức quan trọng. Chúng ta sẽ bắt đầu tìm hiểu các khái niệm quan trọng trong phương pháp truyền thông này.

Baud rate (tốc độ Baud): như trong ví dụ trên về việc truyền 1 bit trong 1ms, ta thấy rằng để việc truyền và nhận không đồng bộ xảy ra thành công thì các thiết bị tham gia phải “thống nhất” nhau về khoảng thời dành cho 1 bit truyền, hay nói cách khác tốc độ truyền phải được cài đặt như nhau trước, tốc độ này gọi là tốc độ Baud. Theo định nghĩa, tốc độ baud là số bit truyền trong 1 giây. Ví dụ nếu tốc độ baud được đặt là 19200 thì thời gian dành cho 1 bit truyền là $1/19200 \sim 52.083\mu s$.

Frame (khung truyền): do truyền thông nối tiếp mà nhất là nối tiếp không đồng bộ rất dễ mất hoặc sai lệch dữ liệu, quá trình truyền thông theo kiểu này phải tuân theo một số quy cách nhất định. Bên cạnh tốc độ baud, khung truyền là một yếu tố quan trọng tạo nên

sự thành công khi truyền và nhận. Khung truyền bao gồm các quy định về số bit trong mỗi lần truyền, các bit “báo” như bit Start và bit Stop, các bit kiểm tra như Parity, ngoài ra số lượng các bit trong một data cũng được quy định bởi khung truyền.

Start bit: start là bit đầu tiên được truyền trong một frame truyền, bit này có chức năng báo cho thiết bị nhận biết rằng có một gói dữ liệu sắp được truyền tới. Ở module USART trong các vi điều khiển, đường truyền thường ở trạng thái cao khi nghỉ (Idle), nếu một vi điều khiển muốn thực hiện việc truyền dữ liệu nó sẽ gởi một bit start bằng cách “kéo” đường truyền xuống mức 0. Như vậy, với các vi điều khiển bit start thường mang giá trị 0 và có giá trị điện áp 0V. Start là bit bắt buộc phải có trong khung truyền.

Data: data hay dữ liệu cần truyền là thông tin chính mà chúng ta cần gởi và nhận. Data không nhất thiết phải là gói 8 bit, với vi điều khiển có thể quy định số lượng bit của data là 5, 6, 7, 8 hoặc 9 (tương tự cho hầu hết các thiết bị hỗ trợ UART khác). Trong truyền thông nối tiếp UART, bit có ảnh hưởng nhỏ nhất (LSB – Least Significant Bit, bit bên phải) của data sẽ được truyền trước và cuối cùng là bit có ảnh hưởng lớn nhất (MSB – Most Significant Bit, bit bên trái).

2.1.2. Tổng quan về chuẩn giao tiếp SPI

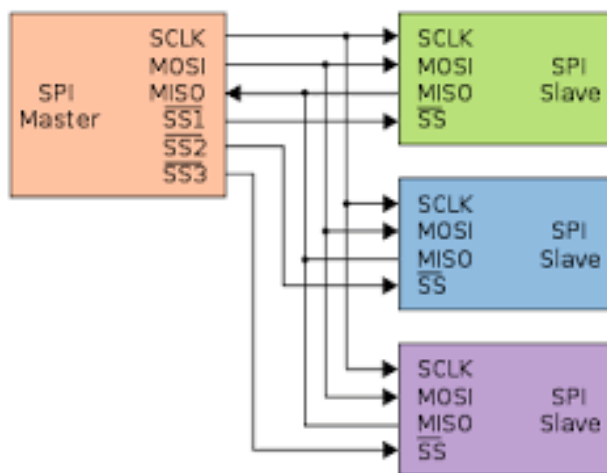
SPI là một chuẩn truyền thông nối tiếp tốc độ cao do hãng Motorola đề xuất. Đây là kiểu truyền thông Master-Slave, trong đó có 1 chip Master điều phối quá trình truyền thông và các chip Slaves được điều khiển bởi Master vì thế truyền thông chỉ xảy ra giữa Master và Slave. SPI là một cách truyền song công (full duplex) nghĩa là tại cùng một thời điểm quá trình truyền và nhận có thể xảy ra đồng thời. SPI đôi khi được gọi là chuẩn truyền thông “4 dây” vì có 4 đường giao tiếp trong chuẩn này đó là SCK (Serial Clock), MISO (Master Input Slave Output), MOSI (Master Output Slave Input) và SS (Slave Select).

SCK: Xung giữ nhịp cho giao tiếp SPI, vì SPI là chuẩn truyền đồng bộ nên cần 1 đường giữ nhịp, mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi. Đây là điểm khác biệt với truyền thông không đồng bộ mà chúng ta đã biết trong chuẩn UART. Sự tồn tại của chân SCK giúp quá trình truyền ít bị lỗi và vì thế tốc độ truyền của SPI có thể đạt rất cao. Xung nhịp chỉ được tạo ra bởi chip Master.

MISO– Master Input / Slave Output: nếu là chip Master thì đây là đường Input còn nếu là chip Slave thì lại là Output. MISO của Master và các Slaves được nối trực tiếp với nhau.

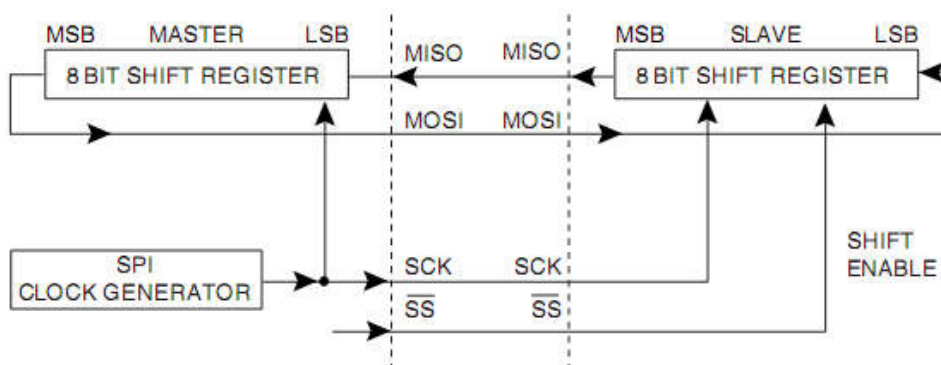
MOSI – Master Output / Slave Input: nếu là chip Master thì đây là đường Output còn nếu là chip Slave thì là Input. MOSI của Master và các Slaves được nối trực tiếp với nhau.

SS – Slave Select: SS là đường chọn Slave cần giao tiếp, trên các chip Slave đường SS sẽ ở mức cao khi không làm việc. Nếu chip Master kéo đường SS của một Slave nào đó xuống mức thấp thì việc giao tiếp sẽ xảy ra giữa Master và Slave đó. Chỉ có 1 đường SS trên mỗi Slave nhưng có thể có nhiều đường điều khiển SS trên Master, tùy thuộc vào thiết kế của người dung.



Hình 2.2. Giao diện SPI.

Về hoạt động mỗi chip Master hay Slave có một thanh ghi dữ liệu 8 bits. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, một bit trong thanh ghi dữ liệu của Master được truyền qua Slave trên đường MOSI, đồng thời một bit trong thanh ghi dữ liệu của chip Slave cũng được truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip được gửi qua lại đồng thời nên quá trình truyền dữ liệu này được gọi là “song công”.



Hình 2.3. Truyền dữ liệu SPI.

Cực của xung giữ nhịp, phase và các chế độ hoạt động: cực của xung giữ nhịp (Clock Polarity) được gọi tắt là CPOL là khái niệm dùng chỉ trạng thái của chân SCK ở trạng thái nghỉ. Ở trạng thái nghỉ (Idle), chân SCK có thể được giữ ở mức cao (CPOL=1) hoặc thấp (CPOL=0). Phase (CPHA) dùng để chỉ cách mà dữ liệu được lấy mẫu (sample) theo xung

giữ nhịp. Dữ liệu có thể được lấy mẫu ở cạnh lên của SCK (CPHA=0) hoặc cạnh xuống (CPHA=1). Sự kết hợp của SPOL và CPHA làm nên 4 chế độ hoạt động của SPI. Nhìn chung việc chọn 1 trong 4 chế độ này không ảnh hưởng đến chất lượng truyền thông mà chỉ cốt sao cho có sự tương thích giữa Master và Slave.

2.1.3. Tổng quan về I2C

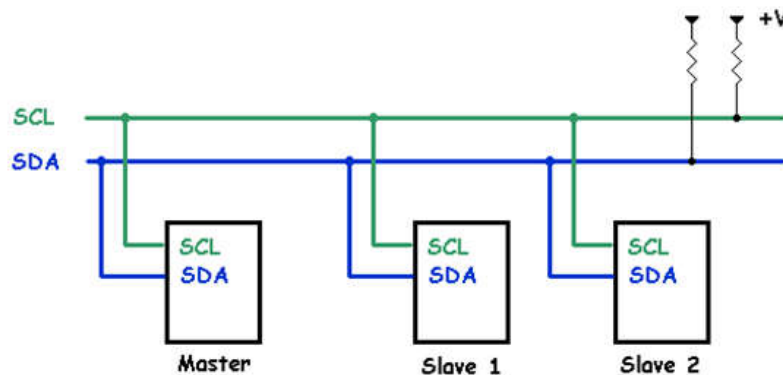
I2C là một loại bus nối tiếp được phát triển bởi hãng sản xuất linh kiện điện tử Philips. Ban đầu, loại bus này chỉ được dùng trong các linh kiện điện tử của Philips. Sau đó, do tính ưu việt và đơn giản của nó, I2C đã được chuẩn hóa và được dùng rộng rãi trong các module truyền thông nối tiếp của vi mạch tích hợp ngày nay.

I2C sử dụng hai đường truyền tín hiệu: một đường xung nhịp đồng hồ (SCL) và một đường dữ liệu (SDA). SCL và SDA luôn được kéo lên nguồn bằng một điện trở kéo lên có giá trị xấp xỉ 4,7 KOhm. Các chế độ hoạt động của I²C bao gồm:

- Chế độ chuẩn (standard mode) hoạt động ở tốc độ 100 Kbit/s.
- Chế độ tốc độ thấp (low-speed mode) hoạt động ở tốc độ 10 Kbit/s.

Tần số xung nhịp đồng hồ có thể xuống 0 Hz. I2C sử dụng 7 bit để định địa chỉ, do đó trên một bus có thể định địa chỉ tới 112 nút, 16 địa chỉ còn lại được sử dụng vào mục đích riêng. Điểm mạnh của I2C chính là hiệu suất và sự đơn giản của nó: một khối điều khiển trung tâm có thể điều khiển cả một mạng thiết bị mà chỉ cần hai lối ra điều khiển.

Ngoài ra I2C còn có chế độ 10 bit địa chỉ tương đương với 1024 địa chỉ, tương tự như 7 bit, chỉ có 1008 thiết bị có thể kết nối, còn lại 16 địa chỉ sẽ dùng để sử dụng vào mục đích riêng.



Hình 2.4. Giao tiếp I2C.

2.2. Tổng quan về công nghệ RFID

Công nghệ RFID là một trong những công nghệ nhận dạng dữ liệu tự động tiên tiến nhất hiện nay có tính khả thi cao và áp dụng trong thực tế rất hiệu quả. RFID đang hiện diện trong rất nhiều lĩnh vực tự động hóa, rất nhiều ứng dụng quản lý và các mô hình tổ chức khác nhau nhằm đem lại những giải pháp nhận dạng dữ liệu tự động tối ưu và hiệu quả hơn.

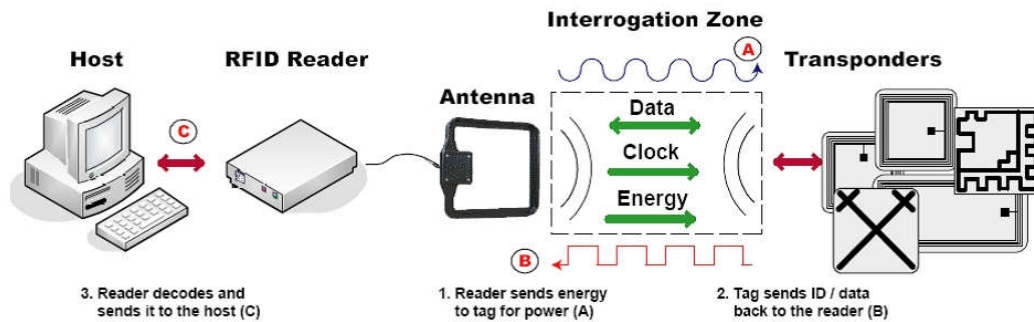
Công nghệ RFID cho phép một thiết bị đọc thông tin chứa trong chip không tiếp xúc trực tiếp ở khoảng cách xa, không thực hiện bất kỳ giao tiếp vật lý nào hoặc giữa hai vật không nhìn thấy. Công nghệ này cho ta phương pháp truyền, nhận dữ liệu từ một điểm đến điểm khác.

Kỹ thuật RFID sử dụng truyền thông không dây trong dải tần sóng vô tuyến để truyền dữ liệu từ các tag (thẻ) đến các reader (bộ đọc). Tag có thể được đính kèm hoặc gắn vào đối tượng được nhận dạng chẳng hạn sản phẩm, hộp hoặc giá kệ (pallet). Reader scan dữ liệu của tag và gửi thông tin đến cơ sở dữ liệu có lưu trữ dữ liệu của tag. Chẳng hạn, các tag có thể được đặt trên kính chắn gió xe hơi để hệ thống thu phí đường có thể nhanh chóng nhận dạng và thu tiền trên các tuyến đường

Dạng đơn giản nhất được sử dụng hiện nay là hệ thống RFID thụ động làm việc như sau: reader truyền một tín hiệu tần số vô tuyến điện từ qua anten của nó đến một con chip. Reader nhận thông tin trở lại từ chip và gửi nó đến máy tính điều khiển đầu đọc và xử lý thông tin lấy được từ chip. Các chip không tiếp xúc không tích điện, chúng hoạt động bằng cách sử dụng năng lượng nhận từ tín hiệu được gửi bởi reader.

Một hệ thống RFID gồm những thành phần cơ bản sau:

- Thẻ RFID (RFID Tag, còn được gọi là transponder): là một thẻ gắn chip cộng antena. Gồm 2 loại: RFID passive tag và active tag:
 - Passive tags: Không cần nguồn ngoài và nhận năng lượng từ thiết bị đọc. Khoảng cách đọc ngắn.
 - Active tags: Được nuôi bằng PIN, sử dụng với khoảng cách đọc lớn.
- Reader hoặc sensor: để đọc thông tin từ các thẻ, có thể đặt cố định hoặc lưu động.
- Antenna: Là thiết bị liên kết giữa thẻ và thiết bị đọc. Thiết bị đọc phát xạ tín hiệu sóng để kích hoạt và truyền nhận với thẻ.
- Server: nhu nhận, xử lý dữ liệu, phục vụ giám sát, thống kê, điều khiển...



Hình 2.5. Hệ thống RFID cơ bản.

Về dải tần hoạt động của hệ thống RFID. Khi phải lựa chọn một hệ thống RFID, yêu cầu đầu tiên là chọn dải tần hoạt động của hệ thống.

- Tần số thấp - Low frequency 125 KHz: Dải đọc ngắn tốc độ đọc thấp.
- Dải tần cao - High frequency 13.56 MHz: Khoảng cách đọc ngắn tốc độ đọc trung bình. Phần lớn thẻ Passive sử dụng dải này.
- Dải tần cao hơn - High frequency: Dải đọc từ ngắn đến trung bình, tốc độ đọc trung bình đến cao. Phần lớn thẻ Active sử dụng tần số này.
- Dải siêu cao tần - UHF frequency 860-960 MHz. Dải đọc rộng Tốc độ đọc cao. Phần lớn dùng thẻ Active và một số thẻ Passive cao tần sử dụng dải này.
- Dải vi sóng - Microwave 2.45-5.8 GHz: Dải đọc rộng tốc độ đọc lớn.

2.3. Tổng quan về ngôn ngữ lập trình Web

2.3.1. HTML

HTML là ngôn ngữ đánh dấu siêu văn bản chỉ rõ một trang Web sẽ được hiển thị như thế nào trên trình duyệt. Bằng cách sử dụng các thẻ và các phần tử html ta có thể:

- Điều khiển hình thức và nội dung của trang.
- Xuất bản các tài liệu trực tuyến và truy xuất thông tin trực tuyến bằng cách sử dụng các liên kết được chèn vào tài liệu html.
- Tạo các biểu mẫu trực tuyến để thu thập thông tin về người dùng, quản lý các giao dịch ...
- Chèn các đối tượng như audio clip, video clip, các thành phần ActiveX, Flash và các Java Applet vào tài liệu html.

HTML tạo thành mã nguồn của trang Web. Khi được xem trên trình soạn thảo, tài liệu này là một chuỗi các thẻ và các phần tử, mà chúng xác định trang web hiển thị như thế nào. Trình duyệt đọc các file có đuôi .htm hay .html và hiển thị trang web đó theo các lệnh

có trong đó. Tất cả các trang web dù xử lý phức tạp đến đâu đều phải trả về dưới dạng mã nguồn HTML để trình duyệt có thể hiểu và hiển thị lên được.

Về cấu trúc cơ bản của HTML, để bắt đầu một file html, ta cần sử dụng thẻ `<html>` để mở và kết thúc bằng thẻ `</html>`. Bên trong một file html gồm có 2 phần cơ bản:

- Phần tiêu đề: Phần tiêu đề bắt đầu bằng thẻ `<head>` và kết thúc bởi thẻ `</head>`. Phần này chứa tiêu đề mà được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ `title`, bắt đầu bằng thẻ `<title>` và kết thúc là thẻ `</title>`. Tiêu đề là phần khá quan trọng. Khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khóa chính yếu cho việc tìm kiếm.
- Phần thân: phần này nằm sau phần tiêu đề. Phần thân bao gồm văn bản, hình ảnh và các liên kết mà ta muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ `<body>` và kết thúc bằng thẻ `</body>`.

Ta sẽ đề cập đến các thẻ HTML cơ bản như Headings (tiêu đề), thẻ khối ``; `<div>`, cách sử dụng font, cách dùng màu, cách tạo bảng, cách chèn ảnh, các thẻ liên kết, thẻ thu thập thông tin.

Headings được sử dụng để trình bày tiêu đề cho phần nội dung hiển thị trên trang Web. Những phần tiêu đề được hiển thị to và in đậm hơn để phân biệt chúng với các phần còn lại của văn bản. Chúng ta cũng có thể hiển thị phần tiêu đề theo một trong sáu kích thước từ h1 đến h6. Tất cả những gì chúng ta làm là định rõ kích thước h1, h2... Thẻ h1 dành cho các tiêu đề quan trọng nhất và giảm dần đến h6.

```
//Ví dụ:  
<h1>This is a man</h1>  
<h2>This is a man</h2>  
<h3>This is a man</h3>
```

Kết quả khi hiển thị ra trang web tương ứng:

This is a man

This is a man

This is a man

Về thẻ khối `` và `<div>`, có những trường hợp chúng ta muốn chia văn bản trong một trang web thành những khối thông tin logic khi đó phần tử `div` và `span` được sử dụng để nhóm nội dung lại với nhau. Thẻ `div` rất thường được sử dụng trong thiết kế layout website.

- Phần tử div dùng để chia tài liệu thành các thành phần có liên quan với nhau.
- Phần tử span dùng để định nghĩa nội dung trong dòng (in-line) còn phần tử div dùng để định nghĩa nội dung mức khối (block-level).

Về cách sử dụng font, thẻ dùng để điều khiển sự hiển thị văn bản trên trang Web. Ngoài ra cũng có thể chỉ định các thuộc tính như kích thước, màu sắc, kiểu chữ... Có thể đặt các thuộc tính cho cả tài liệu bằng cách đặt phần tử vào bên trong thẻ <body>. Ngoài ra, thuộc tính font có thể đặt cho từng từ, từng block trong trang. Các thuộc tính của FONT có thể kết hợp trong cùng một thẻ.

Bảng 2.1. Thuộc tính của văn bản trong HTML.

Thuộc tính	Mô tả
COLOR	Được dùng để chỉ màu của font. Chúng ta có thể dùng tên màu hoặc giá trị thập phân để xác định màu.
SIZE	Được dùng để chỉ kích thước của font. Chúng ta có thể xác định các kích thước FONT từ 1 cho đến 7. Kích thước lớn nhất là 7 và nhỏ nhất là 1. Chúng ta có thể dùng một kích thước chuẩn và chỉ ra những kích thước tiếp theo liên quan đến kích thước chuẩn. Ví dụ, nếu kích thước chính là 3, thì, SIZE = +4 sẽ tăng lên 7, SIZE = -1 sẽ giảm xuống 2
FACE	Được dùng để chỉ định kiểu font (phông chữ)

Về cách sử dụng màu sắc, ta có thể thêm màu vào trang và các phần tử trong trang. COLOR là thuộc tính có thể được sử dụng với nhiều phần tử như phần tử FONT và BODY. Có 3 kiểu màu chính: đỏ, xanh lá và xanh da trời. Mỗi màu chính được xem như một bộ hai số của hệ 16 - #RRGGBB. Số thập lục phân 00 chỉ 0% của màu trong khi đó số thập lục phân FF chỉ 100% của màu. Giá trị cuối cùng là một mã sáu chữ số chỉ màu.

Bảng 2.2. Bảng màu cơ bản trong HTML.

Mã thập lục phân	Màu
#FF0000	Red
#00FF00	Green
#0000FF	Blue
#000000	Black
#FFFFFF	White

Về cách chèn ảnh, ta dùng thẻ dùng để chèn hình ảnh vào trong HTML. Chúng ta cũng có thể đặt thẻ IMG tại vị trí mà hình ảnh được hiển thị. Thẻ IMG không có nội dung, nó hiển thị nội dung bằng cách xác định thuộc tính SRC. Cú pháp là . Trong đó SRC (source) là thuộc tính và giá trị là một URL, chỉ vị trí chính xác của file ảnh. Thuộc tính align của thẻ có thể được sử dụng để điều chỉnh canh lề của ảnh với văn bản xung quanh. . Trong đó, vị trí của ảnh có thể là trên (top), dưới (bottom), ở giữa (middle), trái (left) hoặc phải (right).

Về cách tạo bảng, ta dùng thẻ <table> dùng để tạo bảng biểu trong HTML. Cấu trúc cơ bản như sau:

```
<table>
<tr>
<td> ... </td>
</tr>
</table>
```

Trong đó thẻ <tr> cho biết bắt đầu 1 hàng, <td> là thẻ chỉ đến cột tương ứng với hàng đó.

Về thẻ liên kết, ta dùng thẻ <a> để tạo liên kết với một địa chỉ URL. Cấu trúc như sau:

```
<a href = " http://www.fhq.hcmute.edu.vn"> FHQ-HCMUTE </a>
```

Trên giao diện web sẽ hiển thị chữ “FHQ-HCMUTE” và khi click vào dòng chữ này sẽ liên kết tới một trang web (www.fhq.hcmute.edu.vn).

Về thẻ thu thập thông tin, thẻ <form> là thẻ dùng để thu thập thông tin từ người dùng, chẳng hạn như hồ sơ xin việc làm, mẫu thăm dò ý kiến... Cấu trúc như sau:

```
<form>
    <input type = "text" name = "username" />
    <input type = "password" name = "password" />
</form>
```

Với dạng "text" thì các ký tự hiển thị bình thường. Với dạng "password" thì các ký tự sẽ được thay thế bằng ký tự ‘*’.

2.3.2. CSS

CSS là một file có phần mở rộng là .css, nhiệm vụ của nó là tách riêng phần định dạng (style) ra khỏi nội dung trang HTML.

Khi sử dụng CSS chúng ta sẽ dễ dàng quản lý nội dung trang HTML, dễ điều khiển phần định dạng, và đặc biệt là sẽ tốn ít thời gian khi code hay chỉnh sửa, giả sử các ta có ~100 file HTML có tiêu đề như nhau, các tiêu đề này được định dạng trong từng trang HTML, khi các ta muốn chỉnh sửa các tiêu đề này, thì các ta sẽ phải mở và chỉnh từng trang trong ~100 file, việc này mất bao nhiêu thời gian chắc các ta có thể ước lượng được, còn nếu ~100 file này được kết hợp với 1 file CSS thì việc định dạng tiêu đề trong ~100 file này chỉ mất vài phút.

Chúng ta đừng nhầm lẫn CSS và "style trong HTML" là như nhau, style dùng để định dạng cho nội dung HTML, còn CSS là file chứa các định dạng style, 2 đoạn code style dưới đây không thể gọi là CSS:

HTML viết

```
<html>
<head>
<style>
p { color: #ff0000; }
</style>
</head>
```

```

<body>
<p>Một đoạn văn bản.</p>
</body>
</html>

```

HTML viết

```

<html>
<head>
</head>

<body>
<p style="color: #ff0000;">Một đoạn văn bản.</p>
</body>
</html>

```

Các thuộc tính cơ bản của CSS

Bảng 2.3. Các thuộc tính cơ bản của CSS.

Thuộc tính	Ví dụ	Mô tả
background	background: #ff0000;	Định dạng nền (background) cho thành phần.
border	border: 1px solid #ff0000;	Định dạng đường viền cho thành phần.
border-collapse	border-collapse: collapse;	Thuộc tính border-collapse xác định đường viền của table có tách biệt ra hay không.
border-spacing	border-spacing: 10px;	Xác định khoảng cách giữa các đường viền của các cột lân cận.
bottom	bottom: 10px;	Xác định vị trí dưới cùng của thành phần được định vị trí.

<u>caption-side</u>	caption-side: bottom;	Xác định vị trí một chú thích của <u>table</u> .
<u>clear</u>	clear: both;	Xác định 2 bên của phần tử (left, right), nơi mà phần tử float không được cho phép.
<u>clip</u>	clip: rect(0,0,50px,10px);	Xác định đoạn cho thành phần khi sử dụng thuộc tính position có giá trị "absolute".
<u>color</u>	color: #ff0000;	Xác định màu sắc cho text.
<u>content</u>	content: "."	Sử dụng kèm với bộ chọn ":before", ":after" để chèn nội dung được tạo.
<u>counter-increment</u>	counter-increment: section;	Gia tăng một hoặc nhiều counter (sắp xếp có thứ tự, có hiển thị số)
<u>counter-reset</u>	counter-reset: subsection;	Tạo hoặc reset một hoặc nhiều counter.
<u>cursor</u>	cursor: pointer;	Xác định kiểu con trỏ chuột được hiển thị.
<u>direction</u>	direction: ltr;	Xác định hướng cho văn bản.
<u>display</u>	display: inline;	Xác định loại hiển thị của thành phần.
<u>empty-cells</u>	empty-cells: hide;	Xác định có hay không có đường viền và nền trong một cột rỗng của <u>table</u>
<u>float</u>	float: left;	Xác định có hay không một thành phần được float.
<u>font</u>	font: 12px arial,sans-serif;	Thiết lập font cho thành phần, bao gồm font chữ, độ rộng, ...
<u>height</u>	height: 50px;	Thiết lập chiều cao của thành phần.
<u>left</u>	left: 10px;	Xác định vị trí bên trái của thành phần định vị trí (như position)

<u>letter-spacing</u>	letter-spacing: 2px;	Tăng hoặc giảm khoảng cách giữa các ký tự trong đoạn text.
<u>line-height</u>	line-height: 1.5;	Thiết lập chiều cao giữa các dòng.
<u>list-style</u>	list-style: decimal;	Thiết lập kiểu cho một danh sách.
<u>margin</u>	margin: 15px;	Canh lề cho thành phần.
<u>max-height</u>	max-height: 200px;	Thiết lập chiều cao tối đa của thành phần.
<u>max-width</u>	max-width: 900px;	Thiết lập chiều rộng tối đa của thành phần.
<u>min-height</u>	min-height: 100px;	Thiết lập chiều cao tối thiểu của thành phần.
<u>min-width</u>	min-width: 600px;	Thiết lập chiều rộng tối thiểu của thành phần.
<u>outline</u>	outline: dotted;	Định dạng các đường viền bao ngoài
<u>overflow</u>	overflow: scroll;	Xác định điều gì sẽ xảy ra nếu một thành phần box tràn nội dung.
<u>padding</u>	padding: 15px;	Thiết lập các thuộc tính padding trong một khai báo.
<u>page-break-after</u>	page-break-after: alway;	Xác định các phân chia văn bản ngay sau thành phần.
<u>page-break-before</u>	page-break-before: alway;	Xác định các phân chia văn bản ngay trước thành phần.
<u>page-break-inside</u>	page-break-inside: alway;	Xác định các phân chia văn bản ngay bên trong thành phần.
<u>position</u>	position: absolute;	Xác định loại của phương pháp định vị trí cho thành phần.

<u>quotes</u>	"" ""	Thiết lập các loại dấu ngoặc bao ngoài khi nhúng một trích dẫn.
<u>right</u>	right: 10px;	Xác định vị trí bên phải của thành phần định vị trí (như position)
<u>table-layout</u>	table-layout: fixed;	Thiết lập các thuật toán layout được sử dụng cho table.
<u>text-align</u>	text-align: center;	Sắp xếp các nội dung theo chiều ngang.
<u>text-decoration</u>	text-decoration: underline;	Xác định các trang trí thêm cho text.
<u>text-indent</u>	text-indent: 10px;	Ghi rõ thụt đầu dòng của dòng đầu tiên trong một khối văn bản.
<u>text-transform</u>	text-transform: uppercase;	Thiết lập các ký tự viết hoa cho văn bản.
<u>top</u>	top: 10px;	Xác định vị trí bên trên của thành phần định vị trí (như position)
<u>vertical-align</u>	vertical-align: middle;	Sắp xếp các nội dung theo chiều dọc.
<u>visibility</u>	visibility: visible;	Xác định thành phần có được nhìn thấy hay không.
<u>white-space</u>	white-space: nowrap;	Xác định khoảng trắng có bên trong thành phần được xử lý như thế nào.
<u>width</u>	width: 800px;	Thiết lập chiều rộng cho thành phần.
<u>word-spacing</u>	word-spacing: 5px;	Tăng hoặc giảm không gian giữa các từ trong đoạn văn bản.
<u>z-index</u>	z-index: 100;	Thiết lập thứ tự xếp chồng nhau của một thành phần vị trí.

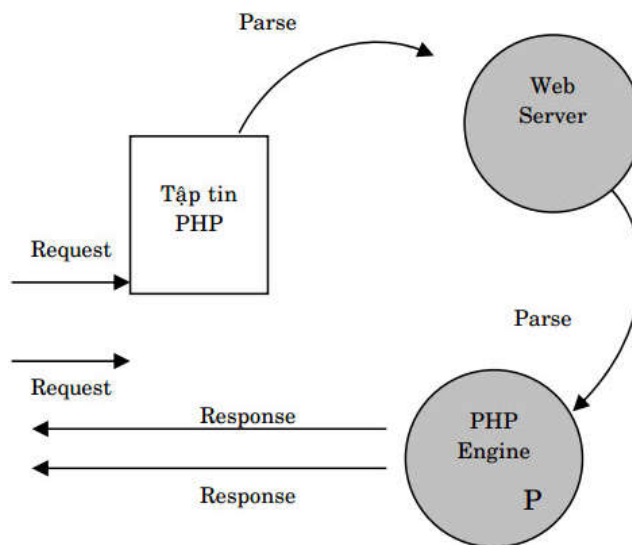
2.3.3. PHP

PHP là một ngôn ngữ lập trình kịch bản được chạy ở phía server nhằm sinh ra mã html trên client. PHP đã trải qua rất nhiều phiên bản và được tối ưu hóa cho các ứng dụng web, với cách viết mã rõ ràng, tốc độ nhanh, dễ học nên PHP đã trở thành một ngôn ngữ lập trình web rất phổ biến và được ưa chuộng.

PHP chạy trên môi trường Webserver và lưu trữ dữ liệu thông qua hệ quản trị cơ sở dữ liệu nên PHP thường đi kèm với Apache, MySQL và hệ điều hành Linux (LAMP).

- Apache là một phần mềm web server có nhiệm vụ tiếp nhận request từ trình duyệt người dùng sau đó chuyển giao cho PHP xử lý và gửi trả lại cho trình duyệt.
- MySQL cũng tương tự như các hệ quản trị cơ sở dữ liệu khác (Postgress, Oracle, SQL server...) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.
- Linux: Hệ điều hành mã nguồn mở được sử dụng rất rộng rãi cho các webserver. Thông thường các phiên bản được sử dụng nhiều nhất là RedHat Enterprise Linux, Ubuntu...

PHP hoạt động như sau, khi người sử dụng gọi trang PHP, Web Server sẽ triệu gọi PHP Engine để thông dịch dịch trang PHP và trả kết quả cho người dùng như hình 2.6.



Hình 2.6. Phương thức hoạt động của ngôn ngữ PHP.

File PHP có thể chứa text thuần, các thẻ HTML, các đoạn mã script khác. File PHP chạy và hiển thị trên trình duyệt ở dạng mã HTML. Một file php có thể chứa một trong các phần mở rộng sau: “.php”, “.php3”, “.phtml”. Mã lệnh của chương trình cần được đặt trong

cặp thẻ <?php?>. Sử dụng dấu hai chấm; để kết thúc một câu lệnh. Văn bản cần được đặt trong cặp dấu nháy đơn ' ' hoặc cặp dấu nháy kép " ".

Về cấu trúc (PHP syntax), 1 đoạn script PHP luôn luôn nằm trong cặp thẻ đóng/mở của PHP (<?php và ?>) và có thể đặt bất kỳ chỗ nào trong file. Ví dụ:

```
<html>
<body>
<?php
echo "Hello World"; //Hiện nội dung Hello World lên nền của trang web
?>
</body>
</html>
```

Ngoài ra, có thể dùng các cặp thẻ đóng mở sau <? và ?>, <?php và ?>, <script language="php"> và </script> hoặc <% và %>. Cách (1) chỉ có hiệu lực khi short_open_tag được mở trong file cấu hình PHP tương tự, cách (4) cũng thực hiện khi asp_tag được mở trong file cấu hình PHP. Mỗi dòng lệnh trong PHP được kết thúc bằng dấu chấm phẩy (;).

Có 3 cách chú thích (comment) trong PHP:

- Cách 1: // comment cho 1 dòng.
- Cách 2: # comment cho 1 dòng.
- Cách 3: /* comment cho nhiều dòng */.

Ta đề cập đến các kiểu dữ liệu thường dùng trong PHP:

- Boolean: là kiểu dữ liệu cơ bản nhất bao gồm 2 giá trị TRUE (đúng) và FALSE (sai). Ví dụ:

```
<?php
$var = TRUE; //gán giá trị TRUE cho biến $var
?>
```

- Integer: : dữ liệu số nguyên bao gồm các giá trị {-2147483648...-2,-1,0,1,2,...2147483647}, chú ý: 2147483648 (>2147483647) sẽ được hiểu là số thực float. Ví dụ:

```
<?php
$a = 1234;
$a = -123;
$a = 0123; // số bát phân (= 83 hệ thập phân)
$a = 0x1A; // số thập lục phân (= 26 hệ thập phân)
?>
```

- Float: kiểu số thực. Ví dụ:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

- String: kiểu chuỗi, là 1 dãy các ký tự liên tiếp và được giới hạn bởi ' (single quote) hoặc " (double quote) hoặc heredoc. Ví dụ:

```
//single quote:
<?php
$url = 'phpbasic.com';
print 'this is a simple string'; //kết quả: this is a simple string
print 'double quote " '; //kết quả: double quote "
print 'single quote ' '; //kết quả: single quote '
print 'value: $url '; //kết quả: value $url
?>

//double quote:
<?php
$url = 'phpbasic.com';
print 'this is a simple string'; //kết quả: this is a simple string
print "double quote " "; //kết quả: double quote "
print "single quote ' "; //kết quả: single quote '
print "value: $url "; //kết quả: value: phpbasic.com
?>
```


- Array: mảng, ta có thể xem mảng giống như 1 tấm bản đồ thế giới, căn cứ vào tọa độ (key) để xác định quốc gia (value). Cấu trúc array([key =>] value, kiểu dữ liệu của key có thể là Integer hoặc String, kiểu dữ liệu của value có thể là bất kỳ. Ví dụ:

```
//Ví dụ mảng 1 chiều:
<?php
$arr = array("foo" => "bar", 12 => true);
print $arr["foo"]; // bar
print $arr[12]; // 1
?>

//Ví dụ: mảng nhiều chiều(2 chiều)
<?php
$arr = array("somearray" => array(6 => 5, 13 => 9, "a" => 42));
print $arr["somearray"][6]; // 5
print $arr["somearray"][13]; // 9
print $arr["somearray"]["a"]; // 42
?>
```

- Ép kiểu, dùng để ép 1 kiểu dữ liệu sang 1 kiểu khác. Ví dụ:

```
<?php
$a = "5 start"; // string
$b = (int) '5 start'; //integer, $b = 5
$c = 25/7; //float $c = 3.5714285714286
$d = (int) (25/7); // int $d = 3
?>
```

Chú ý PHP là 1 ngôn ngữ thông minh, tự động nhận ra kiểu dữ liệu đang sử dụng và có thể thay đổi khi giá trị của biến thay đổi, đó cũng là lý do vì sao trong PHP không cần khai báo kiểu dữ liệu khi khai báo biến. Ví dụ:

```
<?php

$foo = "0"; // $foo is string (ASCII 48)

$foo += 2; // $foo is now an integer (2)

$foo = $foo + 1.3; // $foo is now a float (3.3)
```

```
$foo = 5 + "10 Little Piggies"; // $foo is integer (15)  
$foo = 5 + "10 Small Pigs"; // $foo is integer (15)  
$a = 'car'; // $a is a string  
$a[0] = 'b'; // $a is still a string  
echo $a; // bar ?>
```

Khi lập trình với ngôn ngữ PHP có 2 phương thức truyền/nhận dữ liệu cần chú ý đó là GET/POST. Tuy nhiên ta cần phân biệt rõ 2 phương thức này để có cách xử dụng hợp lý. Về phương thức GET và POST giống nhau là đều gửi dữ liệu tới server để xử lý, sau khi người dùng nhập thông tin vào Form.

Về sự khác nhau:

- POST: Bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL.
GET: Dữ liệu được gửi tường minh, chúng ta có thể nhìn thấy trên URL, đây là lý do khiến nó không bảo mật so với POST. Nó còn bị giới hạn số ký tự bởi URL của web browsers.
- GET thực thi nhanh hơn POST vì những dữ liệu gửi đi luôn được Webbrowser cached lại.
Khi dùng phương thức POST thì server luôn thực thi và trả về kết quả cho client, còn phương thức GET ứng với cùng 1 yêu cầu đó Webbrowser sẽ xem. Trong cached có kết quả tương ứng với yêu cầu đó không và trả về ngay không cần phải thực thi các yêu cầu đó ở phía server.
- Đối với những dữ liệu luôn được thay đổi thì chúng ta nên sử dụng phương thức POST, còn dữ liệu ít thay đổi chúng ta dùng phương thức GET để truy xuất và xử lý nhanh hơn.

Ví dụ về phương thức GET/POST trong PHP:

- Phương thức GET. Trong ví dụ này sẽ hướng dẫn các ta truyền và nhận dữ liệu qua phương thức GET của Form. Để thực hiện ví dụ này chúng ta cần có 2 site: test.php và test2.php. Trang test.php có nhiệm vụ truyền dữ liệu sang trang test2.php thông qua phương thức GET.

Ví dụ: Trang test.php có nội dung như sau:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<form name="frm_testphp" action="test2.php" method="get">
<input type="text" name="txt_name"/><br/>
<input type="text" name="txt_tuoi"/><br/>
<input type="submit" value="OK" name="OK"/>
</form>
```

Trang test2.php có nội dung như sau:

```
<?php
$name=$_GET['txt_name'];
$tstoi=$_GET['txt_tuoi'];
?>
Name: <?php echo $name; ?><br/>
Tuổi: <?php echo $tstoi; ?>
```

- Phương thức POST. Trong ví dụ này sẽ hướng dẫn các ta truyền và nhận dữ liệu qua phương thức POST của Form. Để thực hiện ví dụ này chúng ta sử dụng 2 site: test.php và test2.php. Site test.php có nhiệm vụ truyền dữ liệu sang trang test2.php thông qua phương thức POST.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<form name="frm_testphp" action="test2.php" method="post">
<input type="text"
name="txt_name"/><br/>
<input type="text" name="txt_tuoi"/><br/>
```

```

<input type="submit"
value="OK" name="OK"/>

</form>

Trang test2.php có nội dung như sau:

<?php $name=$_POST['txt_name'];

$tuoi=$_POST['txt_tuoi'];

?>

Name: <?php echo $name; ?><br/>

Tuổi: <?php echo $tuoi; ?>

```

Về cú pháp các câu lệnh điều khiển, câu lệnh của PHP cũng tương tự như Java và C bao gồm các lệnh cơ bản như IF-ELSE, SWITCH-CASE, WHILE, DO-WHILE... các cú pháp vòng lặp như FOR, FOREACH... các cấu trúc lệnh điều khiển cụ thể sẽ không được trình bày ở đây.

Ngoài các câu lệnh cú pháp điều khiển ta còn phải thường xuyên kết hợp với các hàm xử lý file khi lập trình PHP. Các hàm xử lý file thường dùng đó là mở, đọc, ghi và đóng file. Ngoài ra ta còn có các hàm nâng cao đó là kiểm tra file có tồn tại không, kiểm tra file có được cấp quyền ghi không, lấy nội dung một file mà không cần dùng hàm fread, ghi nội dung file mà không cần dùng hàm fwrite, đổi tên file, copy file, xóa file, kiểm tra một đường dẫn folder có tồn tại không, tạo một folder mới... sẽ không được trình bày cụ thể ở đây.

Về các hàm xử lý file thường dùng ta sẽ tìm hiểu cách viết cũng như cú pháp qua từng ví dụ cụ thể:

- Hàm mở file: Để mở một file ta dùng cú pháp sau: fopen(\$path, \$option). Trong đó \$path là đường dẫn đến file cần mở, \$option là quyền cho phép thao tác trên file. Ta có danh sách các quyền được trình bày như bảng 2.

```

//Ví dụ mở file a.php với quyền thao tác là đọc.

$myfile2 = fopen("a.php", "r") or die("Xảy ra lỗi khi mở file");

```

Bảng 2.4. Danh sách các quyền trong PHP.

Mode	Chú thích
r	Read only
r+	Read + Write
w+	Write only
w	Write + Read. Nếu file này tồn tại thì nội dung cũ sẽ bị xóa đi và ghi lại nội dung mới, còn nếu file chưa tồn tại thì nó tạo file mới.
a	Mở dưới dạng append dữ liệu, chỉ có write và nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới.
a+	Mở dưới dạng append dữ liệu, bao gồm write và read. Nếu file tồn tại nó sẽ ghi tiếp nội dung phía dưới, ngược lại nếu file không tồn tại nó tạo file mới.
b	Mở dưới dạng chế độ binary.

- Hàm đọc file. Có 3 cách đọc file thông thường trong PHP đó là đọc từng dòng, đọc từng ký tự và đọc hết file.
 - Ta dùng hàm fgetc(\$fp) để đọc theo từng ký tự, dùng fgets(\$fp) để đọc theo từng dòng.
 - Đối với đọc từng dòng và đọc từng ký tự ta phải dùng hàm feof(\$fp) đặt trong vòng lặp while để sau khi đọc xong nó sẽ chuyển sang dòng mới hoặc ký tự mới.
 - Để đọc hết tất cả file ta dùng hàm fread(\$fp, \$size), trong đó \$fp là đối tượng lúc mở file còn \$size là kích cỡ của file cần đọc. Để lấy kích cỡ của file cần đọc ta dùng hàm filesize(\$path).

```
//Ví dụ mở và đọc file a.php
$myfile2 = fopen("a.php", "r") or die("Xảy ra lỗi khi mở file");
$x2 = fread($myfile2, filesize("a.php"));
```

- Hàm ghi file. Để ghi file thì bắt buộc file của ta phải được mở ở chế độ mode có cho phép ghi file và tiếp đó dùng hàm fwrite để ghi dữ liệu. Việc ghi file

phụ thuộc vào lúc ta mở file như thế nào. Ví dụ lúc ta mở file ghi đè thì lúc ghi file nó sẽ ghi đè, lúc ta mở file ghi kiểu append thì lúc ghi file nó sẽ thêm xuống cuối file, nếu ta mở file chỉ cho đọc thì ta không thể ghi file được.

```
//Ví dụ mở file a.php và ghi chuỗi abcdef vào file.  
  
$file = fopen("a.php", "w") or die("can't open file");  
  
fwrite($file, "abcdef");
```

- Hàm đóng file. Việc mở file để sử dụng mà không đóng file rất nguy hiểm, vì thế sau khi sử dụng xong ta nên đóng file để an toàn hơn. Để đóng file ta dùng hàm fclose(\$fp) trong đó \$fp là đối tượng trả về lúc ta mở file.

```
//Ví dụ mở - đọc và đóng file den.php.  
  
$myfile = fopen("den.php", "r") or die("can't open file ");  
  
$x = fread($myfile, filesize("den.php"));  
  
echo $x;  
  
fclose($myfile);  
  
//Ví dụ mở - ghi và đóng file den.php.  
  
$file = fopen("den.php", "w") or die("can't open file");  
  
fwrite($file, "tat den");  
  
fclose($file);
```

2.4. Tổng quan về thư viện xử lý hình ảnh

Ngày nay có rất nhiều thư viện hỗ trợ xử lý hình ảnh trên nền tảng các ngôn ngữ phổ biến Visual C++, C#,... như là OpenCV, EmguCV.

OpenCV(Open Source Computer Vision) là thư viện hỗ trợ người làm việc xử lý các hình ảnh trên nền tảng ngôn ngữ C++, C, Java, Python, Windows, Linux,... Được viết tối ưu hóa C/C++.

EmguCV là một cross platform .NET, một thư viện xử lý hình ảnh mạnh dành riêng cho ngôn ngữ C#. Thích hợp các ngôn ngữ như C#, Python, VB,... Warpper có thể được dịch bởi Visual Studio, Xamarin Studio và Unity.

Theo như nhóm nghiên cứu và tìm hiểu được thì EmguCv có một số lợi thế như sau:

- EmguCV được viết hoàn toàn bằng C#. Có thể chạy trên bất kỳ nền tảng hỗ trợ bao gồm iOS, Android, Windows Phone, Hệ điều hành Mac OS X và Linux.
- EmguCV có thể được sử dụng từ nhiều ngôn ngữ khác nhau, bao gồm C#, VB.NET, C ++ và Iron Python.
- Nhận dạng ảnh: nhận dạng khuôn mặt, các vật thể ...
- Xử lý ảnh: khử nhiễu, điều chỉnh độ sáng ...
- Nhận dạng cử chỉ.
- Hỗ trợ tài liệu XML và intellisense.
- Sự lựa chọn để sử dụng hình ảnh lớp hoặc trực tiếp gọi chức năng từ OpenCV.
-

Trong đề tài này, nhóm đã xây dựng phần mềm quản lý cũng như điều khiển hoàn toàn trên ngôn ngữ C#, để sử dụng được OpenCV thì nhóm cần phải tốn khá nhiều thời gian để tạo Warpper kết nối giao tiếp giữa ngôn ngữ C++ của OpenCV với C# trong khi đó sử dụng thư viện EmguCV thì hoàn toàn không cần thiết.

Chính vì thế bộ thư viện EmguCV đã được nhóm lựa chọn để nghiên cứu và sử dụng cho đề tài này.

2.5. Tổng quan về cơ sở dữ liệu (Database)

2.5.1. SQL Server

Cơ sở dữ liệu là một tập hợp thông tin có cấu trúc hay còn được hiểu là một tập hợp liên kết dữ liệu.

Bảng 2.5. Ưu & Nhược điểm của CSDL

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> - Giảm sự trùng lặp thông tin - Đảm bảo dữ liệu có thể truy xuất được theo nhiều cách. - Nhiều người có thể sử dụng một CSDL. 	<ul style="list-style-type: none"> - Phải có cơ chế bảo mật do cơ sở dữ liệu có thể được sử dụng bởi nhiều người. - Cần có cơ chế ưu tiên khi truy cập do tính tranh chấp dữ liệu. - Khả năng rủi ro mất dữ liệu cao.

Cơ sở dữ liệu được phân chia ra nhiều loại khác nhau:

- Cơ sở dữ liệu dạng file: dữ liệu được lưu trữ dưới dạng các file có thể là text, ascii, *.dbf. Tiêu biểu cho cơ sở dữ liệu dạng file là *.mdb Foxpro
- Cơ sở dữ liệu quan hệ: dữ liệu được lưu trữ trong các bảng dữ liệu gọi là các thực thể, giữa các thực thể này có mối liên hệ với nhau gọi là các quan hệ, mỗi quan hệ có các thuộc tính, trong đó có một thuộc tính là khóa chính. Các hệ quản trị hỗ trợ cơ sở dữ liệu quan hệ như: MS SQL server, Oracle, MySQL...
- Cơ sở dữ liệu hướng đối tượng: dữ liệu cũng được lưu trữ trong các bảng dữ liệu nhưng các bảng có bổ sung thêm các tính năng hướng đối tượng như lưu trữ thêm các hành vi, nhằm thể hiện hành vi của đối tượng. Mỗi bảng xem như một lớp dữ liệu, một dòng dữ liệu trong bảng là một đối tượng. Các hệ quản trị có hỗ trợ cơ sở dữ liệu hướng đối tượng như: MS SQL server, Oracle, Postgres
- Cơ sở dữ liệu bán cấu trúc: dữ liệu được lưu dưới dạng XML, với định dạng này thông tin mô tả về đối tượng thể hiện trong các tag. Đây là cơ sở dữ liệu có nhiều ưu điểm do lưu trữ được hầu hết các loại dữ liệu khác nhau nên cơ sở dữ liệu bán cấu trúc là hướng mới trong nghiên cứu và ứng dụng.
- Cơ sở dữ liệu phân cấp (blockchain): Dữ liệu được phân tán trên mạng máy tính ngang hàng và luôn được cả mạng lưới kiểm định. Ví dụ: Bitcoin blockchain.

2.5.2. MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên ta có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL được sử dụng cho việc hỗ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,...

Các đặc trưng cơ bản của MySql được nhóm tìm hiểu như sau:

1. Tốc độ: MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà ta có thể có.
2. Dễ sử dụng: MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn .
3. Giá thành: MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.

4. Hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Ta cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity - một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).
5. Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Ta có thể truy cập MySQL tương tác với sử dụng một vài giao diện để ta có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web...
6. Kết nối và bảo mật: MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó ta có thể chia sẻ dữ liệu của ta với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của ta thì không thể nhìn được.
7. Tính linh động: MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.
8. Sự phân phối rộng: MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của ta. Nếu ta không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, ta có thể lấy mã nguồn và tìm tòi nó. Nếu ta không thích một vài cái, ta có thể thay đổi nó.
9. Sự hỗ trợ: Ta có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

Trong đề tài này, nhóm có thiết kế một trang web để khách hàng đặt chỗ chính vì thế cần một CSDL để lưu thông tin khách hàng và MySql là lựa chọn thích hợp.

2.6. Tổng quan về thuật toán xử lý nhận diện ảnh

2.6.1. Thuật toán OCR

OCR là thuật toán nhận dạng ký tự bằng quang học được tạo ra để chuyển các hình ảnh của chữ viết tay hoặc chữ đánh máy (thường được quét bằng máy scanner) thành các văn bản tài liệu.

Nhận dạng ký tự quang học (dùng các kỹ thuật quang học chẳng hạn như gương và ống kính) và nhận dạng ký tự số (sử dụng máy quét và các thuật toán máy tính) lúc đầu được xem xét như hai lĩnh vực khác nhau. Bởi vì chỉ có rất ít các ứng dụng tồn tại với các kỹ thuật quang học thực sự, bởi vậy thuật ngữ Nhận dạng ký tự quang học được mở rộng và bao gồm luôn ý nghĩa nhận dạng ký tự số.

Một số hạn chế của OCR nhóm đã tìm hiểu được:

- Hầu hết các phần mềm sử dụng OCR chỉ nhận dạng được khoảng 80 - 90% trên hình ảnh rõ nét.
- Đối với những hình ảnh có màu nền mà màu chữ không có nhiều chênh lệch, hay các hình chụp chữ viết tay thì kết quả nhận dạng không khả quan cho lắm
- Đến thời điểm hiện tại thì OCR chưa hỗ trợ tất cả các ngôn ngữ.

2.6.2. Thuật toán Canny

Biên là vấn đề quan trọng trong trích chọn đặc điểm nhằm tìm hiểu ảnh. Cho đến nay thì chưa có định nghĩa chính xác về biên, trong mỗi ứng dụng người ta đưa ra các độ đo khác nhau về biên, một trong số các độ đo đó là sự thay đổi đột ngột về cấp xám. Tập hợp các điểm biên tạo nên biên hay đường bao của đối tượng. Người ta đưa ra 2 phương pháp phát hiện biên cơ bản:

1. Phát hiện trực tiếp: là phương pháp phát hiện biên dựa vào sự thay đổi của mức xám sử dụng các kỹ thuật thay đổi theo hướng.
2. Phát hiện gián tiếp.

Kỹ thuật phát hiện biên bằng phương pháp Canny là phương pháp dò biên trực tiếp rất hiệu quả áp dụng cho các loại ảnh nhiễu.

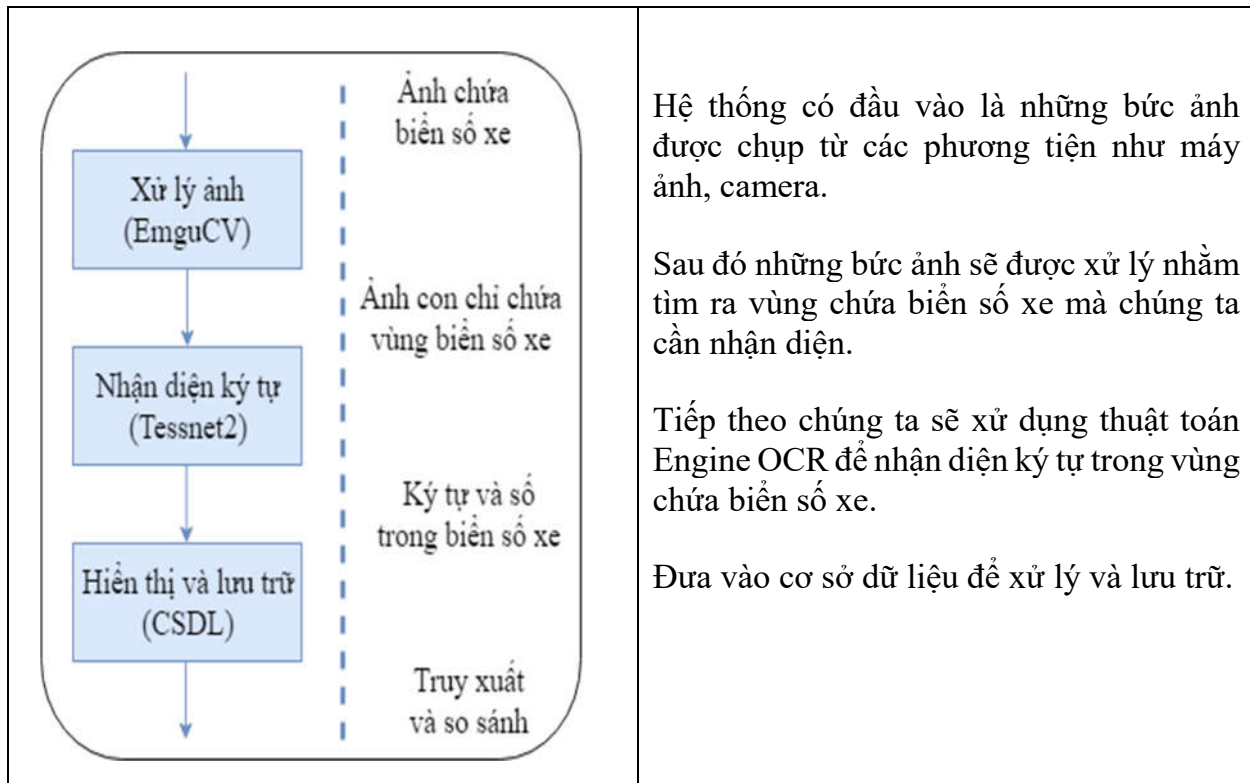
Canny đưa ra ba điểm chính mà một phương pháp phát hiện biên phải xác định được đó là:

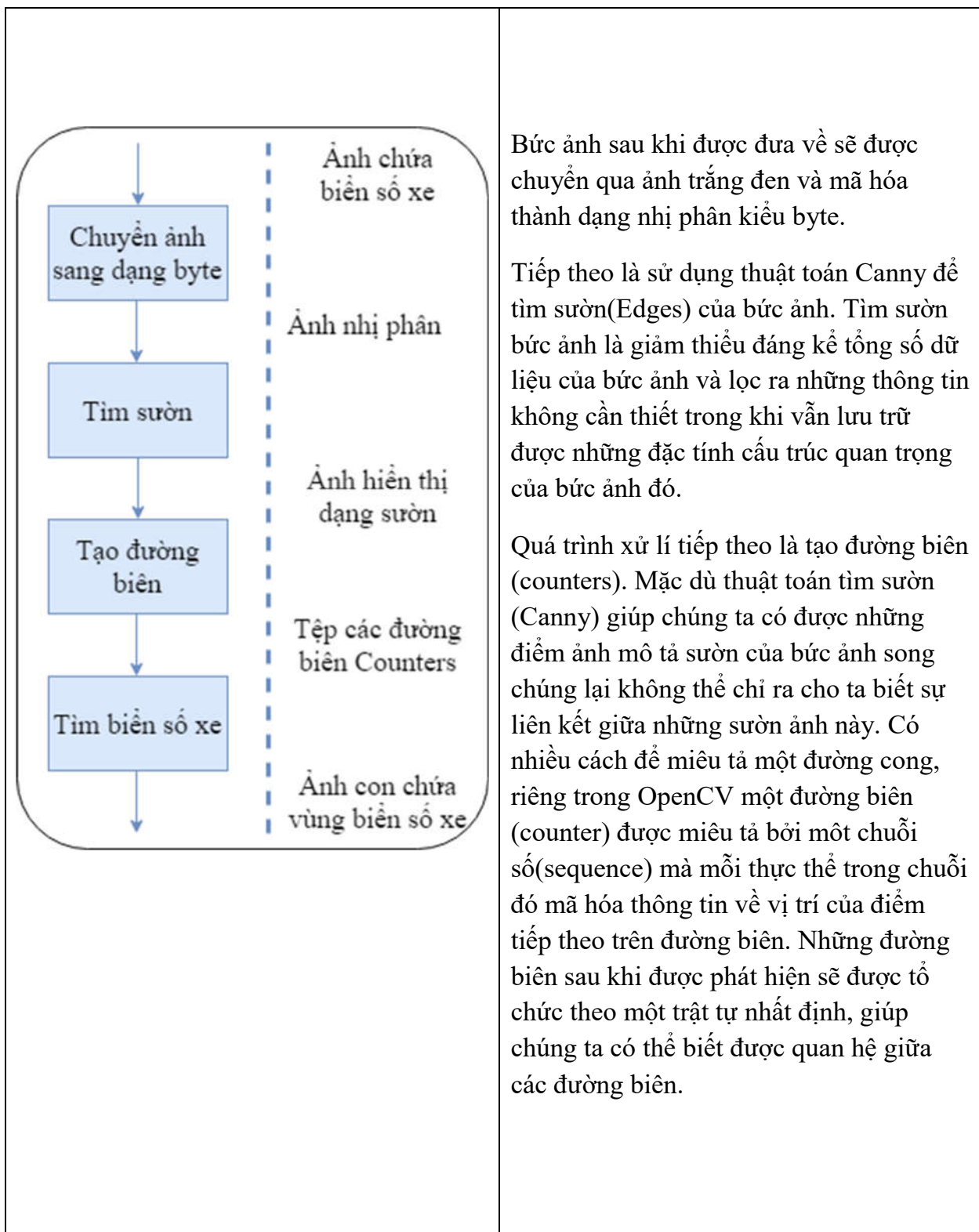
1. Mức lỗi: Phương pháp phải làm sao chỉ có hiệu quả đối với các điểm biên, phải tìm ra tất cả các biên và không có đường biên nào bị bỏ sót.
 2. Định vị: Khoảng cách giữa các điểm biên được tìm thấy trong giải thuật và biên trong thực tế phải càng nhỏ càng tốt.
 3. Hiệu xuất: Không được phép chỉ ra nhiều biên trong khi chỉ có một biên tồn tại
- Giải thuật phát hiện biên Canny được trình bày như sau:

1. Đọc ảnh I cần xử lý
2. Tạo một mặt nạ G để nhân xoắn với I. Độ lệch tiêu chuẩn của mặt nạ này chính là tham số để tách cạnh.
3. Tạo một mặt nạ cho đạo hàm bậc nhất của Gaussian theo hướng x, y và gọi là G_x , G_y và giá trị vẫn được giữ như ở bước 2.
4. Nhân xoắn ảnh I cùng với G dọc theo các hàng tạo ảnh thành phần x gọi là I_x và theo các cột tạo ra ảnh I_y .
5. Nhân xoắn I_x với G_x để sinh ra I'_x : thành phần x của I được nhân xoắn với đạo hàm của Gaussian, và nhân xoắn I_y với G_y để tạo ra I'_y .

2.6.3. Quy trình xử lý nhận diện ảnh

Bảng 2.6. Quy trình xử lý nhận diện ảnh.



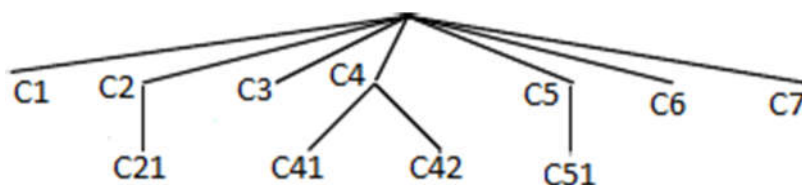


Với yêu cầu ứng dụng đưa ra là tìm biển số xe, ta sẽ xét đến cách tổ chức theo dạng cây của các đường biên.



Hình 2.7. Ví dụ về tổ chức dạng cây của đường biên.

Để dễ hiểu, ta xét một ví dụ về một bức ảnh biển số xe sau khi qua tìm sườn và tạo đường biên. Có 12 đường biên được tạo ra, và chúng sẽ được sắp xếp theo dạng cây với đỉnh là đường biên c0, tiếp theo là 7 nhánh con từ c1 đến c7. Trong nhánh con c2 lại chứa một nhánh con nữa là c21, trong nhánh con c4 chứa 2 nhánh con là c41 và c42, trong nhánh con c5 chứa một nhánh con c51. Mô hình lưu trữ dạng cây^[3] của hình trên được mô tả theo hình vẽ dưới đây:



Hình 2.8. Mô hình lưu trữ Counters dạng cây.

Trong file counters được tạo ra sẽ có các tham số HNext, HPrev, VNext và VPrev để mô tả dạng cây này. Với HNext (Horizontal Next) để chỉ các đường viền đồng mức theo chiều ngang, và VNext (Vertical) để chỉ các đường viền con theo chiều dọc. Để xác định vùng chứa biển số xe, ta sẽ dựa vào tỉ lệ kích thước biển số xe và số kí tự chứa trong biển số xe. Theo quy định về biển số xe ở Việt Nam, biển số xe ô tô gồm biển trước và biển sau không giống nhau, kích thước chiều cao và chiều dài của biển trước là 110 x 470mm, biển sau là: 200 x 280 mm, đối với xe mô tô là 140 x 190 mm. Vậy nếu chỉ xét trường hợp nhận diện biển phía sau thì tỉ lệ vùng chứa biển số phải nằm trong khoảng $1 \div 2$ (~ 1.4). Về số kí tự trong biển số xe là từ 5 đến 8 kí tự bao gồm cả số và chữ. Do đó ta sẽ xác định những đường viền nào có ít nhất 3 nhánh con mới có thể là vùng chứa biển số xe. Kết hợp 2 điều kiện này lại ta sẽ tìm được vùng chứa biển số xe với độ chính xác khá cao.

Chương 3. Thiết kế và xây dựng hệ thống

Sau khi nhóm đã tìm hiểu một số khái niệm liên quan cần thiết, nhóm tiến hành thiết kế và xây dựng phần cứng cũng như phần mềm.

3.1. Giới thiệu và tóm tắt

3.1.1. Yêu cầu chung của hệ thống

Với tiêu chí thiết kế, xây dựng một hệ thống bãi giữ xe thông minh, an toàn, dễ sử dụng. Hệ thống có các tính năng như sau:

- Phần mềm quản lý bãi xe trên PC có chức năng quản lý xe ra/vào sử dụng camera kết hợp với công nghệ RFID và nhận diện biển số xe.
- Đóng mở Barrier khi mã thẻ RFID chính xác
- Cho phép điều chỉnh được số lượng thẻ sử dụng qua phần mềm.
- Cho phép đặt chỗ trước trên Web.
- Hiển thị tình trạng hoạt động của bãi xe.

3.1.2. Phương án thiết kế

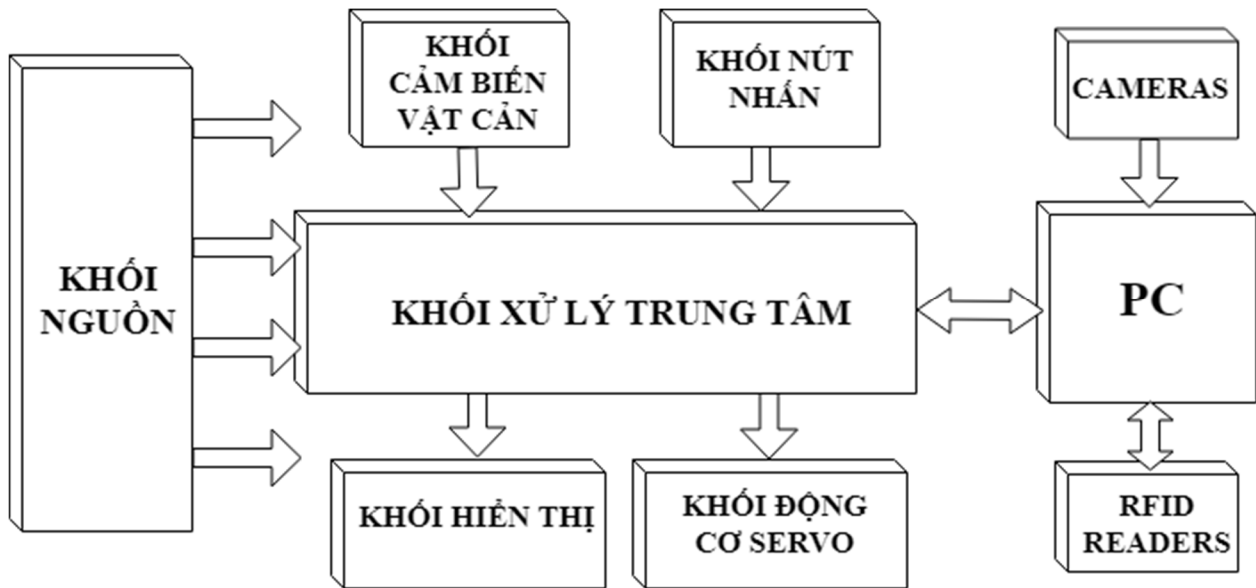
Với các yêu cầu của hệ thống trên nhóm đã tiến hành đề ra các phương án thiết kế cho phần cứng như sau:

- Khối xử lý sẽ tận dụng các module có sẵn ngoài thị trường như: Module Arduino Uno, Arduino Pro Mini, Module RC522, Buzzer, Servo 5V, ...
- Khối nguồn sẽ lấy trực tiếp từ máy tính và Arduino.
- Thiết kế vẽ board mạch kết nối các module lại.

Với các yêu cầu của hệ thống trên nhóm đã tiến hành đề ra các phương án thiết kế cho phần mềm như sau:

- Sử dụng ngôn ngữ C# để thiết kế giao diện điều khiển, quan sát xe vào/ra.
- Sử dụng cơ sở dữ liệu SQL để lưu trữ thông tin thẻ và hình ảnh cũng như xử lý các tác vụ cần thiết.
- Sử dụng ngôn ngữ PHP để thiết kế web đặt chỗ.
- Sử dụng cơ sở dữ liệu MySQL để lưu thông tin khách hàng và vị trí đặt chỗ.
- Sử dụng phần mềm hỗ trợ lập trình Arduino.

3.2. Sơ đồ khối



Hình 3.1. Sơ đồ khối của hệ thống.

- **Khối xử lý trung tâm:** Nhận tín hiệu từ các ngoại vi rồi tiến hành điều khiển hệ thống hoạt động.
- **Khối RFID Readers:** Đọc tín hiệu từ thẻ Tag gửi về cho vi điều khiển.
- **Khối động cơ Servo:** Điều khiển Servo đóng mở để điều khiển xe ra/vào.
- **Khối cảm biến vật cản:** Xác định và tính toán số lượng xe ra/vào.
- **Khối hiển thị:** Hiển thị thông tin trạng thái hoạt động bãi giữ xe.
- **Khối nút nhấn:** Điều khiển servo đóng/mở.
- **Khối nguồn:** Cung cấp nguồn cho toàn mạch của hệ thống.
- **PC:** phần mềm trên PC có chức năng kết hợp với camera và đầu đọc RFID để quản lý dữ liệu xe ra vào, cùng đó là kết hợp với khối xử lý trung tâm để quản lý hệ thống.

Nguyên lý hoạt động chung:

Khi khối RFID nhận tín hiệu từ đầu đọc thẻ tag, sẽ gửi tín hiệu lên vi điều khiển sau đó vi điều khiển sẽ gửi mã đó cho PC để PC điều khiển camera bắt đầu chụp hình. Sau khi chụp hình xong PC lại gửi tín hiệu về vi điều khiển để điều khiển Servo đóng/mở. Sau khi xe tiến vào bãi sẽ có cảm biến thu phát hồng ngoại phát hiện để thông báo tình trạng bãi giữ xe hiện tại. Ngoài ra, nhóm còn thiết kế một giao diện web để khách hàng có thể đặt xe tránh tình trạng khi đến bãi xe không còn chỗ.

3.3. Thiết kế phần cứng

Theo sơ đồ khối ta sẽ thiết kế phần cứng bao gồm 6 phần:

- Khối xử lý trung tâm: Arduino Uno.
- Khối RFID: Module RC522.
- Khối hiển thị: LCD 16x2.
- Khối cảm biến vật cản: Module cảm biến hồng ngoại FC-51.
- Khối động cơ Servo: Động cơ Servo mini 5V.
- Khối nguồn: sử dụng nguồn từ máy tính.

3.3.1. Khối động cơ Servo

- **Yêu cầu thiết kế**

Khối này điều khiển Barrier đóng mở khi có nút nhấn tác động.

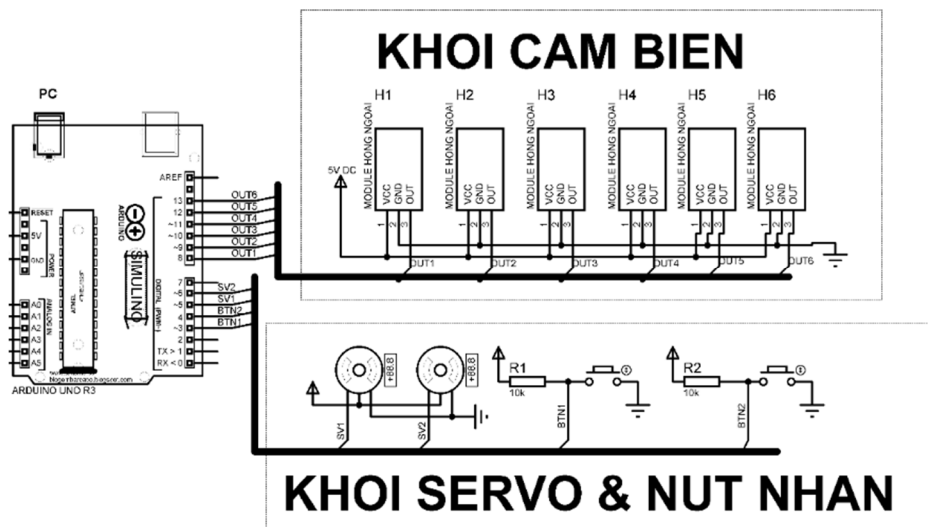
- **Lựa chọn phần cứng**

Đề ứng dụng vào việc đóng mở cửa, đóng mở barrier người ta thường nghĩ đến động cơ servo. Vì do hệ thống nhỏ nên ta sẽ chọn loại động cơ servo 5V mini là đủ đáp ứng cho yêu cầu điều khiển, tối đa giải pháp kinh tế.

- **Kết nối Arduino**

Do servo mini chỉ có 3 chân và đề tài này sử dụng 2 servo nên nguồn, mass và chân điều khiển nên ta tiến hành kết nối chân điều khiển với chân số 5 và chân số 6 của Arduino.

- **Sơ đồ nguyên lý**



Hình 3.2. Sơ đồ kết nối Servo.

- **Giải thích nguyên lý**

Khi có tín hiệu từ Arduino gửi về Servo sẽ điều khiển Barrier đóng/mở.

3.3.2. Khối nút nhấn

- **Yêu cầu thiết kế**

Trong đề tài này, nhóm thiết kế khối nút nhấn để điều khiển servo đóng/mở khi có xe ra/vào.

- **Lựa chọn phần cứng**

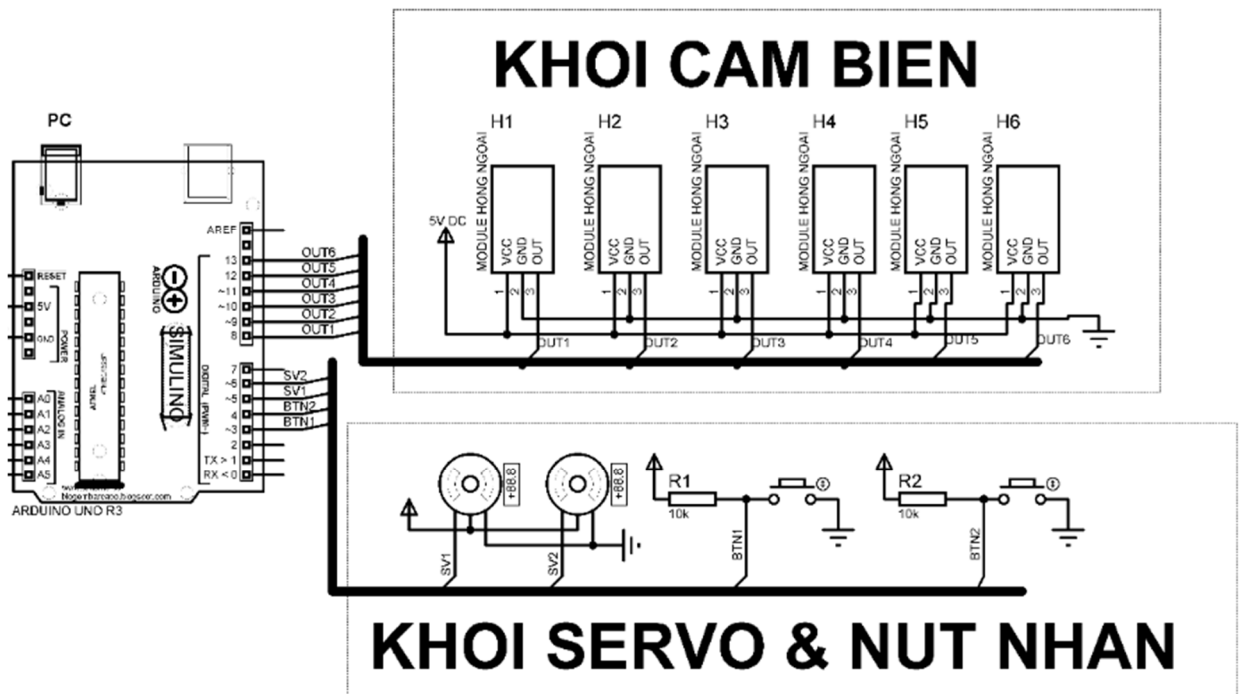
Có rất nhiều loại nút nhấn hiện nay như nút nhấn 2 chân, nút nhấn 4 chân, nút nhấn thường, nút nhấn dán, nút nhấn PLC.

Do yêu cầu đặt ra khá là đơn giản chỉ điều khiển servo đóng/mở và giá cả cũng không chênh lệch bao nhiêu nhưng để cho mô hình thêm phong phú nhóm đã chọn nút nhấn PLC

- **Kết nối**

Ta tiến hành kết nối 2 nút nhấn với chân số 3 và số 4 của adruino.

- **Sơ đồ nguyên lý**



Hình 3.3. Sơ đồ kết nối nút nhấn.

- **Giải thích nguyên lý**

Khi chụp hình thành công, ta nhấn button để mở/đóng servo cho xe vào/ra.

3.3.3. Khối cảm biến vật cản

- **Yêu cầu thiết kế**

Trong đề tài này, nhóm thiết kế khối cảm biến để phát hiện xa ra/vào bãi, đọc vị trí xe cũng như kiểm soát được số lượng xe ra/vào.

- **Lựa chọn phần cứng**

Trong các ứng dụng phát hiện vật thể, phát hiện chuyển động, phát hiện vị trí người ta thường ưu tiên nghĩ đến việc dùng các cảm biến hồng ngoại. Hiện nay trên thị trường có rất nhiều loại cảm biến hồng ngoại ngõ ra analog hoặc digital, với độ chính xác và khả năng chống nhiễu khác nhau tùy thuộc vào mức giá. Bên cạnh cảm biến hồng ngoại, các cảm biến siêu âm cũng được ứng dụng rộng rãi trong việc phát hiện vị trí và dò đường với ưu điểm ít nhiễu và độ chính xác tương đối cao.

Trong đề tài này nhóm chọn sử dụng module Led thu phát hồng ngoại với ngõ ra số vì:

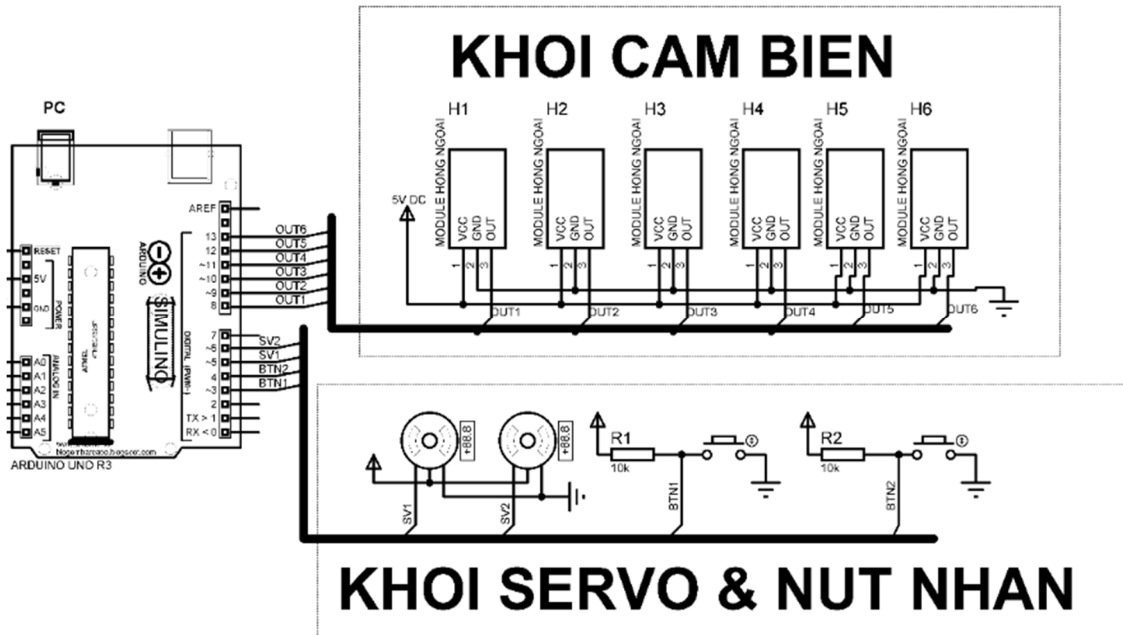
- Thuận tiện cho việc điều khiển.
- Giá thành hợp lý.

Tuy cảm biến này chưa chống nhiễu ánh sáng được thực sự tốt, tuy nhiên vẫn đáp ứng được yêu cầu đặt ra.

- **Kết nối**

Ta tiến hành kết nối 6 cảm biến tới các chân từ chân số 8 tới chân số 13 của arduino.

- **Sơ đồ nguyên lý**



Hình 3.4. Sơ đồ kết nối module hồng ngoại FC-51.

- **Giải thích nguyên lý**

Khối cảm biến được kết nối với Arduino Uno R3 để thông báo tình trạng xe ra vào của bãi trên màn hình LCD.

3.3.4. Khối hiển thị

- **Yêu cầu thiết kế**

Khối hiển thị có chức năng hiển thị thông tin điều khiển giúp cho người điều khiển dễ dàng quan sát.

Trong đề tài này, khối hiển thị được nhóm sử dụng để hiển thị tình trạng của bãi giữ xe đang hoạt động như thế nào cũng như giúp người dùng biết được bãi xe hiện đang còn chỗ hay đã hết.

- **Lựa chọn phần cứng**

Để thiết kế việc hiển thị trong các hệ thống điện tử nhỏ, người ta thường dùng LCD, Led 7 đoạn, Led ma trận. Tuy nhiên Led 7 đoạn và Led ma trận thường chỉ thích hợp cho việc hiển thị số là chính. Ngày nay, thiết bị hiển thị LCD (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của vi điều khiển.

LCD có rất nhiều ưu điểm so với các dạng hiển thị khác:

- Nó có khả năng hiển thị kí tự đa dạng, trực quan (chữ, số và kí tự đồ họa).

- Dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau.
- Tốn rất ít tài nguyên hệ thống và giá thành rẻ.

Từ các ưu điểm đó LCD đã được chọn cho việc hiển thị thông tin điều khiển của hệ thống. Trên thị trường có nhiều loại LCD khác nhau với kích thước và tính năng đa dạng, các LCD thường dùng trong điều khiển như LCD 16x2, 20x4, 128x64. Ở đây hệ thống ta chỉ cần hiển thị ít thông tin, cũng như để thiết kế hệ thống gọn gàng nhỏ đẹp ta chọn LCD 16x2 là đủ cho yêu cầu thiết kế.

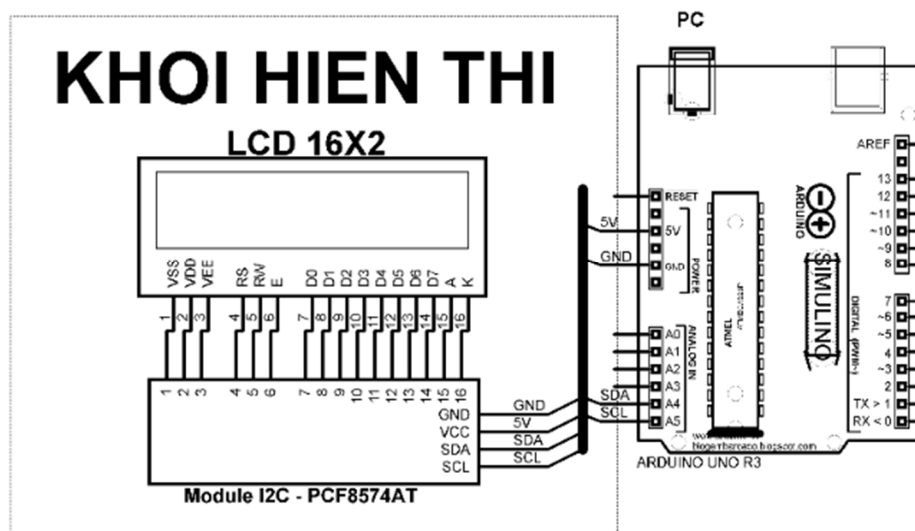
Ở đây ta chọn Arduino giao tiếp với LCD qua module I2C vì:

- Hạn chế được số lượng dây
- Giúp thiết kế phần cứng dễ dàng, gọn gàng.
- **Kết nối**

Bảng 3.1. Bảng kết nối chân giữa LCD và Arduino.

STT	LCD I2C	Arduino
1	VCC	5V
2	GND	GND
3	SDA	Pin A4
4	SCL	Pin A5

- **Sơ đồ nguyên lý**



Hình 3.5. Sơ đồ kết nối LCD.

- **Giải thích nguyên lý**

Khối hiển thị được kết nối với Arduino Uno R3 để hiển thị thông tin tình trạng bãi giữ xe.

3.3.5. Khối RFID

- **Yêu cầu thiết kế**

Khối RFID có nhiệm vụ đọc mã UID từ thẻ RFID sau đó chuyển mã UID thành chuỗi với số Hex rồi gửi về Arduino mang đi điều khiển.

- **Phương án chọn phần cứng**

Như yêu cầu hệ thống phần mềm trên máy tính sẽ kết hợp giữa các camera và công nghệ RFID để quản lý xe ra vào. Ở đây ta sử dụng công nghệ RFID thụ động để ứng dụng cho hệ thống.

Với các ứng dụng RFID người ta sẽ thường chọn các module đầu đọc thẻ có sẵn với các tần số khác nhau như 125Khz, 13.56Mhz, 433Mhz...

- **Lựa chọn**

Ở đây ta chọn đầu đọc thẻ 12.56Mhz RC522 vì khoảng cách không cần xa, vì khoảng cách không cần xa để giao tiếp với Arduino Uno tạo thành đầu đọc thẻ cho hệ thống bãi giữ xe của chúng ta. Bên đó ta sẽ gắn thêm một buzzer, cho đầu đọc, để lập trình buzzer báo khi có thẻ quét vào.

Module RFID RC522 sử dụng IC MFRC522 của Phillip dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56Mhz, 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.

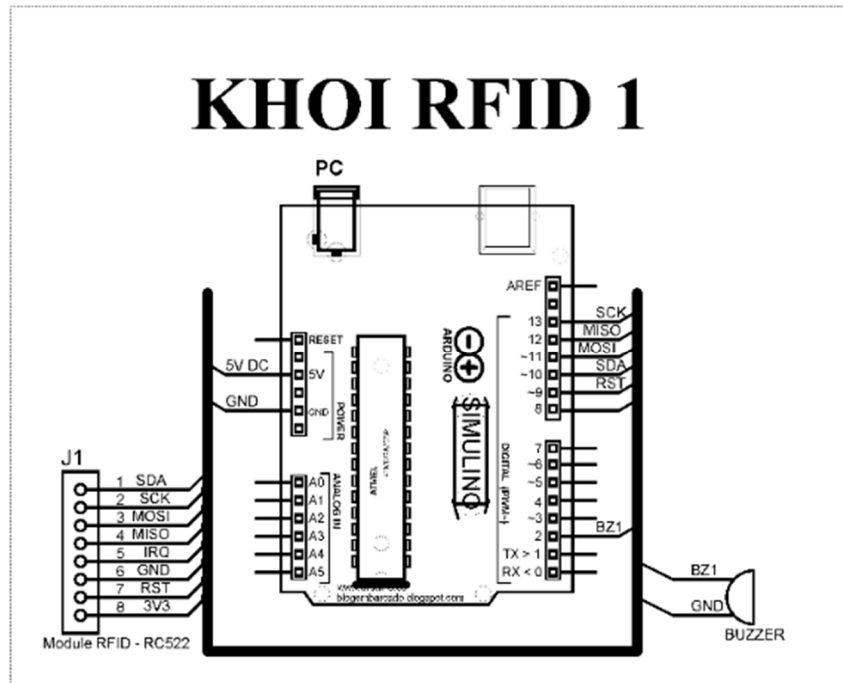
- **Kết nối Arduino**

Để có thể đọc được mã UID từ thẻ tag ta tiến hành kết nối module RC522 với Arduino.

Bảng 3.2. Sơ đồ kết nối RC522 với Arduino

STT	RFID RC522	Arduino
1	3.3V	3.3V
2	RST	Pin 9
3	GND	GND
4	MISO	Pin 11
5	MOSI	Pin 12
6	SCK	Pin 13
7	SDA	Pin 10

- Sơ đồ nguyên lý



Hình 3.6. Sơ đồ kết nối khối RFID.

- Giải thích nguyên lý

Khi có thẻ tag được quét vào hay thẻ mới tác động vào RC522 thì buzzer sẽ kêu để báo hiệu đồng thời sẽ gửi mã thẻ đã quét cho vi xử lý trung tâm.

3.3.6. Khối xử lý trung tâm

- Yêu cầu thiết kế

Khối xử lý trung tâm được lập trình để:

- Giao tiếp với module RC522 để nhận tín hiệu điều khiển chụp hình.
- Giao tiếp với PC để gửi dữ liệu cũng như lưu trữ dữ liệu hình ảnh, mã thẻ.
- Giao tiếp với Servo, nút nhấn để điều khiển đóng/mở servo khi có khách ra/vào.
- Giao tiếp với cảm biến để biết số lượng xe vào/ra.

- Phương án lựa chọn

Để thực hiện được những công việc điều khiển như trên ta có thể sử dụng các loại vi điều khiển như Arduino, PLC hoặc PIC... Tuy nhiên PLC ta có thể thấy chỉ được sử dụng trong các công nghiệp lớn và mắc tiền, còn trong đề tài này chỉ là một hệ thống nhỏ nên ta chỉ cần sử dụng các loại vi điều khiển là đủ. Vừa tiết kiệm chi phí, vừa dễ sử dụng.

- **Lựa chọn**

Trong đề tài này ta sử dụng vi điều khiển AT328 được tích hợp sẵn trong board Arduino Uno R3 thuận tiện cho lập trình và tiết kiệm thời gian vì:

- Arduino hiện là đang là vi điều khiển được phát triển rộng rãi và sử dụng phổ biến. Được hỗ trợ cực kì rộng lớn cho nhiều ứng dụng thực tế.
- Arduino được lập trình dựa trên ngôn ngữ Java mang lại khả năng xử lý chuỗi mạnh mẽ, tạo nên sự thuận lợi cho yêu cầu hệ thống của chúng ta, vì khi giao tiếp với PC ta sẽ dường như làm việc với chuỗi để xử lý đến >90% trong code chương trình.
- Giá thành rẻ phù hợp với sinh viên.

3.3.7. Khối nguồn

- **Yêu cầu thiết kế**

Khối này có nhiệm vụ cung cấp điện cho các vi xử lý hoạt động.

- **Phương án**

Điện áp và dòng điện cung cấp cho các linh kiện hoạt động bình thường được liệt kê theo danh sách trong bảng

Bảng 3.3. Dòng và áp quy định các thiết bị trong đồ án.

STT	Thiết bị	Số lượng	Điện áp	Dòng điện	Tổng dòng điện
1	Arduino Uno R3	3	5V	30mA	90mA
2	Module RC522	2	3.3V	13-26mA	26-52mA
3	Servo Mini	2	5V	550mA	1100mA
4	Buzzer	2	5V	25mA	50mA
5	LCD 16x2	1	5V	25mA	25mA
6	FC-51	6	5V	43mA	258mA
7	Button	2	5V	0.5mA	1mA
Tổng dòng điện					~1.6A

- **Phương hướng thiết kế**

Để Arduino có thể điều khiển qua giao diện C#, cần kết nối trực tiếp Arduino với máy tính, cho nên nguồn sử dụng cho Arduino sẽ là nguồn USB liên kết Arduino với máy tính.

Buzzer hoạt động khi các chân digital điều khiển được kích hoạt mức cao, như vậy nguồn điều khiển buzzer là nguồn trực tiếp từ Arduino.

3.3.8. Sơ đồ nguyên lý toàn mạch

Sơ đồ nguyên lí như trang đính kèm.

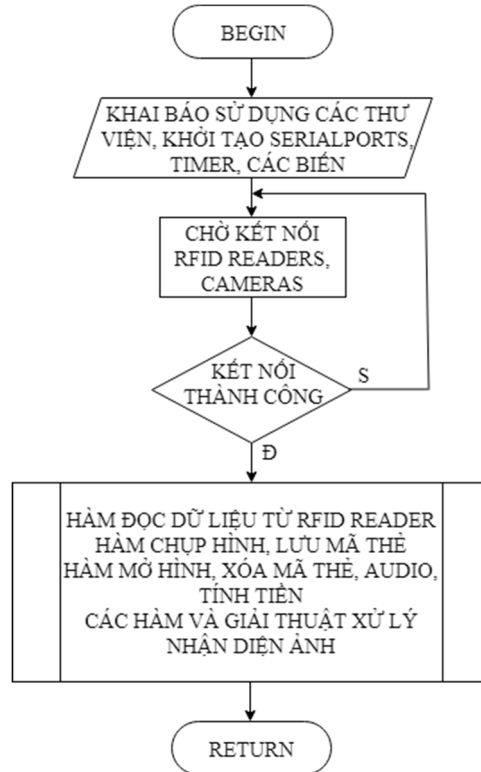
3.4. Thiết kế phần mềm

3.4.1. Thiết kế phần mềm cho PC

Từ yêu cầu của hệ thống ta tiến hành thiết kế giao diện phần mềm để quản lý xe ra/vào như sau:

- Đọc hình ảnh từ 2 camera.
- Sử dụng đầu đọc thẻ RFID để chụp ảnh xe vào cũng như lấy ảnh để so sánh xe ra.
- Có chức năng SET THẺ để, nhận dạng và từ chối thẻ lạ không có trong hệ thống.
- Dữ liệu hệ thống như mã thẻ, hình ảnh được lưu vào cơ sở dữ liệu SQL.
- Dữ liệu đặt chỗ được lưu vào cơ sở dữ liệu MySQL.
- Có thể làm việc ở chế độ FullScreen.

3.4.1.1. Lưu đồ giải thuật chương trình chính



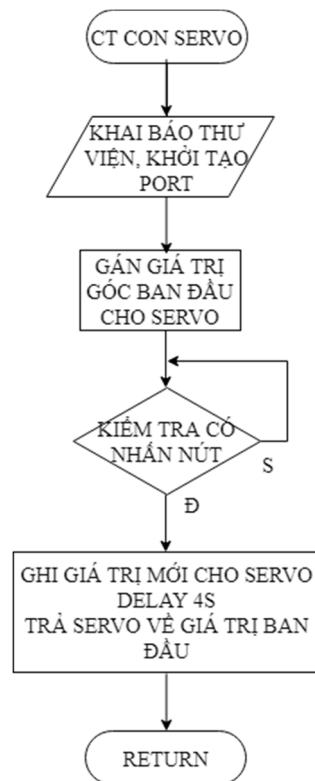
Hình 3.7. Lưu đồ phần mềm quản lý trên PC.

Giải thích lưu đồ:

- Đầu tiên khởi tạo các thư viện hỗ trợ, khởi tạo các cổng kết nối UART, khởi tạo các biến sử dụng trong chương trình.
- Tiếp theo khi phần mềm đã khởi tạo xong chúng ta tiến hành kết nối tới các đầu đọc thẻ, Camera.
- Tiếp theo là tới vòng lặp bao gồm:
 - Đọc dữ liệu từ RFID Reader để lưu hình ảnh xe vào sau đó xử lý ảnh, lấy hình ảnh so sánh xe ra và tính tiền theo hàm đã được lập trình sẵn.
 - Tự động xóa thẻ khi thẻ đó được sử dụng ở ngõ ra.

3.4.1.2. Lưu đồ giải thuật của chương trình con

- Lưu đồ giải thuật cho Servo và nút nhấn

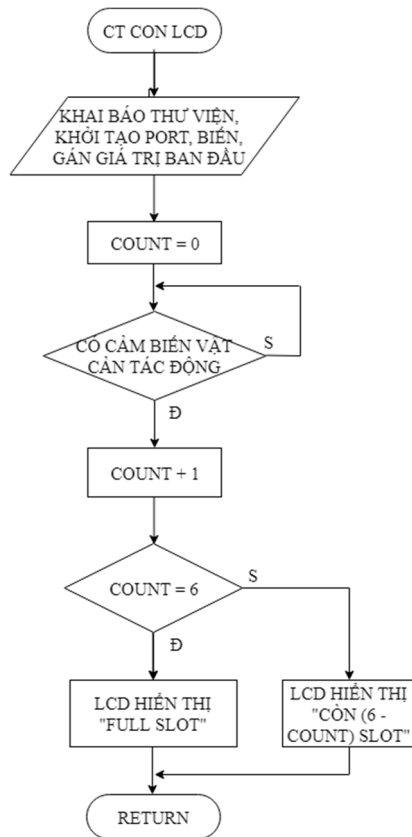


Giải thích lưu đồ:

- Khởi tạo thư viện, khởi tạo Serial port.
- Gán giá trị ban đầu cho servo.
- Kiểm tra có nhấn nút hay không?
- Nếu có thì tiến hành gán giá trị mới cho servo sau đó delay 4s rồi trả về giá trị ban đầu.
- Nếu không thì tiếp tục kiểm tra.

Hình 3.8. Lưu đồ giải thuật servo và nút nhấn

- Lưu đồ giải thuật cho LCD

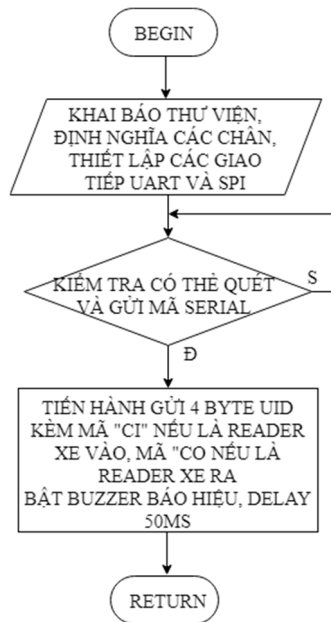


Giải thích lưu đồ:

- Khai báo thư viện, khởi tạo cái biến, gán giá trị ban đầu.
- Kiểm tra có tác động của cảm biến hay không.
- Nếu có thì tăng biến COUNT ngược lại thì tiếp tục kiểm tra.
- Khi có cảm biến tác động, kiểm tra biến COUNT có bằng 6 hay không.
- Nếu có thì hiển thị lên LCD “ Full Slot”, ngược lại thì hiển thị giá trị COUNT đang đếm.

Hình 3.9. Lưu đồ giải thuật LCD

- Lưu đồ giải thuật cho RFID Reader



Giải thích lưu đồ:

- Đầu tiên sẽ khởi tạo các thư viện cần thiết, định nghĩa các chân sử dụng, thiết lập các chuẩn giao tiếp sử dụng như UART, SPI.
- Kiểm tra có quét thẻ hay không. Nếu có thì tiến hành gửi 4 byte UID của thẻ + theo mã “ci” nếu là xe vào và ngược lại gửi 4 byte UID của thẻ + mã “co” nếu là xe ra.
- Bật buzzer báo hiệu khi có thẻ được quét

Hình 3.10. Lưu đồ giải thuật RFID Reader

3.4.2. Thiết kế trang Web đặt chỗ

Nhóm thiết kế trang Web đặt chỗ có chức năng như sau:

- Cho phép khách điền thông tin của mình trước khi book chỗ.
- Hiển thị trạng thái các chỗ đã được book hay chưa được book.
- Hiển thị thông báo khi đặt chỗ thành công cũng như chỗ đã được đặt.

3.4.3. Chương trình cho RFID-Reader

INPUT	OUTPUT
<pre> #include <SPI.h> #include <MFRC522.h> #include <Servo.h> #define RST_PIN 9 #define SS_PIN 10 Servo myservo_vao; int BELL_PIN = 2; int sv; MFRC522 mfrc522(SS_PIN, RST_PIN); void setup () { pinMode (BELL_PIN, OUTPUT); Serial.begin(9600); </pre>	<pre> #include <SPI.h> #include <MFRC522.h> #include <Servo.h> #define RST_PIN 9 #define SS_PIN 10 Servo myservo_ra; int BELL_PIN = 2; int sv; MFRC522 mfrc522(SS_PIN, RST_PIN); void setup () { pinMode (BELL_PIN, OUTPUT); Serial.begin(9600); </pre>

<pre> while (! Serial); SPI.begin(); mfrc522.PCD_Init(); myservo_vao.attach(4); myservo_vao.write(0); } void loop() { if(mfrc522.PICC_IsNewCardPresent()) return; if (! mfrc522.PICC_ReadCardSerial()) return; Serial.print("ci"); Serial.print(mfrc522.uid.uidByte[0], HEX); Serial.print(mfrc522.uid.uidByte[1], HEX); Serial.print(mfrc522.uid.uidByte[2], HEX); Serial.println(mfrc522.uid.uidByte[3] , HEX); digitalWrite(BELL_PIN, HIGH); delay(50); digitalWrite(BELL_PIN, LOW); delay(50); digitalWrite(BELL_PIN, HIGH); delay(50); digitalWrite(BELL_PIN, LOW); delay(1000); while (Serial.available()) { sv = Serial.read(); Serial.println(sv); delay(200); } if (sv == '1') { myservo_vao.write(100); delay(3000); myservo_vao.write(0); } } </pre>	<pre> while (! Serial); SPI.begin(); mfrc522.PCD_Init(); myservo_vao.attach(4); myservo_vao.write(0); } void loop() { if(mfrc522.PICC_IsNewCardPresent()) return; if (! mfrc522.PICC_ReadCardSerial()) return; Serial.print("co"); Serial.print(mfrc522.uid.uidByte[0], HEX); Serial.print(mfrc522.uid.uidByte[1], HEX); Serial.print(mfrc522.uid.uidByte[2], HEX); Serial.println(mfrc522.uid.uidByte[3] , HEX); digitalWrite(BELL_PIN, HIGH); delay(50); digitalWrite(BELL_PIN, LOW); delay(50); digitalWrite(BELL_PIN, HIGH); delay(50); digitalWrite(BELL_PIN, LOW); delay(1000); while (Serial.available()) { sv = Serial.read(); Serial.println(sv); delay(200); } if (sv == '1') { myservo_ra.write(100); delay(3000); myservo_ra.write(0); } } </pre>
--	--

Chương 4. Kết quả thực hiện

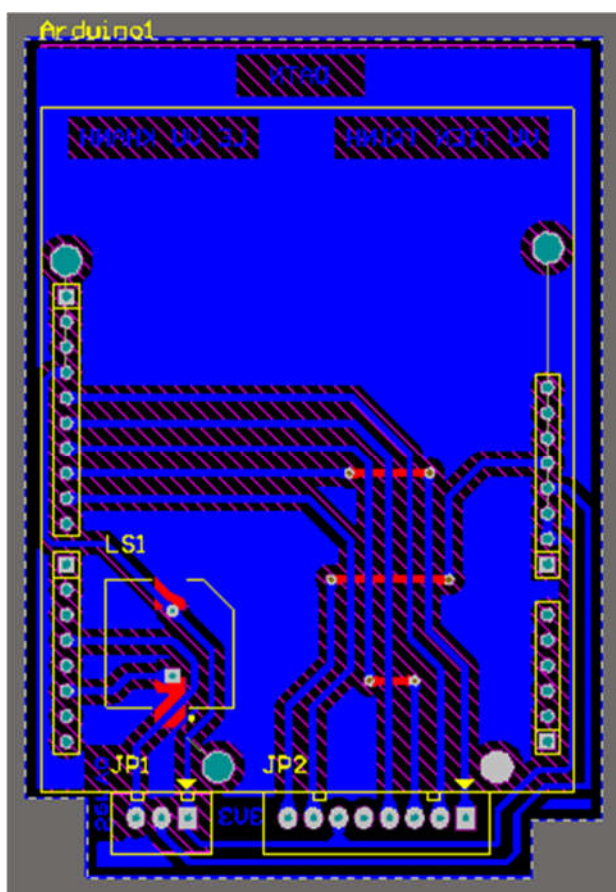
4.1. Phần cứng

4.1.1. Các công cụ sử dụng

- Đồng hồ VOM.
- Chì hàn.
- Chuẩn bị các module có sẵn theo thiết kế.
- Chuẩn bị các linh kiện như sơ đồ nguyên lý.
- Chuẩn bị các phần mềm sử dụng.

4.1.2. Vẽ mạch in mạch đã thiết kế

Sau khi thiết kế sơ đồ khối, sơ đồ nguyên lý toàn mạch ta tiến hành vẽ mạch in đã thiết kế.



Hình 4.1. Mạch in toàn mạch

4.1.3. Board mạch đã hoàn thiện

Sau khi thiết kế sơ đồ nguyên lý phần cứng và tiến hành thi công mạch nay nhóm đã hoàn thành xong sản phẩm

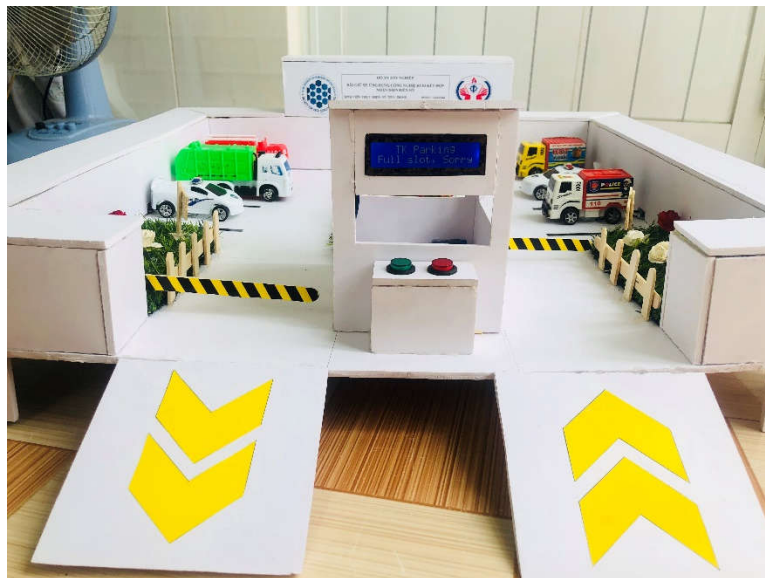


Hình 4.2. Board mạch hoàn thiện.

4.1.4. Thi công mô hình bãi xe

Để thể hiện tính ứng dụng một cách trực quan của hệ thống vào thực tế nhóm đã tiến hành tiến hành thiết kế và thi công bãi giữ xe như thực tế.

Mô hình bãi giữ xe được lắp các barrier đóng, mở và được trang trí để hoạt động như một bãi giữ xe ngoài thực tế.



Hình 4.3. Mô hình bãi giữ xe.

4.2. Phần mềm

4.2.1. Phần mềm quản lý bãi giữ xe trên PC

Theo yêu cầu và phương án thiết kế đã đề ra ta tiến hành viết phần mềm quản lý bãi xe trên PC sử dụng ngôn ngữ lập trình C#.

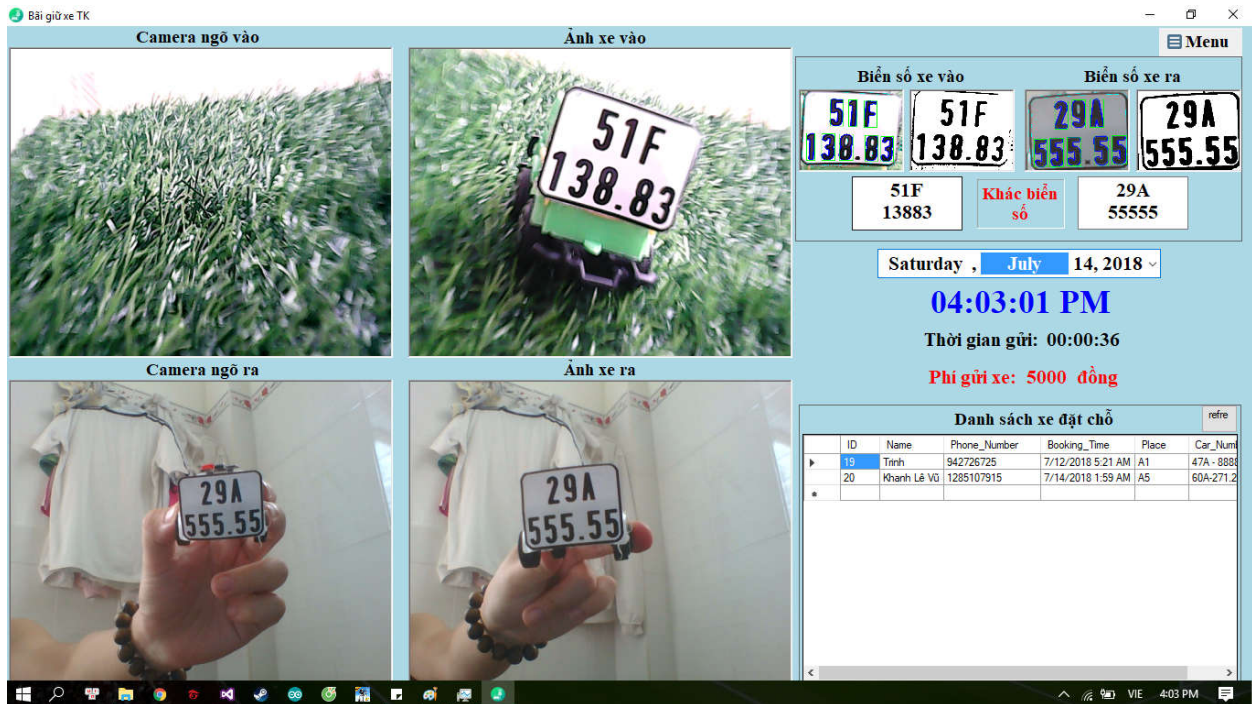
Phần mềm được lập trình có chức năng:

- Giao tiếp với camera ngõ vào/ra của bãi xe.
- Nhận tín hiệu từ 2 đầu đọc thẻ RFID để chụp ảnh xe ngõ vào/ra.
- Lưu hình ảnh vào PC, lưu dữ liệu vào thẻ xe, giờ vào xuống cơ sở dữ liệu SQL Server, lấy hình ảnh từ cơ sở dữ liệu, tính số giờ ra của xe, tính tiền giữ xe tự động.
- Có chức năng SET THẺ cho hệ thống lưu vào cơ sở dữ liệu (chỉ các mã thẻ RFID được lưu vào cơ sở dữ liệu mới hoạt động được trên hệ thống).
- Lấy dữ liệu đặt chỗ từ cơ sở dữ liệu MySql.

Các thư viện hỗ trợ trong quá trình lập trình:

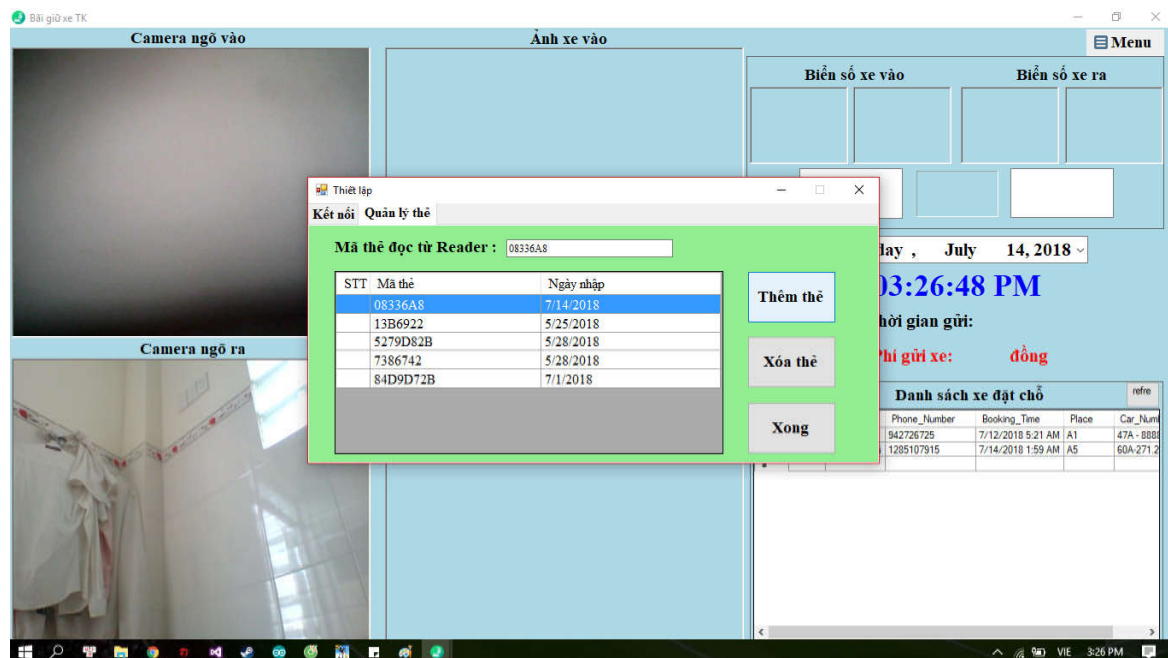
- Thư viện hỗ trợ hình ảnh AForge.NET.
- Thư viện giao tiếp cơ sở dữ liệu SQL-Helper.
- Thư viện EmguCV và OpenCV.
- Thư viện MySql.

Hình 4.4 như mô tả dưới đây là giao diện chính của phần mềm nhóm đã thiết kế



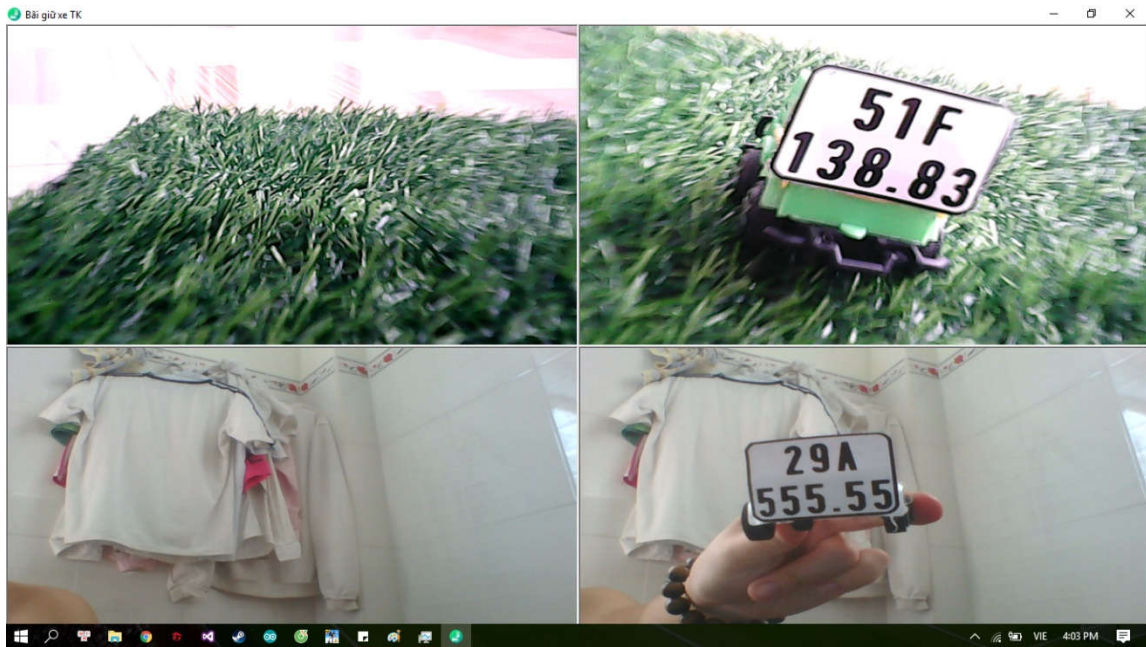
Hình 4.4. Giao diện chính phần mềm quản lý bãi xe.

Để tiện cho việc quản lý các thẻ cũng như số lượng thẻ dùng nhóm đã thiết kế thêm giao diện SET THẺ như hình 4.5



Hình 4.5. Giao diện SET THẺ trên phần mềm quản lý bãi xe.

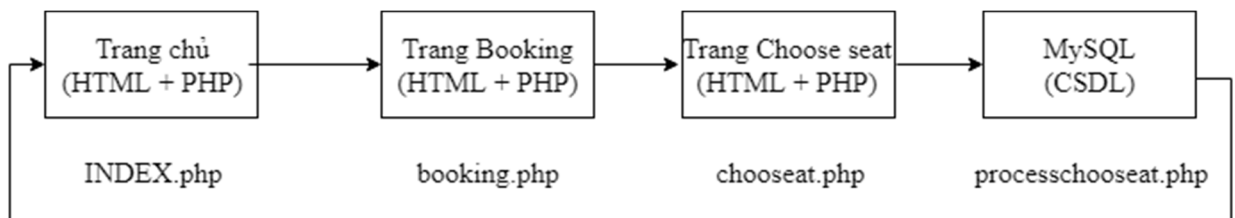
Ngoài ra để trực quan và dễ quan sát chế độ Full Screen cũng được nhóm thiết kế



Hình 4.6. Giao diện phần mềm chế độ Fullscreen.

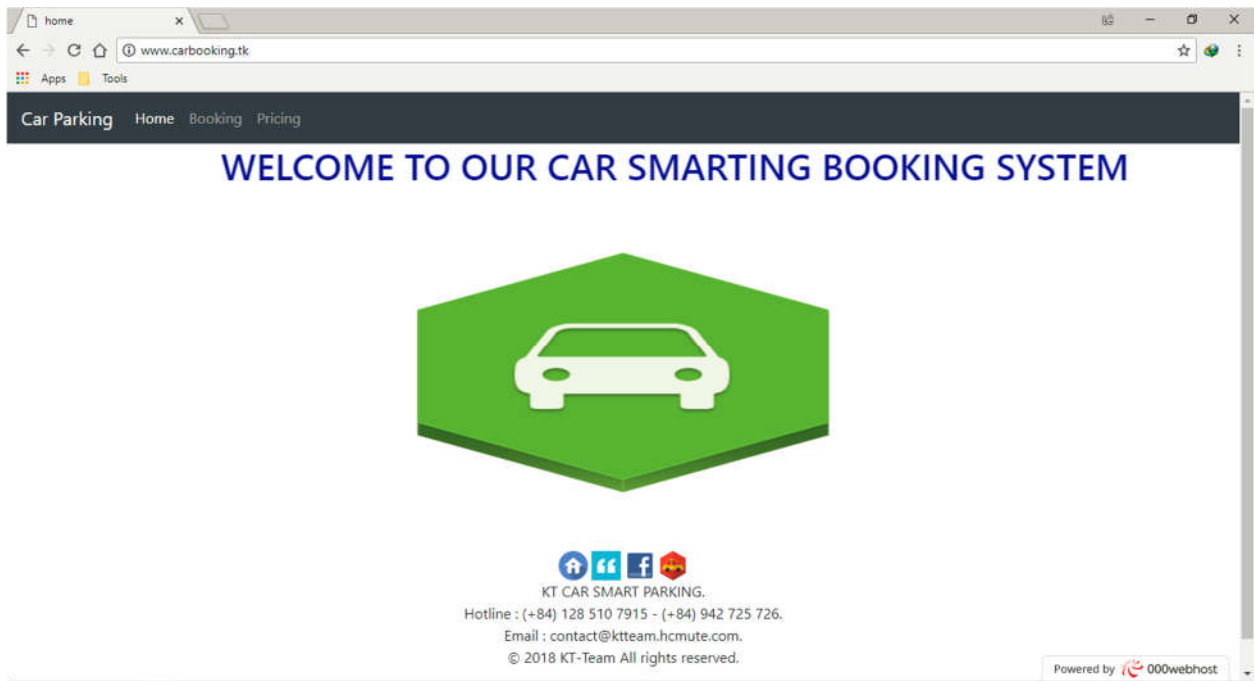
4.2.2. Web đặt chỗ

Theo yêu cầu đã được đặt ra nhóm đã thiết kế Web gồm 2 trang: giới thiệu và đặt chỗ. Từ đó ta thiết kế website có sơ đồ hoạt động như sau:



Hình 4.7. Sơ đồ hoạt động của Web.

Giao diện các trang website sau khi thiết kế như hình sau. Hình 4.8 là giao diện trang web chính



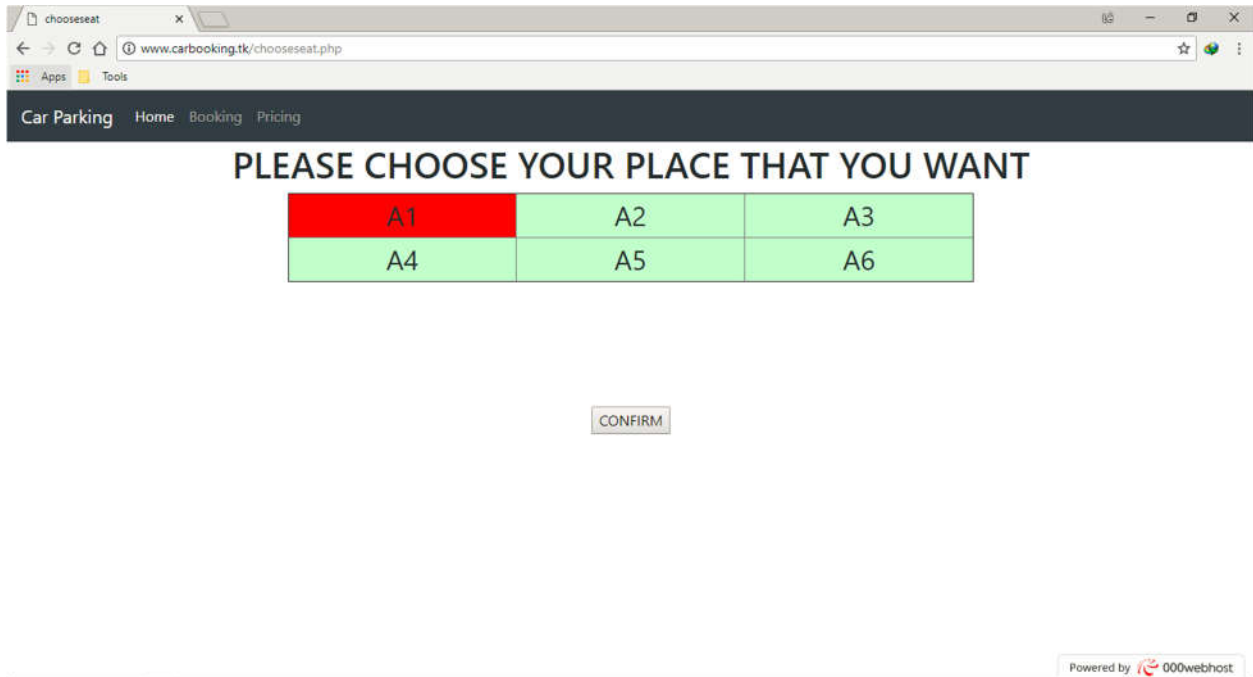
Hình 4.8. Trang chính của Web.

Trước khi tiến hành chọn chỗ book ta cần điền thông tin cần thiết để tên, số điện thoại như hình 4.9 mô tả ở dưới

The screenshot shows a web browser window with the address bar displaying 'www.carbooking.tk/booking.php'. The page has a blue header and a light blue sidebar. The main content area is white and contains a green envelope icon with a checkmark. To the right of the icon, there is a form titled 'Car Booking Form' in red. The form has four input fields: 'Name' with the value 'Khánh Lê Vũ', 'Phone Number' with the value '01285107915', 'Check In' with the value '12/07/2018 08:00 AM', and 'Car Number' with the value '60A-271.23'. Below the input fields is a green 'BOOK' button. At the bottom right, it says 'Powered by 000webhost'.

Hình 4.9. Trang booking.

Sau cùng là trang chọn chỗ như hình 4.10



Hình 4.10. Trang chọn chỗ Chooseat.

Chương 5: Kết quả so sánh, thực nghiệm, phân tích, tổng hợp

Sau khi hoàn thiện ta tiến hành chạy thử toàn bộ hệ thống để kiểm tra độ ổn định, cũng như tính chính xác và độ trễ khi điều khiển. Sau thực nghiệm ta thấy khi mạch hoạt động thời gian dài mạch điều khiển chạy chưa ổn định và dễ bị nhiễu.

Kế tiếp ta tiến hành kiểm tra các chức năng:

- Các chức năng của phần mềm bãi giữ xe RFID và nhận diện ảnh trên PC.
- Đặt chỗ qua Website.
- Vận hành các thiết bị trên hệ thống.

Cụ thể quá trình kiểm tra hoạt động của phần mềm và các chức năng còn lại có thể xem cụ thể ở phần Video Demo theo đường link:

- Phần mềm quản lý bãi xe:
<https://www.youtube.com/watch?v=Ipz5zeLOk3U>

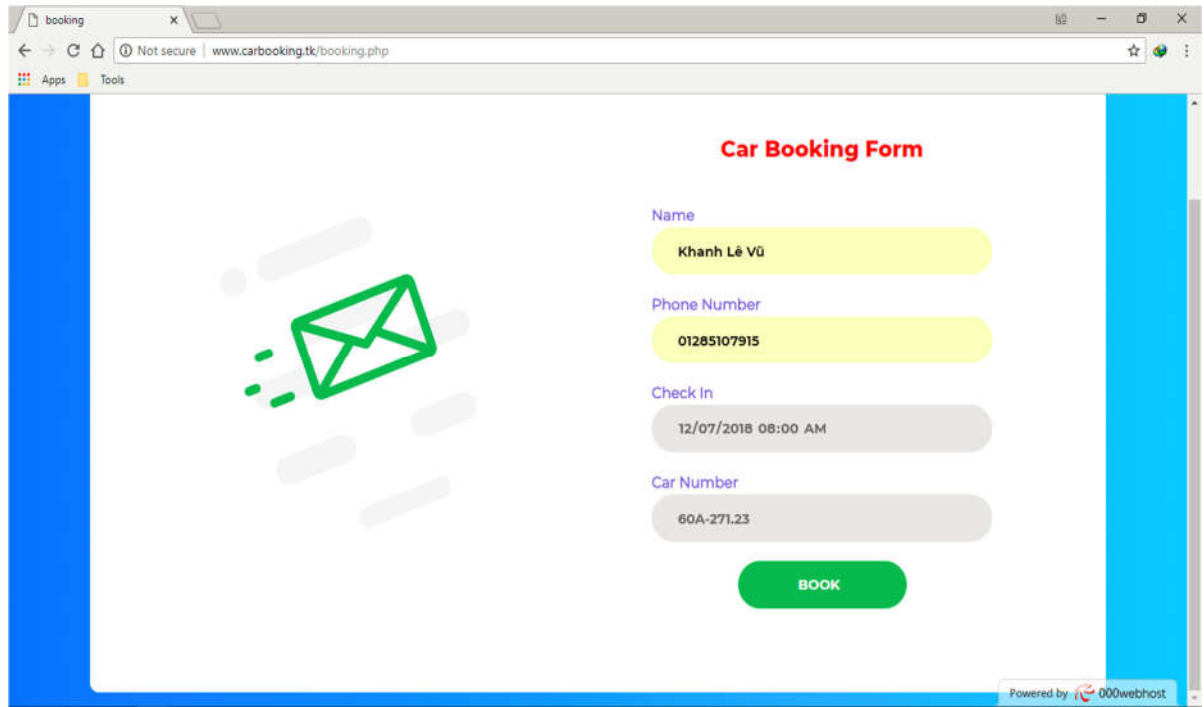
Ở đây sẽ chỉ trình bày lại bằng tóm tắt cũng như một vài hình ảnh của quá trình kiểm tra. Quá trình kiểm tra sẽ tiến thành theo trình tự thực hiện việc khởi động lại hệ thống, kiểm tra các chức năng đặt chỗ. Ta có thể qua sát thông tin hiển thị trên LCD.

Sau khi khởi động mạch điều khiển và phần mềm trên PC, ta tiến hành kết nối camera, đầu đọc thẻ RFID trên phần mềm sau đó tiến hành kết nối phần mềm trên PC với mạch điều khiển. Hình ảnh sau khi bước kết nối hoàn tất như hình

Sau khi kiểm tra xong ta tiến hành kiểm tra chức năng đặt chỗ qua Website. Ta tiến hành truy cập vào trang web hệ thống theo địa chỉ: <http://carbooking.tk/> , sau đó ta tiến hành chọn vào trang booking, tiến hành điền thông tin cần thiết. Sau đó hệ thống sẽ tự động chuyển qua trang cho người dùng chọn chỗ. Kết thúc đặt chỗ xe xuất hiện thông báo “Đặt chỗ thành công” và tự động chuyển về trang chủ của website.

Ta tiến hành xét ví dụ về chọn chỗ A5 như hình sau:

Đầu tiên điền các thông tin được yêu cầu trong trang booking như tên, số điện thoại, biển số xe...



Car Booking Form

Name
Khanh Lê Vũ

Phone Number
01285107915

Check In
12/07/2018 08:00 AM

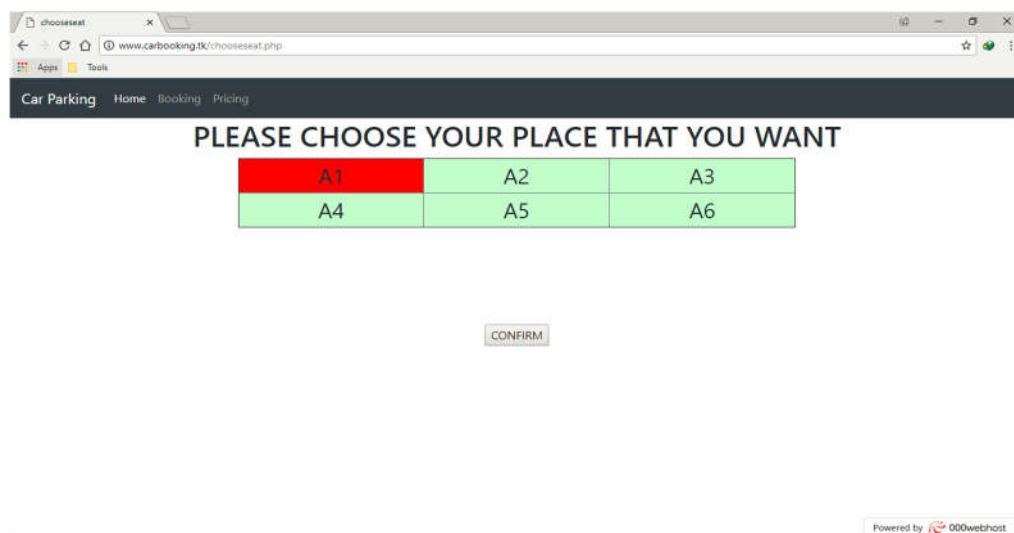
Car Number
60A-271.23

BOOK

Powered by 000webhost

Hình 5.1. Trang điền thông tin.

Sau khi điền thông tin xong tiến hành nhấn nút “book” trên màn hình. Trang web tự động nhảy qua trang chọn chỗ. Màu đỏ là đã được chọn, màu xanh là đang còn trống.



Car Parking Home Booking Pricing

PLEASE CHOOSE YOUR PLACE THAT YOU WANT

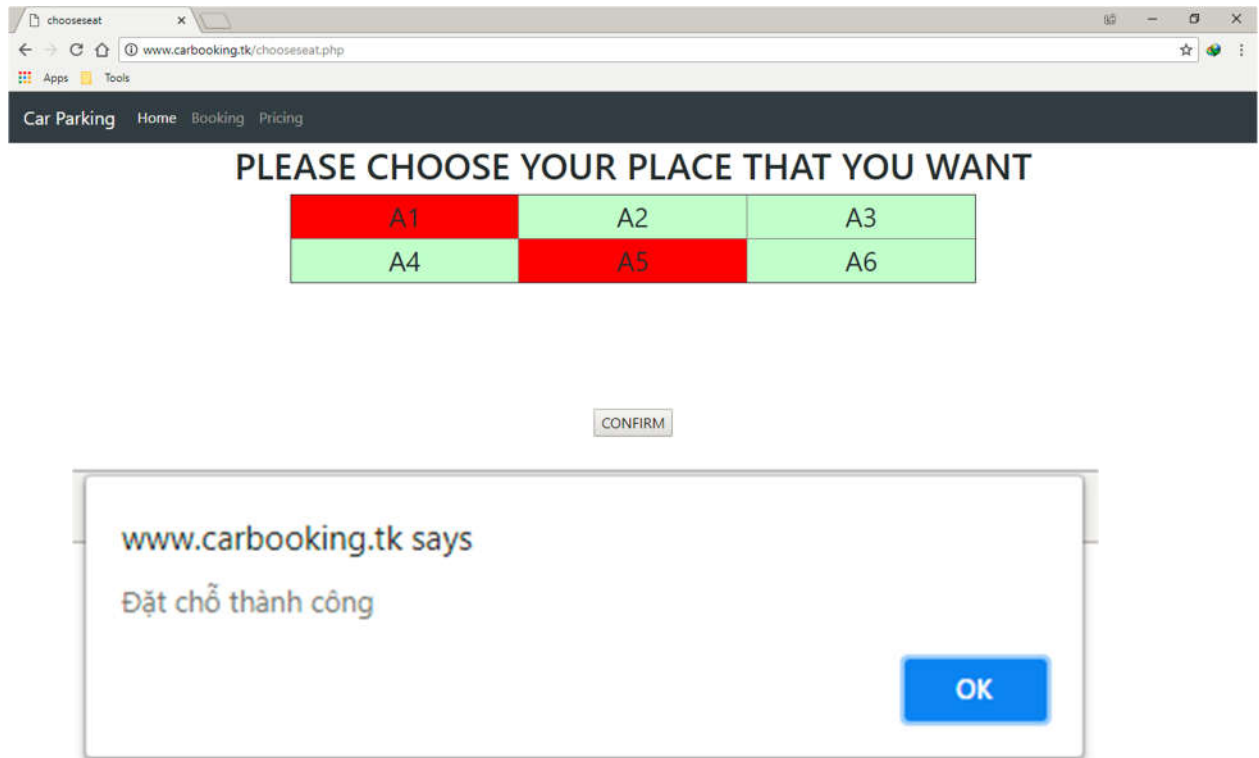
A1	A2	A3
A4	A5	A6

CONFIRM

Powered by 000webhost

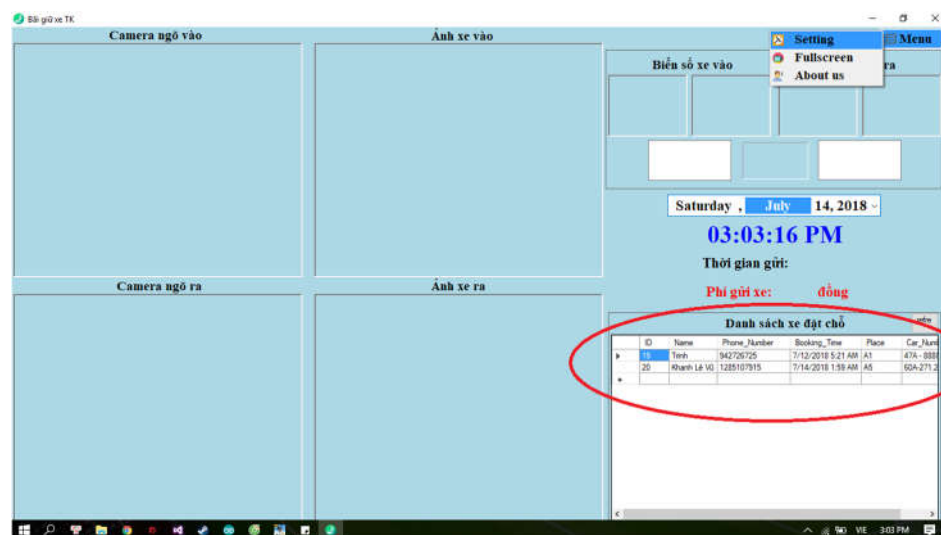
Hình 5.2. Trang chọn chỗ khi chưa chọn chỗ A5.

Sau khi chọn chỗ xong ta nhấn nút “Confirm” hệ thống sẽ xuất hiện một thông báo để cho người dùng biết là đã đặt chỗ thành công như hình.



Hình 5.3. Đặt chỗ thành công.

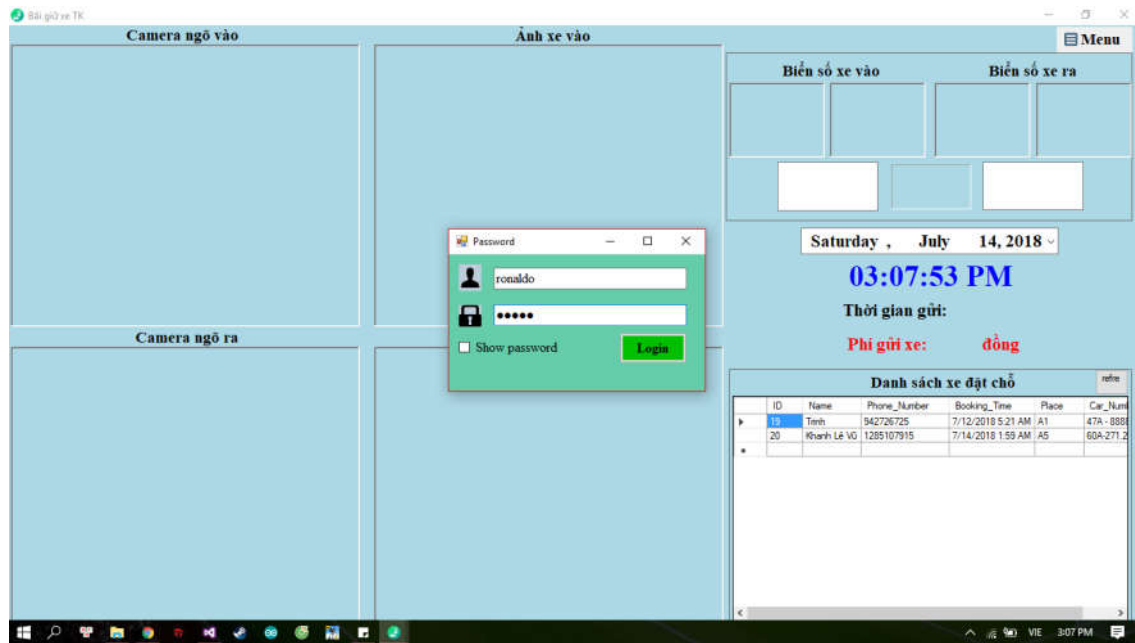
Khi đặt chỗ thành công thông tin đặt chỗ sẽ được hiển thị trên phần mềm quản lý



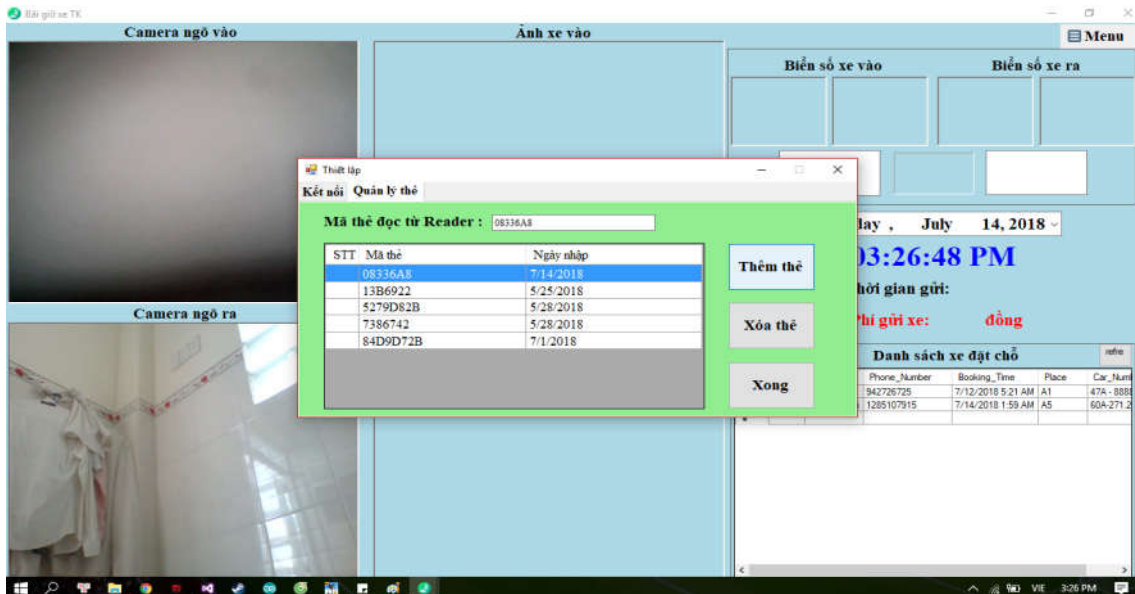
Hình 5.4. Giao diện khi book chỗ thành công.

Cuối cùng ta tiến hành kiểm tra chức năng của phần mềm quản lý trên C# được trình bày như hình sau:

Hình 5.5 mô tả giao diện log in. Khi muốn sử dụng phần mềm cần có user và password để đăng nhập vào hệ thống.

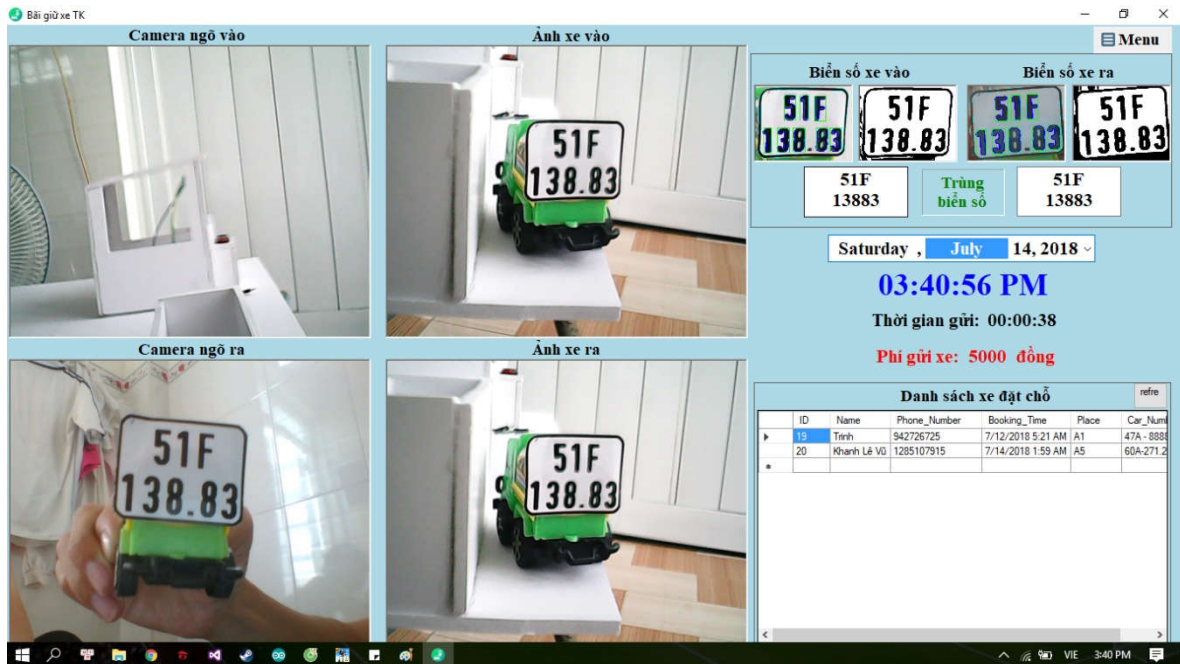


Hình 5.5. Giao diện Log In vào hệ thống.



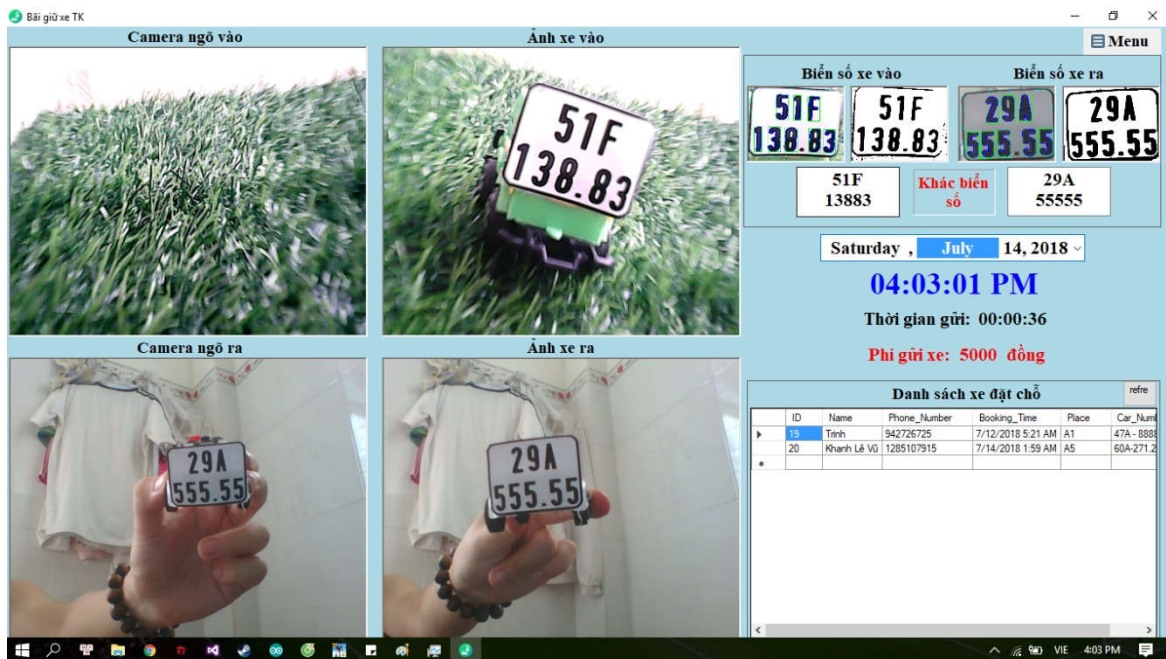
Hình 5.6. Giao diện Set thẻ.

Hình 5.7 mô tả giao diện đang hoạt động và nhận diện được đúng biển số xe



Hình 5.7. Giao diện chụp hình khi xe đúng biển số.

Ngược lại là không nhận diện đúng biển số hay sử dụng sai thẻ khi ra khỏi bãi. Hệ thống sẽ báo là “khác biển số”.



Hình 5.8. Giao diện chụp hình khi xe sai biển số.

Chương 6. Kết luận và hướng phát triển

6.1. Kết luận

Sau thời gian thực hiện đề tài đến nay nhóm đã hoàn thành các yêu cầu đặt ra, đã thiết kế và thi công thành công mô hình bãi giữ xe kết hợp với phần mềm quản lý bãi xe ứng dụng công nghệ RFID và C#.

Hoàn thành đề tài này đã giúp cho nhóm nắm thêm được nhiều kiến thức bổ ích như:

- Hiểu và lập trình các Board Arduino thông dụng trong các ứng dụng điều khiển.
- Hiểu thêm về các chuẩn và phương thức giao tiếp thường gặp với vi điều khiển như: UART, SPI...
- Hiểu thêm về các thuật toán xử lý ảnh
- Nguyên lý hoạt động, cách điều khiển và ứng dụng của công nghệ RFID.
- Nguyên lý hoạt động và cách điều khiển động cơ servo.
- Có thêm nhiều kiến thức về lập trình phần mềm trên C#.
- Ứng dụng được nhiều công cụ hỗ trợ lập trình về xử lý hình ảnh, SQL, MySql, EmguCV... trên C#.

Bên cạnh đó do thời gian cũng như kiến thức còn hạn hẹp nên cũng không tránh khỏi một số điều cần cải thiện như:

- Chưa có tính năng tự động nhận diện biển số cho phần mềm quản lý.
- Phần mềm quản lý chưa có tính bảo mật cao.

6.2. Hướng phát triển

Qua các điểm cần cải thiện được nêu ra đề tài có thể được phát triển theo các hướng rộng hơn và hoàn thiện hơn như:

- Tích hợp, thay thế thêm nhiều cảm biến để nâng tính ổn định, thông minh cho hệ thống.
- Backup dữ liệu lên cloud để tránh trường hợp hệ thống bị mất điện hay xảy ra sự cố.
- Lập trình thêm chức năng tự động nhận dạng biển số, nhận diện người lái xe cho phần mềm quản lý.

- Kết hợp thêm các tấm pin năng lượng mặt trời đặt ở bãi xe, hướng đến sử dụng nguồn năng lượng sạch, xanh và tiết kiệm.
- Tăng cường tính bảo mật để hệ thống hoạt động bảo mật hơn.

6.3. Ứng dụng trong tương lai gần

Do được thiết kế dưới dạng mô hình, nên phần cứng hệ thống cần được cải thiện nhiều để ứng dụng vào thực tế. Tuy nhiên, về cơ bản hệ thống đã vận hành được đúng các yêu cầu được đề ra có thể ứng dụng vào:

- Làm giải pháp cho các bãi xe có công nghệ thông minh.
- Làm phương thức quản lý, điều khiển cho các bãi xe thông minh, hiện đại.
- Tạo tiền đề phát triển cho các hệ thống IoT.

Phần mềm quản lý bãi xe có thể lập trình lại để kết nối với camera an ninh công nghiệp, camera IP để ứng dụng vào quản lý bãi xe cho các trung tâm thương mại, siêu thị, trường học... tương tự với các hệ thống phần mềm quản lý bãi xe thông minh hiện có.

Tài liệu tham khảo

- [1]. Giáo trình “Lập trình Android trong ứng dụng điều khiển – Nguyễn Văn Hiệp” – Đại Học Sư Phạm Kỹ Thuật Tp.HCM tháng 8/2015.
- [2]. Giáo trình “Thực tập vi điều khiển PIC – Nguyễn Đình Phú” – Đại Học Sư Phạm Kỹ Thuật Tp.HCM tháng 8/2015.
- [3]. Giáo trình “Lập trình hướng đối tượng với PHP – Đinh Vũ Quốc Trung” – Đại Học FPT.
- [4]. Giáo trình “Xử lý ảnh – Nguyễn Thanh Hải” – Đại Học Sư Phạm Kỹ Thuật Tp.HCM.
- [5]. https://en.wikipedia.org/wiki/Canny_edge_detector

Phụ Lục

- 1. Code chương trình cho vi điều khiển (Đĩa CD)**
- 2. Chương trình cho RFID-Reader (Đĩa CD)**
- 3. Chương trình điều khiển C# (Đĩa CD)**
- 4. Chương trình Web (Đĩa CD)**
- 5. AT Mega 328P Datasheet**
- 6. LCD Datasheet**
- 7. SG90 Datasheet**
- 8. Video Demo**
- 9. Hướng dẫn vận hành hệ thống**
 - B1: Kiểm tra các mối nối dây điện, các thiết bị.
 - B2: Bật nguồn, khởi động hệ thống.
 - B3: Kiểm tra các thiết bị có nguồn chưa.
 - B4: Gắn đầu đọc thẻ, camera, khối xử lý trung tâm với PC qua cổng USB.
 - B5: Mở phần mềm trên máy tính, kết nối đầu đọc thẻ, camera, khối xử lý trung tâm.
 - B6: Có xe vào tiến hành quét thẻ, mở Barrier.
 - B7: Đặt chỗ trước trên Web.