

**Đại học GTVT TP HCM**  
**Khoa Điện – ĐTVT**  
**Bộ môn Tư Động Hóa**

# CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

## **Độc lập – Tự do – Hạnh phúc**

*TP Hồ Chí Minh, ngày 22 tháng 08 năm 2011*

## BẢN NHẬN XÉT

## **LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

HO VÀ TÊN: (1) LÊ HỒNG LONG MSSV: TD06024

(2) NGUYỄN VĂN THẮNG

## NGÀNH: TỰ ĐỘNG HÓA CÔNG NGHIỆP

LÓP: TD06

## TÊN ĐỀ TÀI:

# **ỨNG DỤNG XỬ LÝ ẢNH NHẬN DẠNG BIỂN SỐ XE VÀ ĐIỀU KHIỂN MÔ HÌNH NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG**

GIÁO VIÊN HƯỚNG DẪN: TS. NGUYỄN MINH THANH

## 1. NỘI DUNG CỦA ĐỀ TÀI

- Giới thiệu tổng quan về ứng dụng xử lý ảnh nhận dạng biển số xe và điều khiển mô hình nhà giữ xe ô tô tự động.
  - Sử dụng Matlab để lập trình phần mềm nhận dạng biển số xe ô tô.
  - Sử dụng PLC S7 – 300 và phần mềm giám sát hệ thống WinCC để điều khiển mô hình nhà giữ xe ô tô tự động.
  - Tạo giao thức kết nối Matlab và WinCC để truyền nhận dữ liệu.
  - Xuất Excel để báo cáo thu chi và trạng thái hoạt động của hệ thống nhà giữ xe ô tô tự động.
  - Thiết kế và thi công mô hình nhà giữ xe ô tô tự động nhiều cửa vào.

## **2. NHÂN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

---

---

---

---

---

---

### 3. ĐÁNH GIÁ LUẬN VĂN

**LOẠI:** ..... **ĐIỂM:** .....

TP HCM, ngày ..... tháng ..... năm 2011

## GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

**Đại học GTVT TP HCM**  
**Khoa Điện – ĐTVT**  
**Bộ môn Tư Động Hóa**

# CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

## **Độc lập – Tự do – Hạnh phúc**

*TP Hồ Chí Minh, ngày 22 tháng 08 năm 2011*

## BẢN NHẬN XÉT

## **LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

HỌ VÀ TÊN: (1) LÊ HỒNG LONG MSSV: TD06024

(2) NGUYỄN VĂN THẮNG

## **NGÀNH: TỰ ĐỘNG HÓA CÔNG NGHIỆP**

LÓP: TD06

## TÊN ĐỀ TÀI:

# ỨNG DỤNG XỬ LÝ ẢNH NHẬN DẠNG BIỂN SỐ XE VÀ ĐIỀU KHIỂN MÔ HÌNH NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG

**GIÁO VIÊN PHẢN BIỆN: TS. LÊ QUANG ĐỨC**

#### **4. NỘI DUNG CỦA ĐỀ TÀI**

- Giới thiệu tổng quan về ứng dụng xử lý ảnh nhận dạng biển số xe và điều khiển mô hình nhà giữ xe ô tô tự động.
  - Sử dụng Matlab để lập trình phần mềm nhận dạng biển số xe ô tô.
  - Sử dụng PLC S7 – 300 và phần mềm giám sát hệ thống WinCC để điều khiển mô hình nhà giữ xe ô tô tự động.
  - Tạo giao thức kết nối Matlab và WinCC để truyền nhận dữ liệu.
  - Xuất Excel để báo cáo thu chi và trạng thái hoạt động của hệ thống nhà giữ xe ô tô tự động.
  - Thiết kế và thi công mô hình nhà giữ xe ô tô tự động nhiều cửa vào.

## 5. NHÂN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

## 6. ĐÁNH GIÁ LUẬN VĂN

**LOẠI:** ..... **ĐIỂM:** .....

TP HCM, ngày ..... tháng ..... năm 2011

GIÁO VIÊN PHẢN BIỆN

(Ký ghi rõ họ tên)

## LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn các thầy cô trong khoa Điện – Điện Tử Viễn Thông và bộ môn Tự Động Hóa trường Đại Học Giao Thông Vận Tải TP. Hồ Chí Minh đã tận tụy dạy dỗ, truyền đạt cho chúng em những kiến thức quý báu trong những năm học vừa qua để chúng em có kiến thức hoàn thành tốt đề tài luận văn tốt nghiệp này.

Chúng em xin chân thành cảm ơn Giáo Viên Hướng Dẫn - Thầy Nguyễn Minh Thạnh và các Thầy Cô trong trường đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian học tập và thực hiện luận văn tốt nghiệp.

Chúng em xin chân thành cảm ơn các anh chị đi trước của lớp TD04, TD05 đã nhiệt tình giúp đỡ chúng em về mặt kiến thức cũng như kinh nghiệm trong suốt quá trình thực hiện đề tài.

Chúng con xin chân thành cảm ơn gia đình và người thân đã ủng hộ, động viên chúng con trong quá trình học tập, nghiên cứu.

Nhóm xin chân thành cảm ơn các bạn trong tập thể TD06 đã tham gia đóng góp ý kiến trong suốt quá trình thực hiện để nhóm có thể hoàn thành tốt đề tài này.

Mặc dù chúng em đã cố gắng hết sức để hoàn thành tốt đề tài môn học, nhưng với hạn chế về kiến thức và kinh nghiệm thực tế chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em kính mong nhận được sự tận tình chỉ bảo của quý thầy cô và ý kiến đóng góp của các bạn để đề tài được hoàn thiện hơn.

Sau thời gian 5 năm học tập, vận dụng kiến thức học được nhóm nỗ lực thực hiện đề tài với tinh thần nghiêm túc, trách nhiệm và sự chỉ bảo tận tình của giáo viên hướng dẫn, chúng em cơ bản đã hoàn thành tốt đề tài cũng như nhiệm vụ mà luận văn tốt nghiệp đe ra.

Nhóm sinh viên thực hiện

Lê Hồng Long – Nguyễn Văn Thắng

## MỤC LỤC

<b>TÓM TẮT.....</b>	<b>xv</b>
<b>CHƯƠNG 1 .....</b>	<b>1</b>
<b>CƠ SỞ LÝ THUYẾT XỬ LÝ ẢNH SỐ.....</b>	<b>1</b>
1.1    Mở đầu.....	1
1.2    Khái quát về hệ thống thị giác và cảm biến thị giác .....	1
1.3    Khái quát quá trình xử lý ảnh số .....	4
1.4    Các thành phần của hệ thống nhận dạng biển số xe .....	6
1.4.1    Thành phần thu thập ảnh, Camera và vấn đề định dạng ảnh .....	6
1.4.2    Phần mềm lập trình xử lý ảnh.....	7
<b>CHƯƠNG 2 .....</b>	<b>8</b>
<b>PHẦN MỀM MATLAB.....</b>	<b>8</b>
2.1    Tổng quan về Matlab.....	8
2.1.1    Khái niệm về Matlab .....	8
2.1.2    Cấu trúc dữ liệu của Matlab .....	8
2.1.3    Các Toolbox của Matlab .....	9
2.1.4    Giao diện Matlab.....	9
2.2    GUI trong Matlab .....	11
2.2.1    Giới thiệu.....	11
2.2.2    Thao tác trên giao diện thiết kế GUI.....	12
2.3    Toolbox xử lý ảnh Matlab .....	14
2.3.1    Các loại kiểu dữ liệu ảnh trong Matlab .....	14
2.3.2    Các hàm xử lý ảnh cơ bản của Matlab.....	17
<b>CHƯƠNG 3 .....</b>	<b>21</b>

<b>QUY TRÌNH VÀ THUẬT TOÁN NHẬN DẠNG BIỂN SỐ XE .....</b>	<b>21</b>
3.1      Quy trình nhận dạng biển số xe .....	21
3.2      Thuật toán nhận dạng biển số xe.....	21
3.2.1    Tiền xử lý.....	21
3.2.2    Tự động trích vùng bảng số xe .....	22
3.2.3    Chỉnh độ nghiêng.....	25
3.2.4    Phân đoạn các ký tự có trong vùng biển số.....	27
3.2.5    Nhận dạng ký tự.....	29
3.3      Hình ảnh kết quả nhận dạng .....	30
3.3.1    Nhận dạng biển số 1 hàng ký tự (Hình 3.14) .....	30
3.3.2    Nhận dạng biển số 2 hàng ký tự (Hình 3.15) .....	30
3.3.3    Nhận dạng biển số xanh với 2 hàng ký tự (Hình 3.16).....	31
3.3.4    Nhận dạng biển số nhiều ký tự (Hình 3.16) .....	31
<b>CHƯƠNG 4 .....</b>	<b>32</b>
<b>LIÊN KẾT GIỮA WINCC VÀ MATLAB THÔNG QUA OPC .....</b>	<b>32</b>
4.1      Tổng quan về giao thức OPC (Hình 4.1).....	32
4.2      OPC WinCC.....	33
4.3      OPC Matlab .....	34
4.3.1    Tạo kết nối OPC trong Matlab .....	34
4.3.2    Tạo nhóm biến dữ liệu liên kết với WinCC .....	34
4.3.3    Đọc và ghi dữ liệu giữa Matlab và WinCC .....	35
<b>CHƯƠNG 5 .....</b>	<b>36</b>
<b>TỔNG QUAN VỀ NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG.....</b>	<b>36</b>
5.1      Tổng quan .....	36

5.2	Các giải pháp nhà giữ xe ô tô tự động .....	38
5.2.1	Giải pháp “Xếp chồng” (Hình 5.2) .....	38
5.2.2	Giải pháp “Nhà gửi xe nhiều tầng” (Hình 5.3).....	39
5.2.3	Giải pháp “Nhà gửi xe tự động lộ thiên” (Hình 5.4).....	40
5.2.4	Giải pháp “Nhà gửi xe tự động dạng ngầm” (Hình 5.6).....	42
5.2.5	Nhược điểm chung của hệ thống đỗ xe tự động gấp phải.....	44
5.3	Lựa chọn phương án nhà giữ xe ô tô tự động.....	46
<b>CHƯƠNG 6</b>	<b>.....</b>	<b>48</b>
<b>ĐIỀU KHIỂN ĐỘNG CƠ BUỚC</b>	<b>.....</b>	<b>48</b>
6.1	Động cơ bước .....	48
6.1.1	Phân loại và cấu tạo .....	48
6.1.2	Đặc tính của động cơ bước.....	52
6.2	Các phương pháp điều khiển động cơ bước .....	54
6.2.1	Chế độ điều khiển bước đủ kiểu 1 pha.....	55
6.2.2	Chế độ điều khiển bước đủ kiểu 2 pha.....	55
6.2.3	Chế độ điều khiển nửa bước .....	56
6.3	Mạch điều khiển động cơ bước.....	57
6.3.1	Sơ đồ nguyên lý (Hình 6.14) .....	57
6.3.2	Mạch lái động cơ bước (Hình 6.15).....	57
<b>CHƯƠNG 7</b>	<b>.....</b>	<b>59</b>
<b>BỘ ĐIỀU KHIỂN KHẢ LẠP TRÌNH PLC S7-300</b>	<b>.....</b>	<b>59</b>
7.1	Tổng quan PLC S7 – 300 .....	59
7.2	Module CPU (Hình 7.4) .....	62
7.3	Module IM (Hình 7.6) .....	66

7.4	Module tín hiệu .....	67
<b>CHƯƠNG 8 .....</b>		<b>70</b>
<b>TỔNG QUAN VỀ HỆ SCADA .....</b>		<b>70</b>
8.1	Khái quát hệ thống SCADA (Hình 8.1).....	70
8.2	Mô hình phân cấp chức năng .....	72
8.2.1	Mô hình phân cấp (Hình 8.2) .....	72
8.2.2	Chức năng nhiệm vụ .....	74
8.3	Cấu trúc hệ SCADA (Hình 8.4) .....	75
8.3.1	Cấu trúc .....	75
8.3.2	Phần cứng .....	78
8.3.3	Phần mềm .....	80
8.4	Đặc tính chính của hệ thống SCADA .....	81
8.4.1	Kiến trúc hệ thống .....	81
8.4.2	Các đơn vị đo lường từ xa RTU (Remote Terminal Unit) .....	83
8.4.3	Cơ sở dữ liệu .....	83
8.4.4	Truyền thông máy tính .....	83
8.5	Ứng dụng của hệ thống SCADA .....	84
8.5.1	Ưu điểm của hệ thống SCADA .....	84
8.5.2	Nhược điểm của hệ thống SCADA .....	84
<b>CHƯƠNG 9 .....</b>		<b>85</b>
<b>GIỚI THIỆU PHẦN MỀM WINCC VÀ GIAO DIỆN CỦA ĐỀ TÀI.....</b>		<b>85</b>
9.1	Giới thiệu WinCC .....	85
9.2	Đặc trưng cơ bản .....	86
9.3	Chức năng của WinCC .....	86

9.4	Một dự án mẫu của WinCC .....	88
9.5	Giao diện WinCC thiết kế cho đè tài – Bài đồ xe tự động .....	89
9.5.1	Sơ đồ tổng quan hệ thống nhà giữ xe ô tô tự động (Hình 9.4) .....	89
9.5.2	Giao diện đăng nhập điều khiển hệ thống (Hình 9.5) .....	89
9.5.3	Giao diện điều khiển hệ thống bán tự động (Hình 9.6) .....	90
9.5.4	Giao diện điều khiển hệ thống tự động (Hình 9.7) .....	90
<b>CHƯƠNG 10</b>	.....	<b>91</b>
<b>GIẢI THUẬT ĐIỀU KHIỂN NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG</b> .....		<b>91</b>
10.1	Giải thuật điều khiển chương trình chính (Hình 10.1).....	912
10.2	Giải thuật chương trình điều khiển bán tự động (Hình 10.2).....	913
10.3	Giải thuật chương trình điều khiển tự động (Hình 10.3).....	914
10.4	Giải thuật xác định vị trí có xe gần nhất để di chuyển tay máy về lấy xe (Hình 10.4).....	915
10.5	Giải thuật đường đi ngắn nhất cho khoang (Hình 10.5).....	916
10.6	Chương trình con điều khiển động cơ bước (Hình 10.6) .....	917
<b>CHƯƠNG 11</b>	.....	<b>98</b>
<b>THIẾT KẾ VÀ THI CÔNG MÔ HÌNH PHẦN CỨNG</b> .....		<b>98</b>
11.1	Thiết kế mạch lái động cơ bước (Hình 11.1).....	98
11.2	Thiết kế cánh tay máy lấy xe (Hình 11.2) .....	98
11.3	Mô hình nhà giữ xe ô tô tự động (Hình 11.3).....	99
<b>CHƯƠNG 12</b>	.....	<b>100</b>
<b>KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN CỦA ĐÈ TÀI</b> .....		<b>100</b>
12.1	Kết quả đạt được .....	100
12.2	Hạn chế của đè tài .....	100

12.3	Hướng phát triển đề tài .....	101
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>102</b>
<b>PHỤ LỤC .....</b>		<b>104</b>
<b>CHƯƠNG TRÌNH MATLAB .....</b>		<b>104</b>
<b>CHƯƠNG TRÌNH PLC S7 – 300 .....</b>		<b>138</b>

## **DANH MỤC HÌNH ẢNH**

Hình 1.1 Các bước cơ bản của xử lý ảnh số .....	4
Hình 2.1 Giao diện sử dụng của Matlab.....	9
Hình 2.2 Giao diện thiết kế của GUI.....	11
Hình 2.3 Thao tác chọn bảng thuộc tính của đối tượng .....	12
Hình 2.4 Bảng thuộc tính của đối tượng .....	13
Hình 2.5 Giao diện lập trình M-file của GUI .....	14
Hình 2.6 Biểu diễn dữ liệu ảnh Index .....	15
Hình 2.7 Ảnh Greyscale .....	15
Hình 2.8 Ảnh nhị phân .....	16
Hình 2.9 Ảnh màu thực RGB .....	16
Hình 3.1 Các khâu tiền xử lý ảnh để nâng cao chất lượng ảnh đầu vào .....	22
Hình 3.2 Các loại hình dáng biển số xe ở Việt Nam.....	22
Hình 3.3 Tự động trích vùng biển số xe kiểu 1 hàng ký tự .....	23
Hình 3.4 Tự động trích vùng biển số xe kiểu 2 hàng ký tự .....	24
Hình 3.6 Biển số có độ nghiêng .....	25
Hình 3.7 Chính độ nghiêng theo phương thẳng đứng.....	25
Hình 3.8 Chính độ nghiêng theo phương nằm ngang .....	26
Hình 3.9 Chính độ nghiêng cho biển số 1 hàng ký tự theo phương ngang.....	26
Hình 3.10 Chính độ nghiêng cho biển số 2 hàng ký tự theo phương thẳng đứng.....	27
Hình 3.11 Chuyển sang nhị phân và loại bỏ ốc vít trên biển số .....	27
Hình 3.12 Phân đoạn từng ký tự có trong biển số xe .....	28
Hình 3.13 Bảng dữ liệu mẫu lưu trữ trong Matlab .....	29
Hình 3.14 Nhận dạng biển số 1 hàng ký tự .....	30
Hình 3.15 Các khâu nhận dạng biển số xe hoàn chỉnh .....	30
Hình 3.16 Các khâu nhận dạng biển số xe hoàn chỉnh .....	31
Hình 3.16 Nhận dạng biển số nhiều ký tự .....	31
Hình 4.1 Mô tả giao tiếp Matlab và WinCC qua OPC.....	32
Hình 4.2 Tạo OPC cho WinCC.....	33

Hình 4.3 Tạo OPC server và biến dữ liệu.....	34
Hình 5.1 Ô tô đậu dày đặc trên vỉa hè và lòng đường tại TP Hồ Chí Minh .....	36
Hình 5.2 Giải pháp đậu xe “xếp chồng” .....	39
Hình 5.3 Mô hình nhà gửi xe nhiều tầng .....	39
Hình 5.4 Giải pháp nhà gửi xe tự động .....	40
Hình 5.5 Mô hình nhà gửi xe tự động hình trụ .....	41
Hình 5.6 Mô hình nhà gửi xe tự động dạng ngầm .....	42
Hình 5.7 Quá trình xây dựng một nhà gửi xe tự động dạng ngầm.....	43
Hình 5.8 Cảnh quan phía trên của nhà gửi xe tự động dạng ngầm .....	43
Hình 5.9 Nhà gửi xe tự động hình trụ .....	47
Hình 6.1 Cấu tạo động cơ bước nam châm vĩnh cửu.....	49
Hình 6.2 Đâu dây động cơ bước lưỡng cực.....	49
Hình 6.3 Đâu dây stator động cơ bước đơn cực .....	50
Hình 6.4 Đâu dây stator động cơ nhiều pha .....	50
Hình 6.5 Động cơ bước từ trờ thay đổi .....	51
Hình 6.6 Động cơ bước hỗn hợp.....	52
Hình 6.7 Động cơ bước .....	54
Hình 6.8 Mã xung điều khiển bước đủ kiểu 1 pha.....	55
Hình 6.9 Biểu diễn hoạt động điều khiển đủ bước kiểu 1 pha .....	55
Hình 6.10 Mã xung điều khiển bước đủ kiểu 2 pha.....	55
Hình 6.11 Biểu diễn hoạt động điều khiển đủ bước kiểu 2 pha .....	56
Hình 6.12 Mã xung điều khiển kiểu nửa bước .....	56
Hình 6.13 Biểu diễn hoạt động điều khiển kiểu nửa bước .....	56
Hình 6.14 Sơ đồ nguyên lý mạch điều khiển động cơ bước .....	57
Hình 6.15 Mạch lái động cơ bước.....	57
Hình 7.1 Các module được gắn trên cùng 1 thanh ray.....	59
Hình 7.2 Định nghĩa địa chỉ cho các module.....	60
Hình 7.3 Hình các module được gắn trên nhiều rack.....	61
Hình 7.5 Các thông số của các vùng nhớ của PLC .....	65

Hình 7.6 Module IM.....	66
Hình 7.7 Cấu tạo của Module SM331 AI 2*12 .....	68
Hình 7.8 Tầm đo và phương pháp đo cho các loại module.....	69
Hình 8.1 Hệ thống giám sát và thu thập dữ liệu SCADA .....	70
Hình 8.2 Mô hình phân cấp chức năng của hệ thống SCADA.....	72
Hình 8.3 Một hệ thống SCADA điển hình .....	74
Hình 8.4 Cấu trúc chung của một hệ scada .....	75
Hình 8.5 Hệ thống Scada thiết kế cho nhà máy đường.....	77
Hình 8.6 Mô hình một hệ SCADA .....	78
Hình 8.7 Cấu trúc của thiết bị đầu cuối (RTU).....	79
Hình 8.8 Cấu trúc phần mềm của hệ thống SCADA .....	80
Hình 8.9 Kiến trúc phân bố tiêu biểu của một hệ SCADA .....	81
Hình 9.1 Tạo một dự án WinCC .....	85
Hình 9.2 Cửa sổ làm việc WinCC Explorer .....	86
Hình 9.3 Một giao diện HMI được tạo từ WinCC .....	88
Hình 9.4 Sơ đồ tổng quan hệ thống nhà giữ xe ô tô tự động .....	89
Hình 9.5 Giao diện đăng nhập điều khiển hệ thống .....	89
Hình 9.6 Giao diện điều khiển hệ thống bán tự động .....	90
Hình 9.7 Giao diện điều khiển hệ thống bán tự động .....	90
Hình 10.1 Giải thuật chương trình chính điều khiển nhà giữ xe ô tô tự động....	92
Hình 10.2 Giải thuật chương trình điều khiển bán tự động.....	93
Hình 10.3 Giải thuật chương trình điều khiển tự động .....	94
Hình 10.4 Giải thuật xác định vị trí có xe gần nhất để tay máy về lấy xe .....	95
Hình 10.5 Chương trình con chọn đường đi ngắn nhất cho khoang.....	96
Hình 10.6 Chương trình con điều khiển động cơ bước .....	97
Hình 11.1 Mạch lái điều khiển 3 động cơ bước.....	98
Hình 11.2 Mạch lái điều khiển 3 động cơ bước.....	98
Hình 11.3 Mô hình nhà giữ xe ô tô tự động .....	99

## TÓM TẮT

*Hiện nay ở các thành phố lớn như TP Hồ Chí Minh, Hà Nội ... số lượng xe ô tô ngày càng nhiều mà diện tích đất sử dụng ngày càng bị thu hẹp nên việc đỗ xe ô tô hết sức hạn chế, phải tận dụng lòng lề đường, sân bãi để làm nơi đỗ xe tạm thời vừa rất chiếm diện tích vừa không an toàn. Để giải quyết vấn đề này đã và đang có nhiều giải pháp như: tầng hầm đỗ xe, nhà đỗ xe nhiều tầng... nhưng các giải pháp này chưa phát huy hiệu quả cao. Việc ứng dụng công nghệ tự động hóa vào hệ thống nhà giữ xe ô tô là một giải pháp mới mẻ ở Việt Nam để giải quyết vấn đề cấp bách nói trên. Nhìn nhận được nhu cầu thực tế đó, nhóm đưa ra giải pháp: “**Ứng dụng xử lý ảnh nhận dạng biển số xe và Điều khiển nhà giữ xe ô tô tự động – dạng lô thiêng hình trụ**”, hệ thống sử dụng ngôn ngữ Matlab để nhận dạng biển số xe kết hợp PLC S7 300, phần mềm WinCC của hãng Siemens điều khiển và giám sát hoạt động của nhà giữ xe. Để quản lý thông tin xe vào ra, toàn bộ thông tin và dữ liệu như biển số xe, các thông số quá trình gửi và nhận xe được cập nhật tự động vào báo cáo thu chi cuối ngày trên Excel.*

Nội dung thực hiện đề tài bao gồm các vấn đề chính như sau:

- Lý thuyết xử lý và nhận dạng ảnh.
- Phần mềm lập trình và thiết kế giao diện người dùng Matlab.
- Quy trình và thuật toán nhận dạng biển số xe.
- Liên kết Matlab và WinCC thông qua giao thức OPC.
- Các giải pháp nhà giữ xe tự động.
- Hệ thống giám sát, điều khiển và thu thập dữ liệu SCADA.
- Bộ điều khiển khả trình PLC S7-300.
- Phần mềm WinCC để thiết kế giao diện HMI.
- Điều khiển động cơ bước.
- Lưu đồ giải thuật và lập trình điều khiển hệ thống nhà giữ xe.
- Thiết kế và thi công mô hình nhà giữ xe ô tô dạng hình trụ 4 tầng.
- Xuất cáo thu chi và thông số của hệ thống ra Excel.

## CHƯƠNG 1

### CƠ SỞ LÝ THUYẾT XỬ LÝ ẢNH SỐ

#### **1.1 Mở đầu**

Xử lý ảnh trong máy tính còn gọi là thị giác máy là một lĩnh vực mang tính khoa học và công nghệ đã và đang rất phát triển. Khái niệm thị giác máy có liên quan tới nhiều ngành học và hướng nghiên cứu khác nhau. Từ những năm 1970 khi mà năng lực tính toán của máy tính ngày càng trở nên mạnh mẽ hơn, các máy tính lúc này có thể xử lý được những tập dữ liệu lớn như các hình ảnh, các đoạn phim thì khái niệm và kỹ thuật về thị giác máy ngày càng được nhắc đến và nghiên cứu nhiều hơn cho tới ngày nay.

Thị giác máy bao gồm lý thuyết và các kỹ thuật liên quan nhằm mục đích tạo ra một hệ thống nhân tạo có thể tiếp nhận thông tin từ các hình ảnh thu được hoặc các tập dữ liệu đa chiều.

Ngày nay, ứng dụng của thị giác máy đã trở nên rất rộng lớn và đa dạng, len lỏi vào mọi lĩnh vực từ quân sự, khoa học, vũ trụ, cho đến y học, sản xuất, và tự động hóa tòa nhà.

#### **1.2 Khái quát về hệ thống thị giác và cảm biến thị giác**

Hệ thống thị giác là những hệ thống tiếp nhận thông tin từ các cảm biến thị giác với mục đích cho phép máy móc đưa ra những quyết định thông minh.

Thị giác máy là một ngành khoa học mới phát triển. Mặc dù đã có những ứng dụng của xử lý ảnh số trong những thập niên đầu của thế kỷ XX vào một số lĩnh vực, nhưng phải đến những năm 1970, những nghiên cứu về lĩnh vực này mới được bắt đầu khi máy tính đã có thể quản lý các quá trình xử lý một lượng lớn dữ liệu như các ảnh số.

Lĩnh vực nghiên cứu của thị giác máy rất rộng, và đặc điểm chung là các bài toán về thị giác máy đều không có một đề bài chung và cách giải duy nhất. Mỗi giải

pháp giải quyết vấn đề đều được một kết quả nhất định cho những trường hợp cụ thể. Thị giác máy có liên quan đến rất nhiều ngành như tự động điều khiển, xử lý ảnh số, quang học, sinh học, toán học, máy học và trí tuệ nhân tạo. Sự kết hợp của những ngành này tạo cho thị giác máy một khả năng ứng dụng hết sức rộng lớn trong mọi lĩnh vực của khoa học, sản xuất và đời sống.

Có thể liệt kê một số ứng dụng của thị giác máy như sau:

- Điều khiển tiến trình (ví dụ: trong các robot công nghiệp, hay các thiết bị, xe tự hành).
- Phát hiện sự kiện (ví dụ: các thiết bị giám sát).
- Tự động hóa tòa nhà
- Mô hình hóa đối tượng (ví dụ: quá trình kiểm ra trong môi trường công nghiệp, xử lý ảnh trong y học).
- Tương tác (đóng vai trò làm đầu vào cho thiết bị trong quá trình tương tác giữa người và máy).
- Nhận dạng biển số xe, hình dạng vật thể, nhận dạng vân tay, mặt người ...

Trong hệ thống máy CNC, hoặc các dây chuyền công nghệ, các cảm biến thị giác thu thập dữ liệu ảnh về đối tượng công nghiệp, xử lý và tách đối tượng ra khỏi ảnh. Sau khi tách đối tượng, hệ thống thị giác máy tính toán các đặc trưng của đối tượng, như vị trí, hướng, để giúp cánh tay máy của robot công nghiệp thao tác chính xác việc gấp hoặc gia công đối tượng.

Với những hệ thống thị giác được tích hợp các camera có độ phân giải lớn, được lập trình chính xác, có thể điều khiển tay máy để thao tác với những vi mạch nhỏ đòi hỏi độ chính xác gần như tuyệt đối.

Cũng như vậy, với xe tự hành đi trong môi trường phức tạp, nhiều vật cản, hệ thống thị giác máy giúp cho xe phát hiện ra những đối tượng, vị trí và khoảng cách của chúng đối với xe. Trong trường hợp này, hệ thống thị giác máy không chỉ đóng vai trò như một cảm biến thị giác, mà còn thực hiện việc vẽ bản đồ đối tượng, cho phép xe tự hành chọn được đường đi thích hợp nhất.

Hệ thống thị giác máy còn được ứng dụng trong những lĩnh vực công nghiệp với vai trò như một cảm biến kiểm soát lỗi bề mặt sản phẩm. Camera thu thập hình ảnh về bề mặt sản phẩm, sẽ truyền dữ liệu vào cho hệ thống xử lý để tìm ra lỗi trên sản phẩm, vị trí lỗi và kích thước lỗi. Với những hệ thống thị giác sử dụng camera hồng ngoại, ta còn có thể đo nhiệt độ sản phẩm và sự phân bố nhiệt độ trên sản phẩm.

Như vậy, có thể nói, trong lĩnh vực công nghiệp, thị giác máy và cảm biến thị giác có thể thay thế một lượng lớn các cảm biến ví trí thông thường, vốn cần rất nhiều trong một dây chuyền sản xuất hoặc CNC, giúp giảm thiểu chi phí và công sức lắp đặt cảm biến, và quan trọng nhất là tạo nên một hệ thống xử lý thông nhất những thông tin về quá trình và đối tượng công nghiệp. Trong lĩnh vực tự động hóa tòa nhà, hệ thống thị giác máy cũng đóng một vai trò ngày càng quan trọng.

Với sự phát triển của các thuật toán xử lý dữ liệu ảnh, ứng dụng các thành tựu mới nhất của công nghệ xử lý và trí tuệ nhân tạo, các cảm biến thị giác ngày nay có thể thực hiện những chức năng thông minh như đếm số người trong phòng, nhận dạng đối tượng chuyển động, nhận dạng khuôn mặt, cảnh báo sự kiện, nhận dạng vân tay....

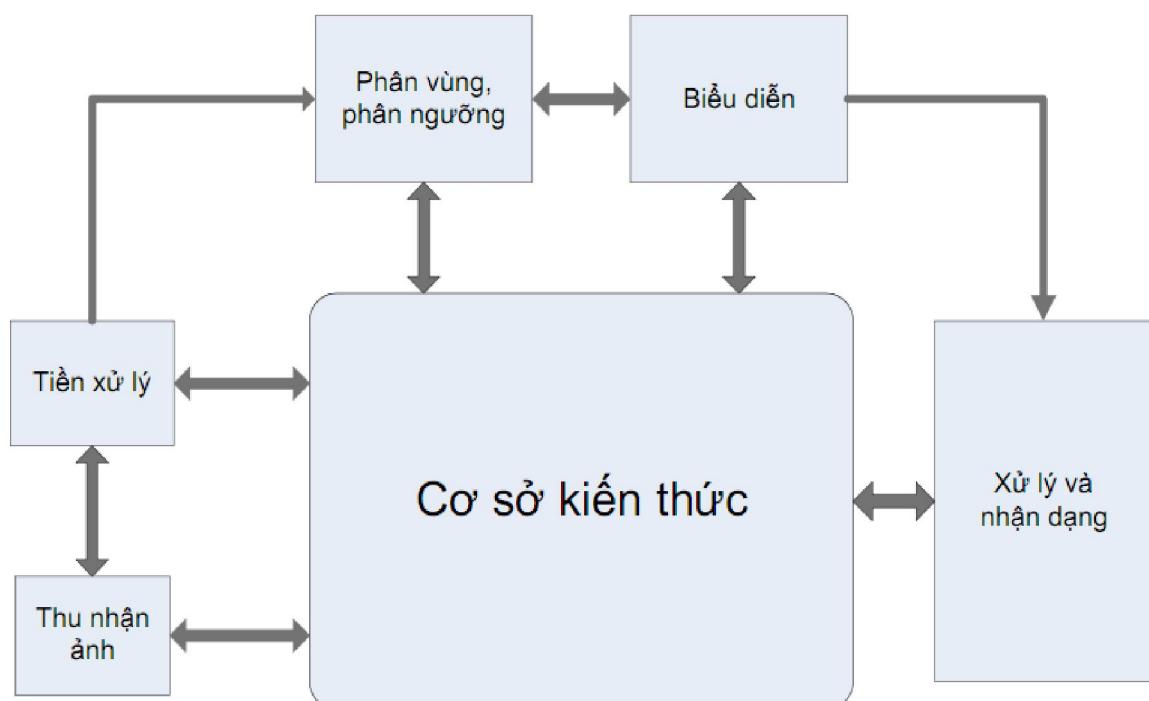
Xu thế phát triển của các cảm biến ngày nay, đó là độ chính xác cao, kèm với tính phân tán, thông minh, khả năng loại bỏ lỗi. Các hệ thống cảm biến thị giác đang chứng tỏ ngày càng đáp ứng được xu thế đấy. Các hệ thống cảm biến thị giác ngày nay, đã có thể thay thế được một lượng lớn cảm biến vị trí trong những bài toán cụ thể, đồng thời, khả năng xử lý thông tin không cần đến sự điều khiển

của bộ điều khiển trung tâm, cho phép chúng có thể tích hợp rộng rãi vào những hệ thống điều khiển phân tán.Thêm vào đó, các cảm biến thị giác có tính linh hoạt rất cao, có thể ứng dụng trong nhiều bài toán với những yêu cầu đo đặc và giám sát khác nhau. Và cuối cùng, do đặc điểm hoạt động của mình, các cảm biến thị giác có thể hoạt động tốt trong những môi trường khắc nghiệt như nhiệt độ cao, phóng xạ, bụi, điện trường, độ ẩm lớn...

Với những lý do đó, hệ thống thị giác máy và cảm biến thị giác đang ngày càng được ứng dụng nhiều trong những hệ thống phức tạp và hiện đại, hoạt động liên tục và đòi hỏi yêu cầu về độ chính xác và xử lý thông tin cao.

### 1.3 Khái quát quá trình xử lý ảnh số

Các bước cơ bản của xử lý ảnh số được mô tả trong sơ đồ dưới đây:



Hình 1.1 Các bước cơ bản của xử lý ảnh số

**Thu thập ảnh:** Ảnh số được thu thập bằng một cảm biến ảnh có khả năng biến thông tin về cường độ sáng và mức xám của ảnh thực thành tín hiệu điện áp dưới

dạng analog. Tín hiệu này sau đó được số hóa để trở thành tín hiệu số.

Hiện nay có một số cảm biến ảnh thực hiện cả việc thu nhận tín hiệu về cường độ sáng của ảnh và số hóa tín hiệu. Trong trường hợp cảm biến không có chức năng số hóa thì cần phải có một bộ biến đổi ảnh tương tự thành ảnh số. Tín hiệu ảnh sau khi được số hóa còn được mã hóa theo những chuẩn video nhất định trước khi được đưa vào quá trình lưu trữ và xử lý.

**Tiền xử lý ảnh:** Sau khi ảnh số được thu thập dưới dạng tín hiệu số, cần phải trải qua giai đoạn tiền xử lý. Chức năng chủ yếu của tiền xử lý là cải thiện ảnh, nâng cao các tính chất của ảnh giúp cho các quá trình xử lý về sau.

**Phân vùng ảnh:** Bước tiếp theo của quá trình xử lý là phân vùng ảnh. Ảnh sau khi đã được cải thiện, sẽ trở nên thuận tiện hơn cho việc phân ngưỡng và phân vùng. Nhiệm vụ chính của phân ngưỡng và phân vùng ảnh là tách ảnh đầu vào thành các đối tượng, vật thể riêng biệt. Kết quả của quá trình phân vùng ảnh, ta sẽ được một tập hợp các điểm ảnh có liên kết với nhau thành các đối tượng, được đánh số phân biệt, thuận tiện cho các quá trình xử lý cao hơn.

Đầu ra của quá trình phân vùng ảnh là các pixel chưa được lọc, bao gồm liên kết của 1 vùng hoặc tất cả các điểm ảnh trong vùng đó. Số liệu này cần được biến đổi thành dạng thích hợp cho máy tính xử lý.

**Phân tích ảnh:** đây là giai đoạn xử lý bậc cao trong hệ thống xử lý ảnh số. Ảnh sau khi được phân vùng thành các đối tượng riêng biệt, đã được đánh số phân biệt, sẽ được phân tích để phục vụ những mục đích khác nhau.

Xác định các đặc trưng hình học của đối tượng: dựa trên cơ sở đối tượng đã thu thập được ta có thể thực hiện xác định các đặc trưng hình học của mỗi đối tượng đầy đủ: vị trí, kích thước, hướng, ... và số đối tượng hay mật độ đối tượng trong ảnh. Đây là các đặc trưng được dùng nhiều trong hệ thống thị giác máy.

**Nhận dạng:** các đối tượng có thể là các vật thể có hình dạng nhất định, hoặc các kí tự số, chữ cái, dấu vân tay... Ảnh sau khi được phân vùng có thể được nhận dạng theo những phương pháp nhất định như phương pháp neutral, để tìm ra mẫu hình dạng mà đối tượng đó thuộc về.

Để hướng dẫn hoạt động của từng khối xử lý, cần có một hệ cơ sở kiến thức để kiểm tra hoạt động và tương tác giữa các khối. Hệ này có nhiệm vụ kiểm soát hoạt động của từng khối và sắp xếp trình tự hoạt động của chúng trong từng thời điểm, giải quyết bài toán xung đột.

#### 1.4 Các thành phần của hệ thống nhận dạng biến số xe

##### 1.4.1 Thành phần thu thập ảnh, Camera và vấn đề định dạng ảnh

Giới thiệu chung về camera: Trong hệ thống xử lý ảnh số, camera là một thiết bị rất quan trọng có chức năng quan sát và thu nhận ảnh đầu vào của hệ thống. Nó thường được coi là hộp đen trong đó có các quá trình biến đổi để chuyển một ảnh thành dạng lưu trữ trong máy tính. Các bước xử lý này bao gồm sự phát sáng, thấu kính, sensor, các phần tử quang điện và bộ số hóa, mỗi thành phần này phối hợp nhằm đưa ra ảnh số cuối cùng. Điểm đặc biệt quan trọng trong nhận dạng ảnh là đặc tính thời gian của camera, vì vậy quá trình xử lý ảnh có vai trò như bộ lấy mẫu trong hệ thống nhận dạng ảnh.

Cảm biến nhín chung gồm 2 thành phần chính. Thành phần thứ nhất tạo ra tín hiệu điện ở đầu ra tỉ lệ với mức năng lượng mà nó nhận được. Thành phần thứ 2 là bộ số hóa, là phần tử biến đổi tín hiệu tương tự thành tín hiệu số. Tùy thuộc vào thành phần số hóa thực chất là bộ chuyển đổi ADC, chúng ta có các tín hiệu với số bit khác nhau: 4 bit, 8bit, 10 bit, 12 bit..., tương ứng ta sẽ được các ảnh có 16, 256....mức xám khác nhau.

Trước đây, người ta sử dụng các camera dựa trên thiết bị phóng tia điện tử, linh kiện bán dẫn, tuy nhiên những thiết bị này thường cồng kềnh, thiếu bền

vững, độ ổn định thấp. Từ những năm 1980, đã bắt đầu xuất hiện các sensor ảnh trong các camera như sensor CMOS, CCD. Các sensor này chứa một số lượng phần tử quang rọi rạc, hay các điểm ảnh (pixel), mỗi phần tử chứa thông tin liên quan đến độ sáng chiếu vào nó.

Độ phân giải của các sensor phụ thuộc vào số điểm ảnh trên nó. Số điểm ảnh của các sensor là đa dạng, từ thấp ( $32 \times 32$  điểm ảnh), đến trung bình ( $256 \times 256$  điểm ảnh) cho đến cao ( $640 \times 480$ ) hoặc cao hơn nữa là  $1280 \times 1024$  điểm ảnh.

Trong đề tài này, vì lý do về điều kiện kinh tế của sinh viên nên không thể đầu tư mua một camera mắc tiền như các camera nói trên nên nhóm đã dùng máy ảnh 7.2 Mp để chụp lại biển số xe sau đó về tiến hành tải ảnh lên để phân tích. Ngoài ra nhóm còn dùng Webcam thay cho Camera để nhận dạng trực tiếp các ảnh biển số xe mẫu, nhưng với chất lượng hình ảnh kém nên việc phân tích ảnh gặp nhiều khó khăn.

#### **1.4.2 Phần mềm lập trình xử lý ảnh**

Nhờ vào ứng dụng của Matlab nhóm nghiên cứu và lập trình chương trình xử lý ảnh biển số xe, sau đó kết quả sẽ được truyền đến chương trình giám sát của hệ thống nhà giữ xe ô tô tự động thông qua giao thức OPC để tiến hành các chu trình tiếp theo của hệ thống.

Giao thức OPC được sử dụng để kết Matlab và WinCC phục vụ cho việc trao đổi và truyền dữ liệu cho nhau. Giao thức OPC sẽ được trình bày cụ thể ở chương sau.

**OPC** được viết tắt là: **OLE (Object Linking and Embedding) for Process Control**. Nghĩa là liên kết và nhúng đối tượng cho quá trình điều khiển.

## CHƯƠNG 2

### PHẦN MỀM MATLAB

#### **2.1 Tổng quan về Matlab**

##### **2.1.1 Khái niệm về Matlab**

Matlab là một ngôn ngữ lập trình thực hành bậc cao được sử dụng để giải các bài toán về kỹ thuật. Matlab tích hợp việc tính toán, thể hiện kết quả, cho phép lập trình, giao diện làm việc rất dễ dàng cho người sử dụng có thể có được những ứng dụng sau đây:

- Sử dụng các hàm có sẵn trong thư viện, các phép tính toán học thông thường.
- Cho phép lập trình tạo ra những ứng dụng mới.
- Cho phép mô phỏng các mô hình thực tế.
- Phân tích, khảo sát và hiển thị dữ liệu.
- Hỗ trợ công cụ đồ họa cực mạnh.
- Cho phép phát triển, giao tiếp với một số phần mềm khác như C++, Fortran, VB, WinCC...

##### **2.1.2 Cấu trúc dữ liệu của Matlab**

Matlab là một hệ thống tương giao, các phân tử dữ liệu là một mảng (không đòi hỏi về kích thước). Chúng cho phép giải quyết vấn đề liên quan đến lập trình bằng máy tính, đặc biệt sử dụng các phép tính bằng ma trận hoặc vector.

**Matlab có các kiểu dữ liệu được liệt kê sau đây:**

- Kiểu đơn Single: kiểu này có lợi về bộ nhớ dữ liệu vì nó đòi hỏi ít byte nhớ hơn, kiểu dữ liệu này không được sử dụng trong các phép toán học, độ chính xác kém hơn.
- Kiểu Double: là kiểu thông dụng nhất của các biến trong Matlab.

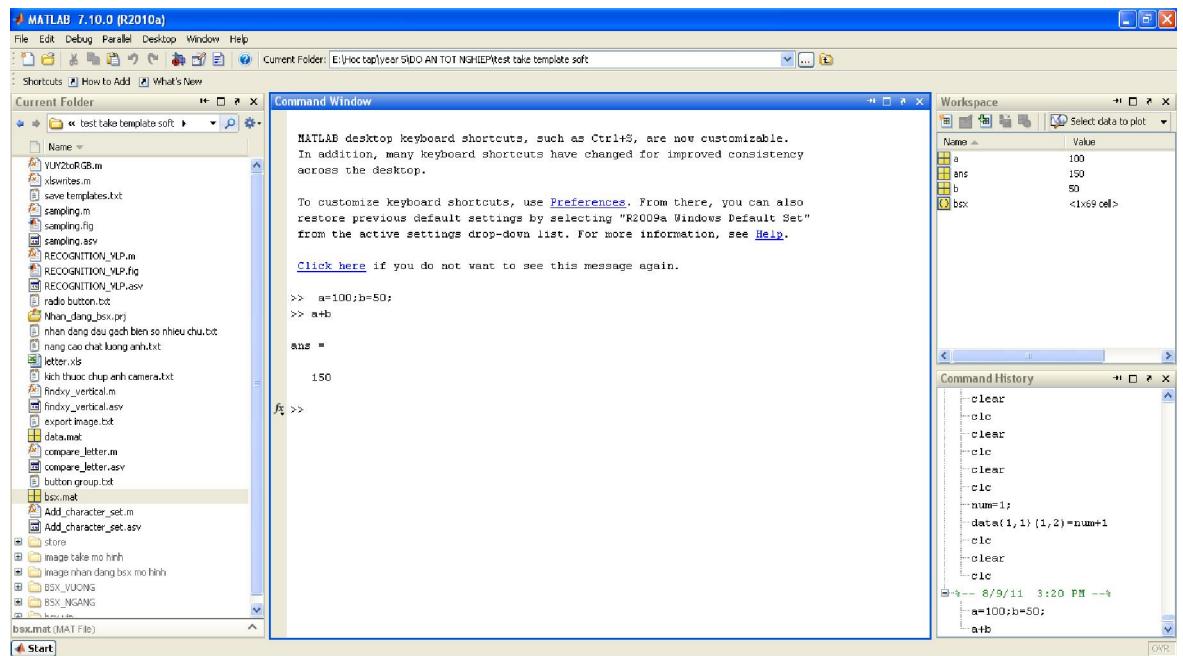
- Kiểu Sparse: được dùng để lưu trữ dữ liệu mà phần lớn là phần tử zero. Khi sử dụng kiểu dữ liệu này sẽ giảm được dung lượng bộ nhớ lưu trữ.
- Kiểu int8, uint8, int16...
- Kiểu Char
- Kiểu Cell
- Kiểu Structure

Trong Matlab kiểu dữ liệu Double là kiểu mặc định sử dụng trong các phép tính số học.

### 2.1.3 Các Toolbox của Matlab

Công cụ này được Matlab cung cấp cho phép bạn ứng dụng các kỹ thuật để phân tích, thiết kế, mô phỏng các mô hình. Ta có thể tìm thấy một số Toolbox như: Simulink, Image Acquisition, Image Processing, Neural Network, OPC toolbox...

### 2.1.4 Giao diện Matlab



Hình 2.1 Giao diện sử dụng của Matlab

**❖ Giao diện sử dụng của Matlab gồm có 5 vùng chính:**

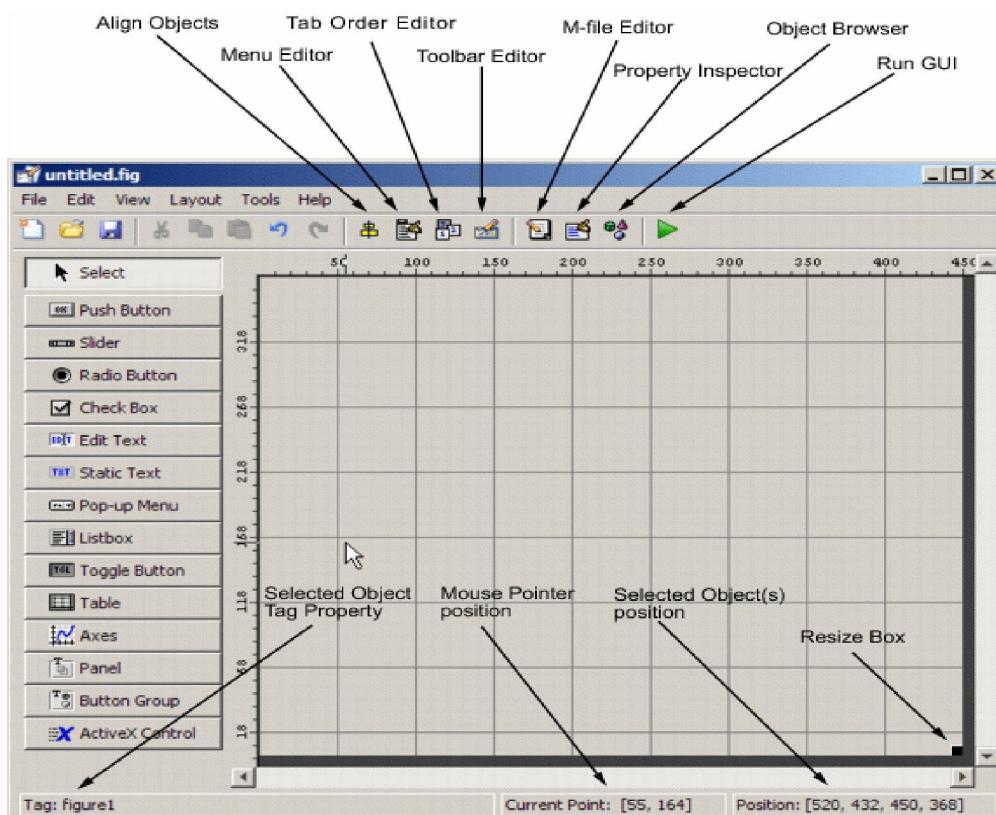
- Trình điều khiển trung tâm của Matlab dùng để thực hiện các thao tác như: tạo hoặc mở các chương trình của matlab, các menu phím tắt ...
- Khung “Current Folder”: phía bên tay trái (hình 2.1) chứa các tập tin trong thư mục hiện hành.
- Khung “Command Window”: là không gian thực hiện các lệnh trong matlab như gán, thực hiện phép tính...
- Khung “Work Space”: thể hiện các biến và kiểu dữ liệu sử dụng.
- Khung “Command History”: thể hiện các lệnh mà chúng ta đã sử dụng. Khi cần thiết có thể tìm kiếm để tái sử dụng lại cho nhanh.

## 2.2 GUI trong Matlab

### 2.2.1 Giới thiệu

GUI là một công cụ tạo giao diện tương tác người sử dụng được Matlab hỗ trợ, bao gồm các nút nhấn, hộp liệt kê, thanh trượt, menu ... giúp cho người sử dụng có thể thiết kế một giao diện tương tác người sử dụng dễ dàng và hiệu quả.

Để mở giao diện thiết kế GUI, trên khung giao diện chính của Matlab các bạn thực hiện thao tác : File>New>GUI. Sau khi hiện ra hộp thoại các bạn chọn Tab “Create New GUI”. Hiện ra trình soạn thảo GUI (Hình 2.2)

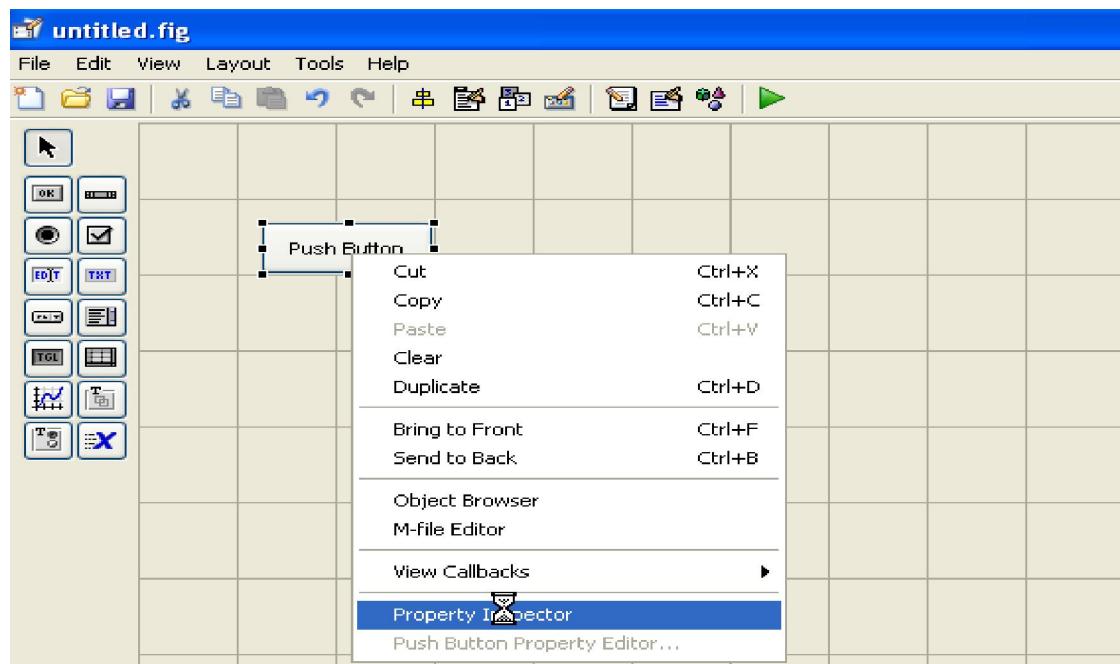


Hình 2.2 Giao diện thiết kế của GUI

### 2.2.2 Thao tác trên giao diện thiết kế GUI

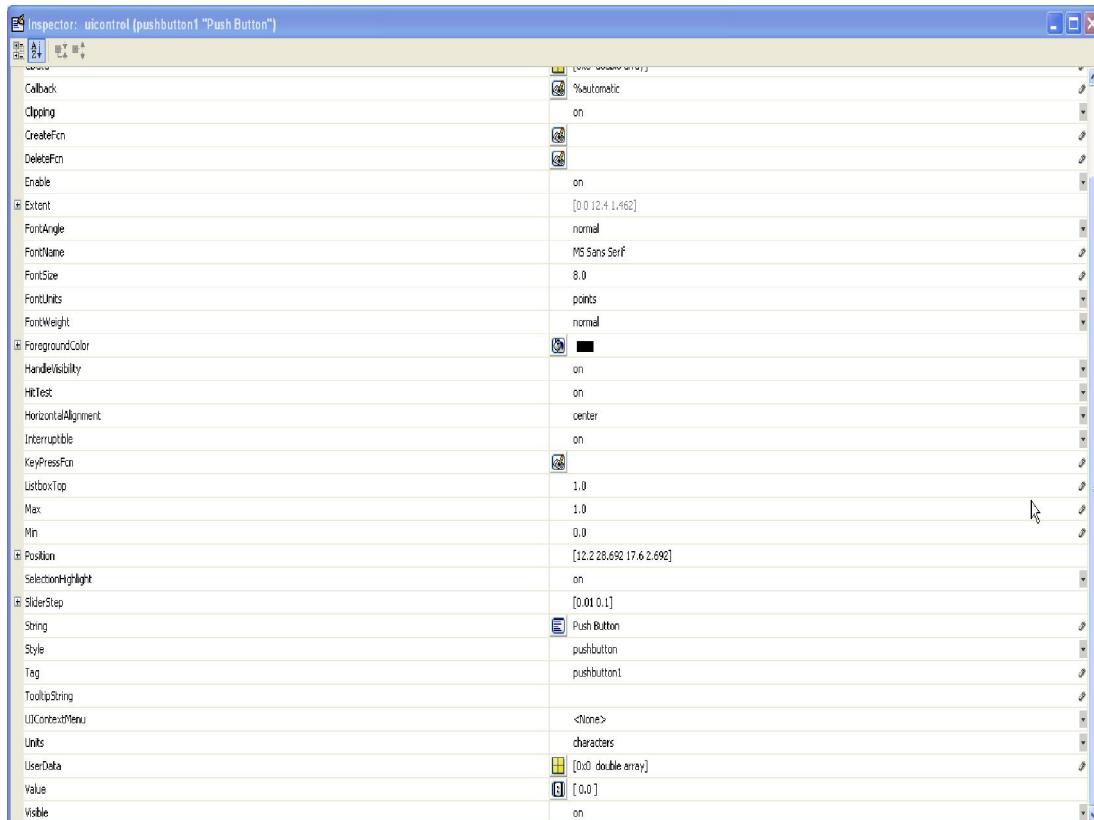
Công cụ GUI cực kỳ đơn giản và dễ sử dụng. Các bạn chỉ cần dùng chuột kéo thả các nút chức năng cần thiết kế. Sau đó chỉnh kích thước sao cho phù hợp.

Phần quan trọng để lập trình ứng dụng được cho các nút chức năng các bạn cần chú ý đến thuộc tính của nó, để vào thuộc tính các bạn chỉ cần đúp chuột vào nút chức năng bạn thiết kế hoặc thực hiện như hình sau đây: (Hình 2.3)



Hình 2.3 Thao tác chọn bảng thuộc tính của đối tượng

Nhìn ở bảng bên dưới có rất nhiều thuộc tính để chỉnh sửa như phông chữ, màu sắc... Hai thuộc tính cần chú ý là tên của đối tượng “String” và biến của đối tượng “Tag” cho đối tượng để dễ quản lý trong lập trình. (Hình 2.4)



Hình 2.4 Bảng thuộc tính của đối tượng

Sau khi tùy chỉnh các thuộc tính, thoát khỏi bảng thuộc tính và tiến hành lưu lại trên khung GUI. Khi đó phần mềm sẽ tự động tạo ra tệp m-file ngay trên thư mục đang lưu trữ để phục vụ cho mục đích lập trình ứng dụng.

Hình 2.5 Giao diện lập trình M-file của GUI

Sau khi lập trình điều khiển ứng dụng cho đối tượng, lưu lại và run-time để kiểm tra kết quả làm được.

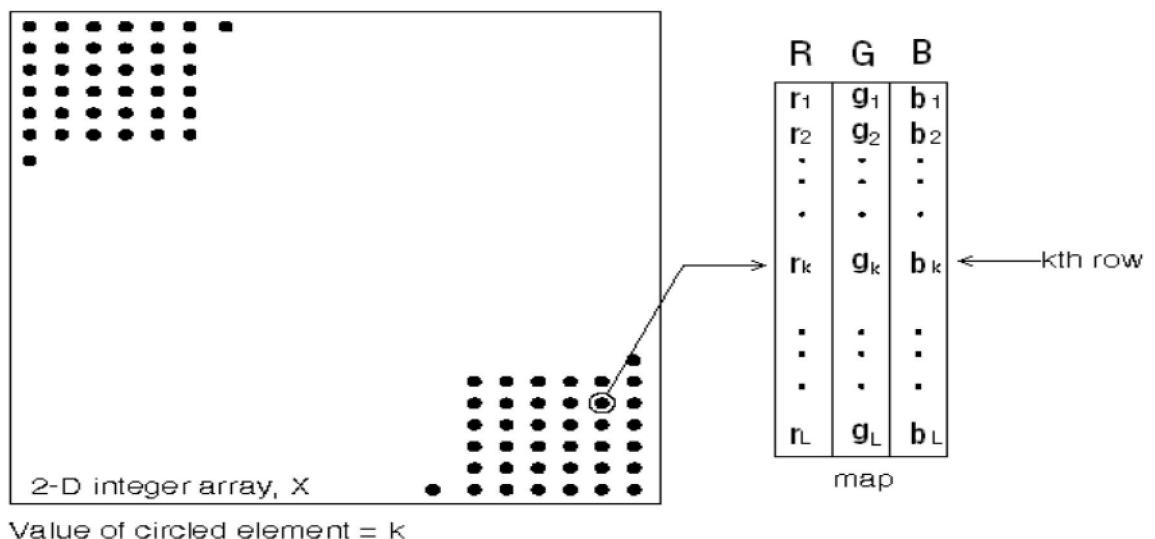
### **2.3 Toolbox xử lý ảnh Matlab**

### 2.3.1 Các loại kiểu dữ liệu ảnh trong Matlab

## ➤ **Ành Index**

Ảnh được biểu diễn bởi hai ma trận, một ma trận dữ liệu ảnh X và một ma trận màu (còn gọi là bản đồ màu) map. Ma trận dữ liệu có thể thuộc kiểu uint8, uint16 hoặc double. Ma trận màu là một ma trận kích thước  $m \times 3$  gồm các thành phần thuộc kiểu double có giá trị trong khoảng [0 1]. Mỗi hàng của ma trận xác định thành phần red, green, blue của một màu trong tổng số  $m$  màu được sử dụng trong ảnh. Giá trị của một phần tử

trong ma trận dữ liệu ảnh cho biết màu của điểm ảnh đó nằm ở hàng nào trong ma trận màu.



Hình 2.6 Biểu diễn dữ liệu ảnh Index

### ➤ Ảnh greyscale

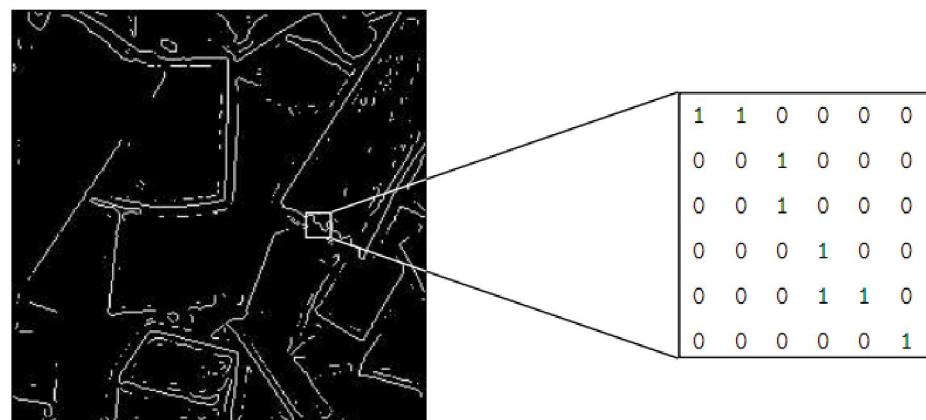
Mỗi ảnh được biểu diễn bởi một ma trận hai chiều, trong đó giá trị của mỗi phần tử cho biết độ sáng (hay mức xám) của điểm ảnh đó. Ma trận này có thể một trong các kiểu uint8, uint16 hoặc double. Ảnh biểu diễn theo kiểu này còn gọi là ảnh trắng đen.



Hình 2.7 Ảnh Greyscale

### ➤ Ảnh nhị phân

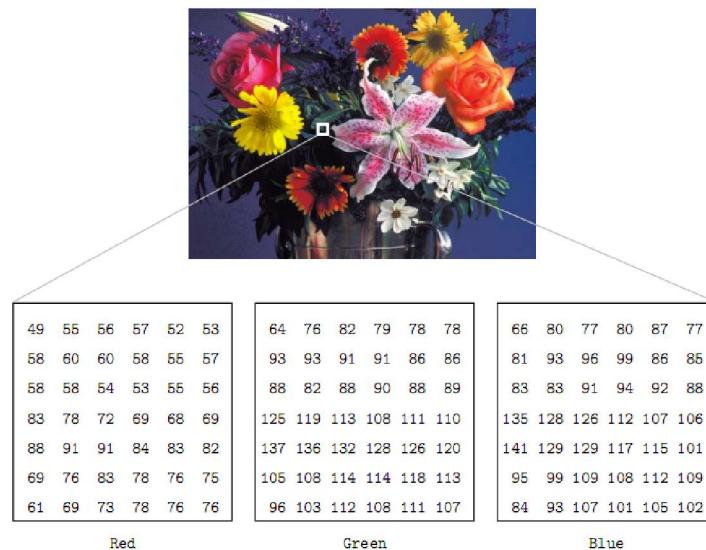
Ảnh được biểu diễn bởi một ma trận hai chiều thuộc kiểu logical. Mỗi điểm ảnh chỉ có thể nhận một trong hai giá trị là 0 (đen) hoặc 1 (trắng)



Hình 2.8 Ảnh nhị phân

### ➤ Ảnh RGB

Còn gọi là ảnh “truecolor” do tính trung thực của nó. Ảnh này được biểu diễn bởi một ma trận ba chiều kích thước  $m \times n \times 3$ , với  $m \times n$  là kích thước ảnh theo pixels. Ma trận này định nghĩa các thành phần màu red, green, blue cho mỗi điểm ảnh, các phần tử của nó có thể thuộc kiểu uint8, uint16 hoặc double.



Hình 2.9 Ảnh màu thực RGB

### 2.3.2 Các hàm xử lý ảnh cơ bản của Matlab

#### ➤ Đọc và ghi dữ liệu ảnh

- Hàm **imread** đọc các file ảnh với bất kỳ các định dạng ảnh đã biết hiện nay và lưu lại dưới dạng một ma trận biểu diễn ảnh trong Matlab.

Cú pháp : A=imread(filename,fmt)

Với fmt là định dạng của ảnh.

- Hàm **imwrite** cho phép lưu một ảnh biểu diễn bằng một ma trận trong Matlab thành một file ảnh dưới một trong các định dạng đã biết.

Cú pháp : imwrite(A,filename,fmt)

Với A là nguồn dữ liệu để xuất ảnh

- Hàm **imfinfo** dùng để xem các thông số của một file ảnh nào đó.

Cú pháp : imfinfo(filename,fmt)

Các thông tin được cung cấp bởi hàm imfinfo là : filename, filemoddate, filesize, format, formatversion, width, height, bitdepth, colortype.

#### ➤ Chuyển đổi giữa các kiểu dữ liệu

Matlab cung cấp sẵn các hàm thực hiện chuyển đổi kiểu cho các ma trận biểu diễn ảnh, bao gồm : im2double, im2uint8 và im2uint16. Tuy nhiên, khi thực hiện chuyển đổi giữa các dữ liệu ảnh cần lưu ý một số điều sau:

- Khi chuyển từ một kiểu dữ liệu dùng nhiều bit sang một kiểu dữ liệu dùng ít bit hơn thì một số thông tin chi tiết về bức ảnh ban đầu sẽ bị mất.
- Không phải lúc nào cũng có thể chuyển đổi kiểu dữ liệu đối với kiểu ảnh indexed, vì các giá trị của ma trận ảnh xác định một địa chỉ trong bản đồ màu chứ không phải là giá trị màu, do đó không thể lượng tử hóa được.

### ➤ Chuyển đổi giữa các định dạng ảnh

Matlab cung cấp các hàm chuyển đổi qua lại giữa các định dạng của ảnh như: Dither, Gray2ind, GraySlice, Im2bw, In2gray, Ind2rgb, Mat2gray, Rgb2gray, Rgb2ind.

### ➤ Các hàm hiển thị ảnh trong Matlab

Matlab cung cấp hai hàm hiển thị cơ bản là image và imagesc. Ngoài ra cũng có hai hàm hiển thị ảnh khác là imview và imshow.

- Hàm **image(X,Y,C)** hiển thị hình ảnh biểu diễn bởi ma trận C kích thước M x N lên trực tọa độ hiện hành. X, Y là các vector xác định vị trí các pixel C(1,1) và C(M,N) trong hệ trực hiện hành.
- Hàm **imagesc** có chức năng tương tự như hàm image, ngoại trừ việc dữ liệu ảnh sẽ được co giãn để sử dụng toàn bộ bản đồ màu hiện hành.
- Hàm **imview** cho phép hiển thị hình ảnh trên một cửa sổ riêng, nền Java, gọi là image Viewer. Image Viewer cung cấp các công cụ dò tìm và xác định các giá trị pixel một cách linh hoạt.
- Hàm **imshow** cũng tạo một đối tượng đồ họa thuộc loại image và hiển thị ảnh trên một figure. Hàm imshow sẽ tự động thiết lập các giá trị của các đối tượng image, axes và figure để thể hiện hình ảnh.

### ➤ Phép quay ảnh

Để thực hiện phép quay ảnh, ta có thể sử dụng hàm **imrotate**. Ngoài hai thông số cơ bản là ảnh gốc và góc quay, người sử dụng cũng có thể xác định phương pháp nội suy sẽ dùng và kích thước của ảnh mới.

Cú pháp : b=imrotate(a,angle,method,Bbox)

Với method là kiểu quay do matlab hỗ trợ, Bbox là thuộc tính đặc trưng cho việc trả về kích cỡ của ảnh. Để biết rõ thêm về các thông số này ta sử dụng help của Matlab.

### ➤ Trích xuất ảnh

Khi cần trích xuất một phần ảnh gốc, ta dùng hàm **imcrop**.

- Xác định cụ thể vị trí của phần ảnh cần trích xuất (dạng hình chữ nhật)

Cú pháp :  $x2=imcrop(x, map, rect)$  % indexed

$x2=imcrop(a, rect)$  % grayscale or RGB

trong đó  $rect=[Xmin\ Ymin\ width\ height]$

- Sử dụng mouse để chọn phần ảnh cần trích xuất. Ta không cần cung cấp thông số rect, khi thực hiện hàm này, con trỏ sẽ chuyển sang dạn chữ thập, người dùng sẽ kéo chuột để chọn phần ảnh cần trích xuất sau đó thả chuột.

### ➤ Chuyển đổi Hough

Được dùng để xác định các thông số cần thiết cho việc tìm đường thẳng chứa trong ảnh số 2 chiều.

Cú pháp:  $[H, theta, rho] = hough(BW)$

Với:

- BW là ảnh đầu vào.
- $(H, theta, rho)$  là các giá trị trả về của lệnh chuyển đổi hough. Với  $H$  là một ma trận chứa thông số chuyển đổi Hough. Theta (đơn vị là độ), rho là một mảng giá trị chứa thông số chuyển đổi hough.
- $(H, theta, rho)$  là thông số đầu vào cho hàm Houghlines (Matlab) có chức năng tìm kiếm các đường thẳng trong ảnh.

### ➤ Hàm Houghpeaks

Hàm tính toán số điểm có giá trị cao nhất trong ma trận  $H$  của chuyển đổi Hough.

Cú pháp:  $peaks = houghpeaks(H, numpeaks)$

Với:

- Peaks là thông số dữ liệu đầu vào của hàm Houghlines.
- $H$  là ma trận có được sau khi thực hiện chuyển đổi Hough

- Numpeaks là số đỉnh cần xác định do người lập trình quy định.

### ➤ **Hàm Houghlines**

Tìm kiếm tọa độ của các đường thẳng chứa trong ảnh đầu vào.

Cú pháp: lines = houghlines(BW, theta, rho, peaks)

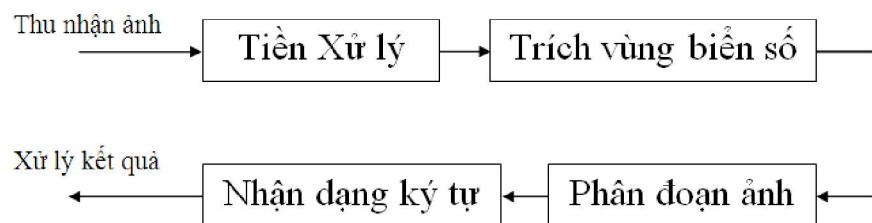
Với:

- Lines: là ma trận chứa tọa độ của các đường thẳng tìm được trong ảnh.
- (BW, theta, rho, peaks) là các thông số đầu vào cho hàm houghlines, các thông số này đã tìm được thông qua chuyển đổi hough và hàm houghpeaks.

## CHƯƠNG 3

### QUY TRÌNH VÀ THUẬT TOÁN NHẬN DẠNG BIỂN SỐ XE

#### 3.1 Quy trình nhận dạng biển số xe

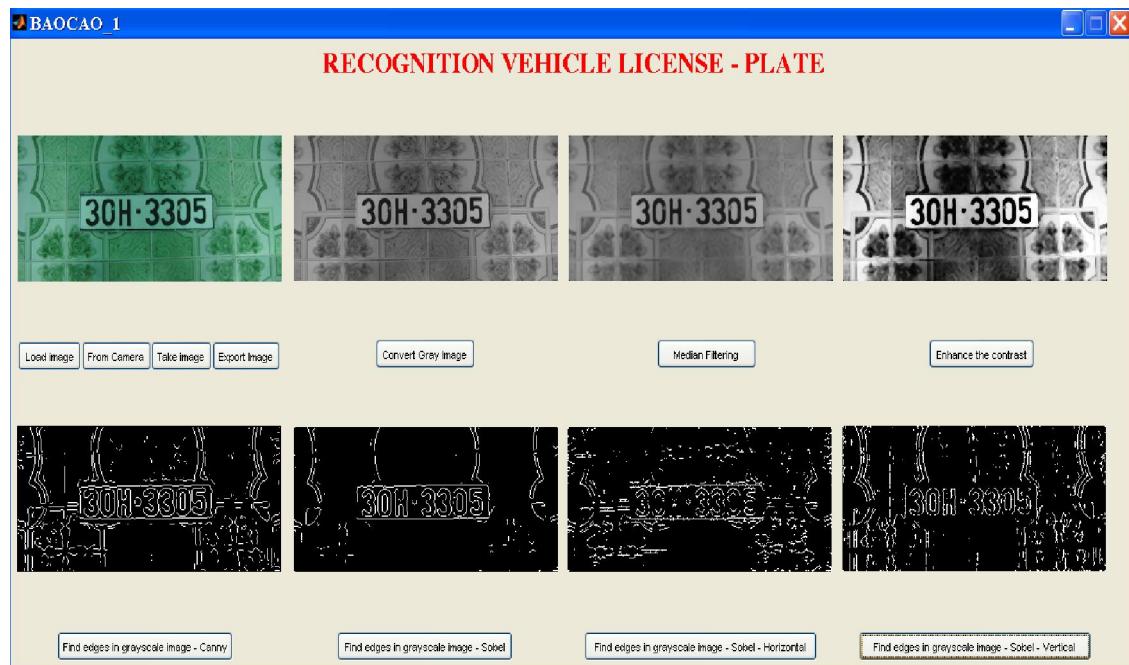


Ảnh được chụp từ camera được xử lý nhằm nâng cao chất lượng ảnh, đường cạnh trong ảnh, lọc nhiễu. Ảnh sau khi tiền xử lý được đưa vào khâu trích vùng biển số, tại đây từ một ảnh chụp toàn bộ xe chương trình nhận dạng sẽ trích ra chính xác vùng chứa biển số xe. Kế tiếp là phân đoạn ảnh sẽ làm nhiệm vụ tách từng ký tự có trong biển số và nhận dạng kết quả. Kết quả sẽ được xử lý và lưu trữ để phục vụ cho các quá trình tiếp theo của hệ thống.

#### 3.2 Thuật toán nhận dạng biển số xe

##### 3.2.1 Tiền xử lý

Để nâng cao chất lượng ảnh và làm giàu các đường cạnh của ảnh đầu vào ta chuyển ảnh sang ảnh xám tiến hành cần bằng lược đồ xám, dùng các bộ lọc để lọc nhiễu. Sau đó dùng bộ lọc Sobel để xác định các đường nét cạnh có trong ảnh để chuẩn bị cho giai đoạn kế tiếp tự động trích vùng chứa biển số xe.



Hình 3.1 Các khâu tiền xử lý ảnh để nâng cao chất lượng ảnh đầu vào

### 3.2.2 Tự động trích vùng bảng số xe

Ở Việt Nam thông thường có hai loại biển số xe là biển số 1 hàng ký tự (Hình 3.2a) và biển số có 2 hàng ký tự (Hình 3.2b)



a. 2 hàng ký tự



b. 1 hàng ký tự

Hình 3.2 Các loại hình dáng biển số xe ở Việt Nam

#### Phương pháp trích vùng biển số xe:

- Xác định tọa độ các đường thẳng chứa trong biển số xe dùng hàm chuyên đổi Hough, Houghpeaks và Houghlines của Matlab.
- Dựa tỷ lệ kích thước chiều dài và chiều rộng của từng loại bảng số để xác định chính xác vùng chứa biển số.

❖ **Với biển số một hàng ký tự:**

Kích thước chiều rộng và chiều cao thực tế sẽ là (47.5\*11 cm), tỷ lệ Rộng/Cao=4.32. Vì trong quá trình phân tích cạnh thì chiều dài và chiều rộng phân tích được không hoàn toàn đúng kích thước thực (có khả năng ngắn hơn hoặc dài hơn) nên ta chọn tỷ lệ kích thước tính được nằm trong khoảng (3.5 ; 5) thì đối tượng cặp cạnh ta tìm được là biển số xe.



Hình 3.3 Tự động trích vùng biển số xe kiểu 1 hàng ký tự

❖ **Đối với biển số hai hàng ký tự:**

Kích thước chiều rộng và chiều cao thực tế sẽ là (28\*21 cm), tỷ lệ Rộng/Cao=1.33. Vì thế ta chọn tỷ lệ kích thước nhận dạng nằm trong khoảng (1.2, ; 1.8) thì đối tượng cặp cạnh ta tìm được có khả năng cao nhất là biển số xe. Sau đó ta trích ra vùng chứa bảng số xe dựa trên các kích thước đã tính được.

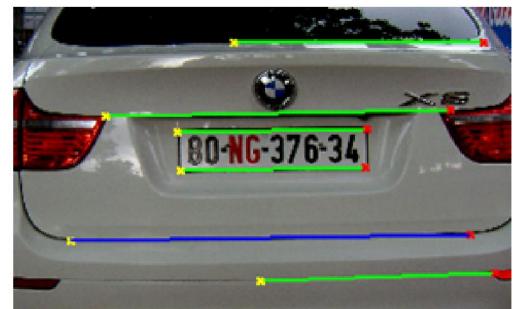


Hình 3.4 Tự động trích vùng biển số xe kiểu 2 hàng ký tự

- ☞ Để tốc độ trích vùng biển số nhanh hơn khi trích vùng biển số 1 hàng ký tự ưu tiên sử dụng bộ lọc với đặc tính xác định đường cạnh theo phương nằm ngang. Và ngược lại với biển số có 2 hàng ký tự ưu tiên sử dụng bộ lọc xác định đường cạnh theo phương thẳng đứng.



a. Theo phương đứng



b. Theo phương ngang

Hình 3.5 Phương pháp xác định cạnh theo phương ngang hoặc đứng

### 3.2.3 Chỉnh độ nghiêng

Sau khi trích được chính xác vùng biển số ta thực hiện việc chỉnh độ nghiêng.



Hình 3.6 Biển số có độ nghiêng

#### Phương pháp:

Phương pháp chỉnh độ nghiêng khá đơn giản chỉ cần ta xác định được giá trị góc nghiêng và chiều nghiêng của biên ảnh hiện tại so với phương chuẩn thẳng đứng hoặc nằm ngang. Thực hiện xoay ảnh ngược lại với giá trị góc tính được.

- Chỉnh độ nghiêng theo phương thẳng đứng (biển số 2 hàng ký tự)

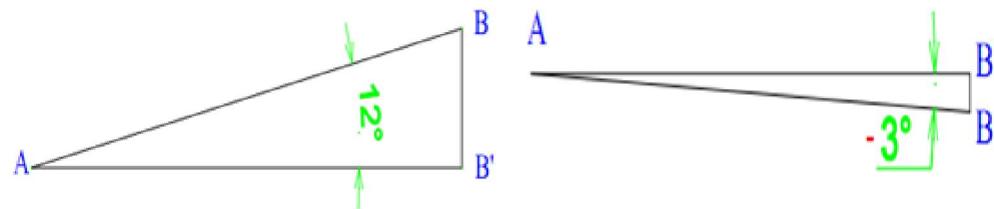
Xác định góc lệch và chiều quay lệch giữa các cạnh phương đứng của biển số AB với phương thẳng đứng AB'. Thực hiện xoay ảnh với giá trị góc và ngược lại với chiều lệch vừa tính toán được.



Hình 3.7 Chỉnh độ nghiêng theo phương thẳng đứng

- Chỉnh độ nghiêng theo phương nằm ngang (biển số 1 hàng ký tự)

Xác định góc lệch và chiều quay lệch giữa các cạnh phương đứng của biển số AB với phương thẳng đứng AB'. Thực hiện xoay ảnh với giá trị góc và ngược lại với chiều lệch vừa tính toán được.



Hình 3.8 Chỉnh độ nghiêng theo phương nằm ngang

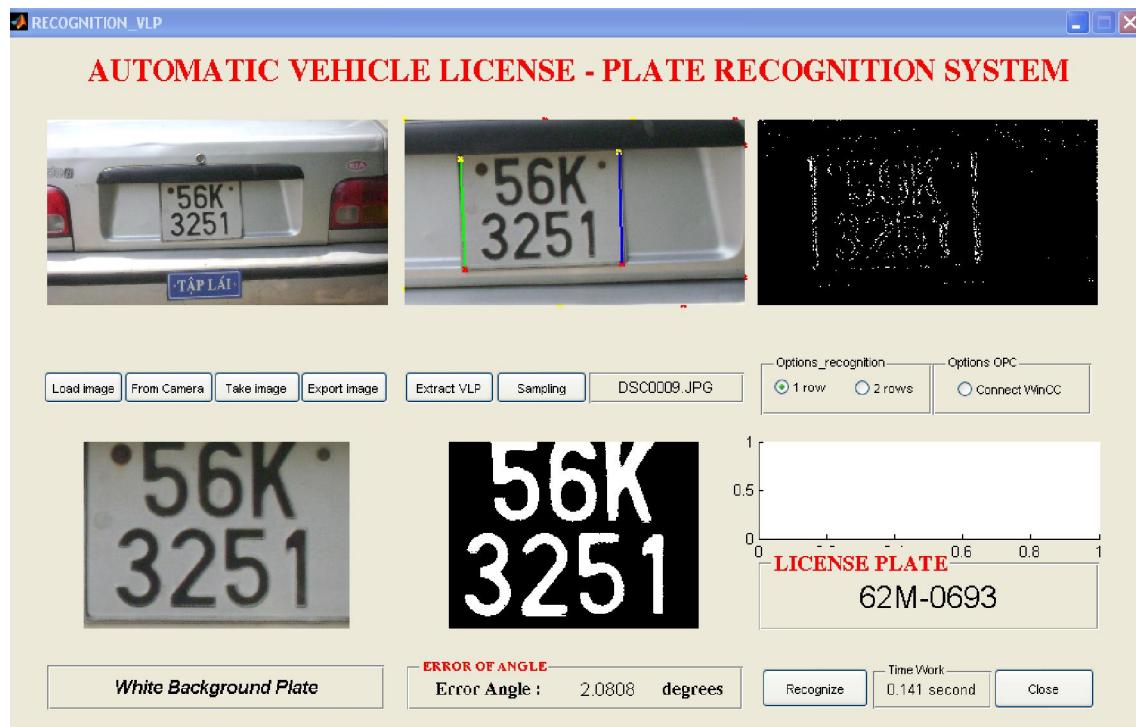
☞ **Hình ảnh kết quả của chương trình chỉnh độ nghiêng:**

- ❖ Biển số 1 hàng ký tự (Hình 3.9)



Hình 3.9 Chỉnh độ nghiêng cho biển số 1 hàng ký tự theo phương ngang

❖ Biển số 2 hàng ký tự (Hình 3.10)



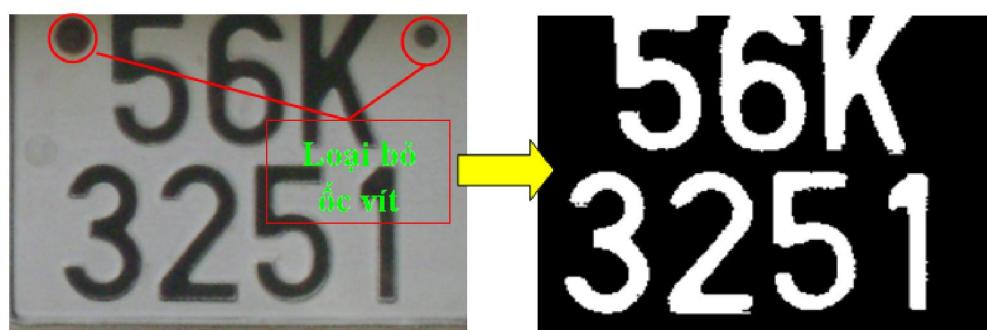
Hình 3.10 Chính độ nghiêng cho biển số 2 hàng ký tự theo phương thẳng đứng

### 3.2.4 Phân đoạn các ký tự có trong vùng biển số

Quá trình phân đoạn các ký tự nhằm mục đích tách riêng từng ký tự có trong biển số để tiến hành nhận dạng.

#### Quy trình:

- Chuyển vùng biển số trích được sang dạng ảnh nhị phân. (Hình 3.11)
- Xóa bỏ các phần tử thừa không phải là ký tự biển số như: ốc vít, gạch phân cách, các vết bẩn ... (Hình 3.11)



Hình 3.11 Chuyển sang nhị phân và loại bỏ ốc vít trên biển số

- Tách từng hàng và từng ký tự có trên biển số để tiến hành nhận dạng. (Hình 3.12)
 

Dựa vào đặc điểm ảnh nhị phân ta dễ dàng có thể tách được từng ký tự có trong biển số theo các bước sau:

  - Để tách từng hàng ký tự ta thực hiện quét dữ liệu điểm ảnh theo phương y (Hình 3.12) đến khi nào giá trị tổng các điểm ảnh theo phương x (Hình 3.12) bằng 0 thì khi đó ta đã trích được hàng ký tự đầu tiên có trong biển số.

Công thức:  $\sum_{n=1 \rightarrow k} f(x_n, y) = 0$  với  $y = 1 \rightarrow m$

☞ Trích được hàng ký tự trong biển số.

Với:

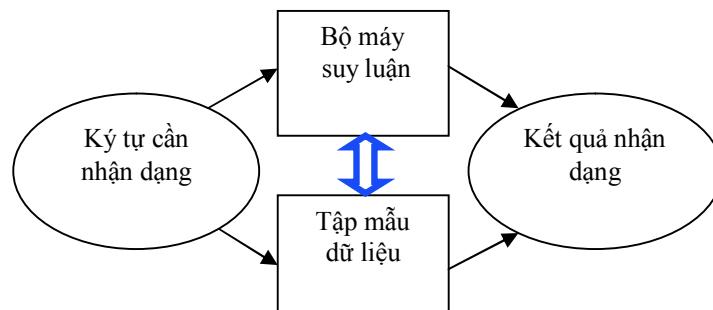
- $f(x, y)$  là tập ma trận ảnh.
- $k$  là giá trị độ dài của tập ảnh theo phương x.
- $m$  là giá trị độ dài của tập ảnh theo phương y.
- Để tách từng ký tự của biển số ta thực hiện quét dữ liệu của hàng ký tự vừa trích được theo phương x (Hình 3.12), tìm kiếm những vùng điểm ảnh có giá trị 1 liên tiếp nhau. Đánh dấu và trích từng vùng ta sẽ tách được từng ký tự trong biển số.



Hình 3.12 Phân đoạn từng ký tự có trong biển số xe

### 3.2.5 Nhận dạng ký tự

#### Sơ đồ tổng quan nhận dạng ký tự



Mỗi ký tự sau khi trích là đầu vào của khâu nhận dạng. Quá trình nhận dạng ký tự được thực hiện như sau:

- Tiến hành lấy mẫu ký tự có kích thước chuẩn là 42\*24 điểm ảnh để lưu tập mẫu chuẩn (mỗi điểm ảnh có giá trị 1 hoặc 0). Dữ liệu lấy mẫu bao gồm các mẫu chữ cái từ A – Z và chữ số từ 0 – 9. Tập mẫu được lưu dưới dạng tập tin dữ liệu của Matlab có dạng là “ \*.mat ”. (Hình 3.13)

	1	2	3	4	5	6	7	8	9	10
1	<42x24 logi...	<42x24 logi...	<42x24 logi...							
2	A	B	C	D	E	F	G	I	2	3
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										

Hình 3.13 Bảng dữ liệu mẫu lưu trữ trong Matlab

- Chuyển kích thước của mỗi ký tự được về kích thước chuẩn để nhận dạng là 42\*24 điểm ảnh.
- So sánh từng ký tự cần nhận dạng với cơ sở dữ liệu có trong tập mẫu chuẩn. Kết quả so sánh nào mà vị trí mẫu dữ liệu có tỷ lệ phần trăm giống cao nhất thì đó chính là kết quả nhận dạng.

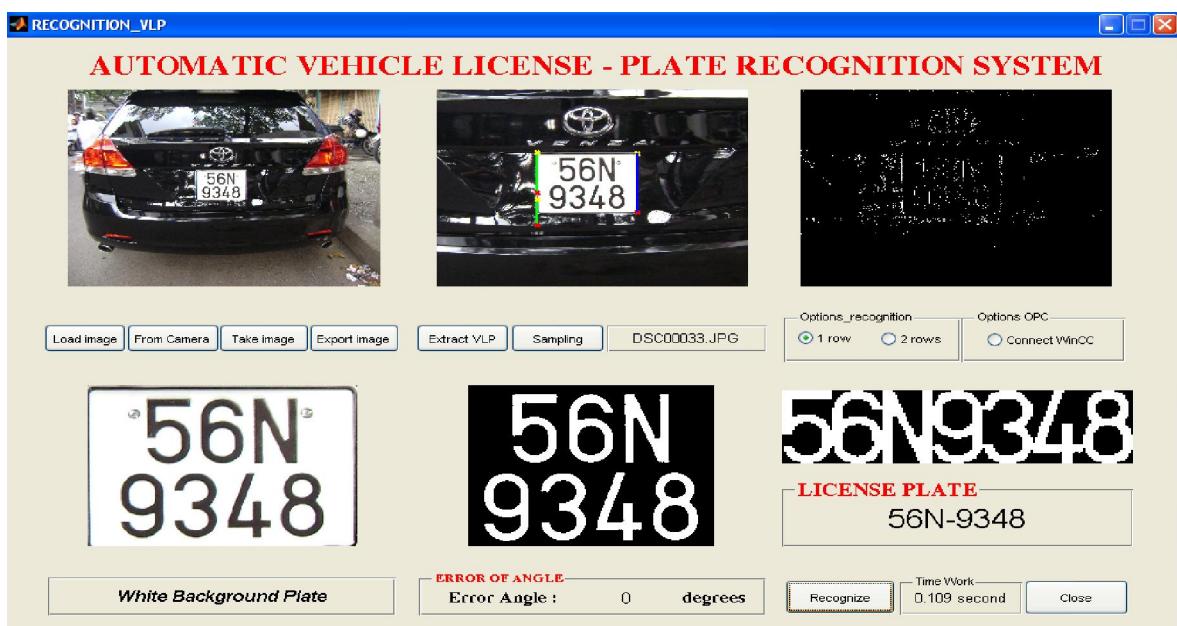
### 3.3 Hình ảnh kết quả nhận dạng

#### 3.3.1 Nhận dạng biển số 1 hàng ký tự (Hình 3.14)



Hình 3.14 Nhận dạng biển số 1 hàng ký tự

#### 3.3.2 Nhận dạng biển số 2 hàng ký tự (Hình 3.15)



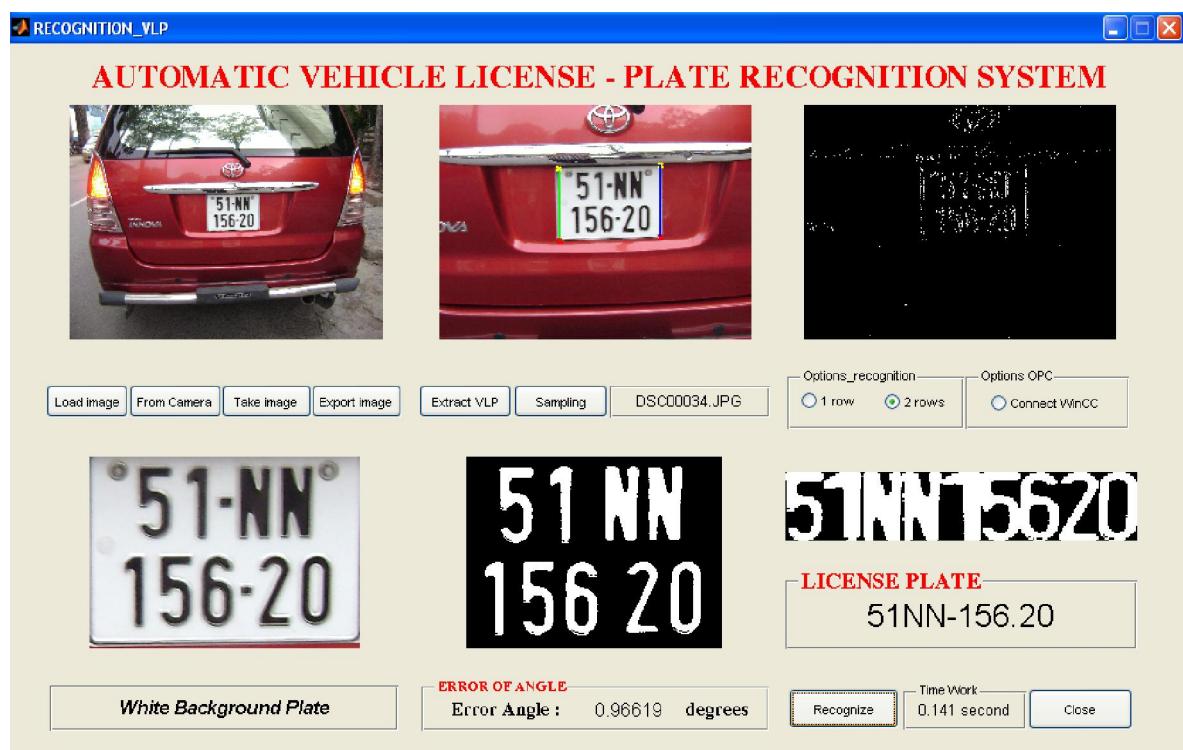
Hình 3.15 Các khâu nhận dạng biển số xe hoàn chỉnh

### 3.3.3 Nhận dạng biển số xanh với 2 hàng ký tự (Hình 3.16)



Hình 3.16 Các khâu nhận dạng biển số xe hoàn chỉnh

### 3.3.4 Nhận dạng biển số nhiều ký tự (Hình 3.16)



Hình 3.16 Nhận dạng biển số nhiều ký tự

## CHƯƠNG 4

### LIÊN KẾT GIỮA WINCC VÀ MATLAB THÔNG QUA OPC

#### 4.1 Tổng quan về giao thức OPC (Hình 4.1)

OPC: OLE (*Object Linking and Embedding*) FOR PROCESS CONTROL.

OPC: ĐƯỢC GỌI LÀ LIÊN KẾT VÀ NHÚNG ĐỐI TƯỢNG CHO ĐIỀU KHIỂN QUÁ TRÌNH.



Hình 4.1 Mô tả giao tiếp Matlab và WinCC qua OPC

OPC là giao thức dùng để liên kết các phần mềm, ứng dụng của các hãng khác nhau nhằm mục đích trao đổi dữ liệu qua lại cho nhau.

Nhiệm vụ của đồ án tốt nghiệp là thực hiện nhận dạng biển số xe bằng chương trình Matlab. Tín hiệu ngắn có xe từ WinCC truyền tới Matlab kích hoạt chương trình nhận dạng làm việc. Kết quả nhận dạng của Matlab truyền tới WinCC để thực hiện điều khiển, giám sát và lưu trữ dữ liệu cho hệ thống. Vì vậy OPC chính là công cụ trung gian làm cầu nối dữ liệu của hai phần mềm.

OPC WinCC cần được cấu hình và thiết lập các biến dữ liệu trao đổi. Bất cứ biến của OPC WinCC thay đổi giá trị thì các biến trong Matlab cũng sẽ thay đổi theo.

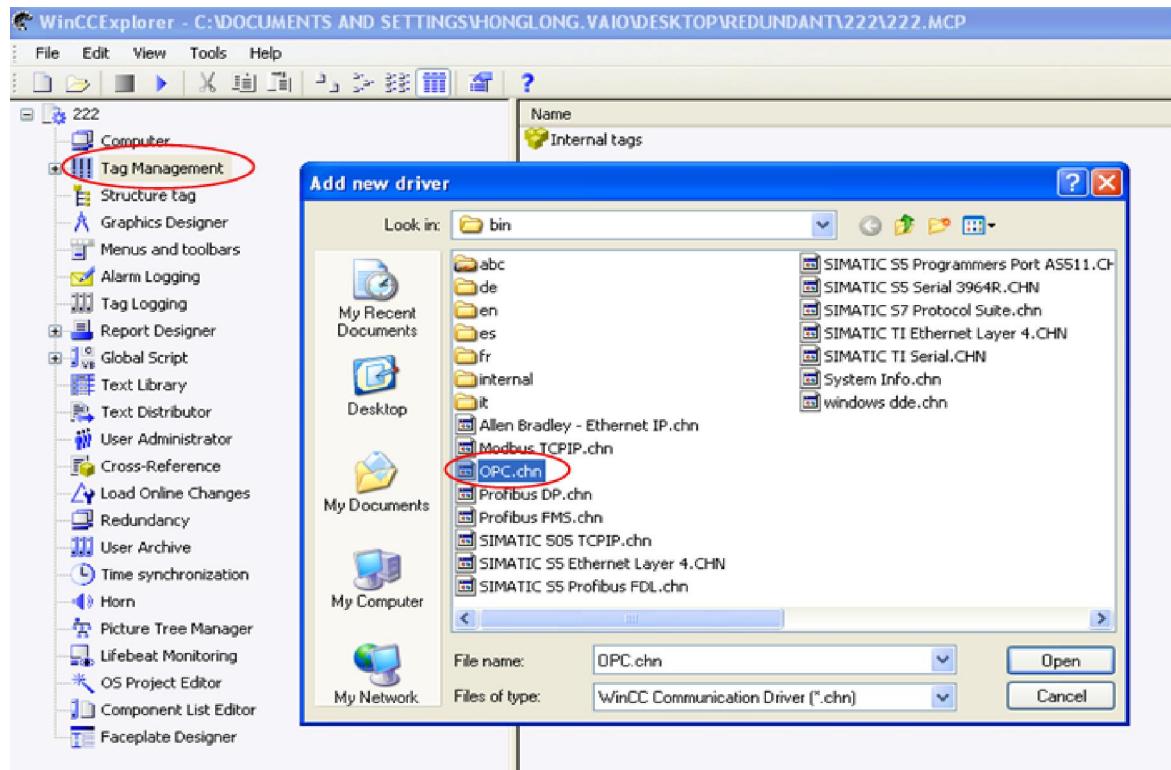
OPC Matlab cần được khai báo thông qua các lệnh được Matlab định nghĩa sẵn. Để truy xuất được dữ liệu từ WinCC cần thực hiện các hàm có chức đọc, ghi do Matlab hỗ trợ.

Để cấu hình cho OPC WinCC và OPC Matlab sẽ được trình bày rõ trong các mục kế tiếp.

## 4.2 OPC WinCC

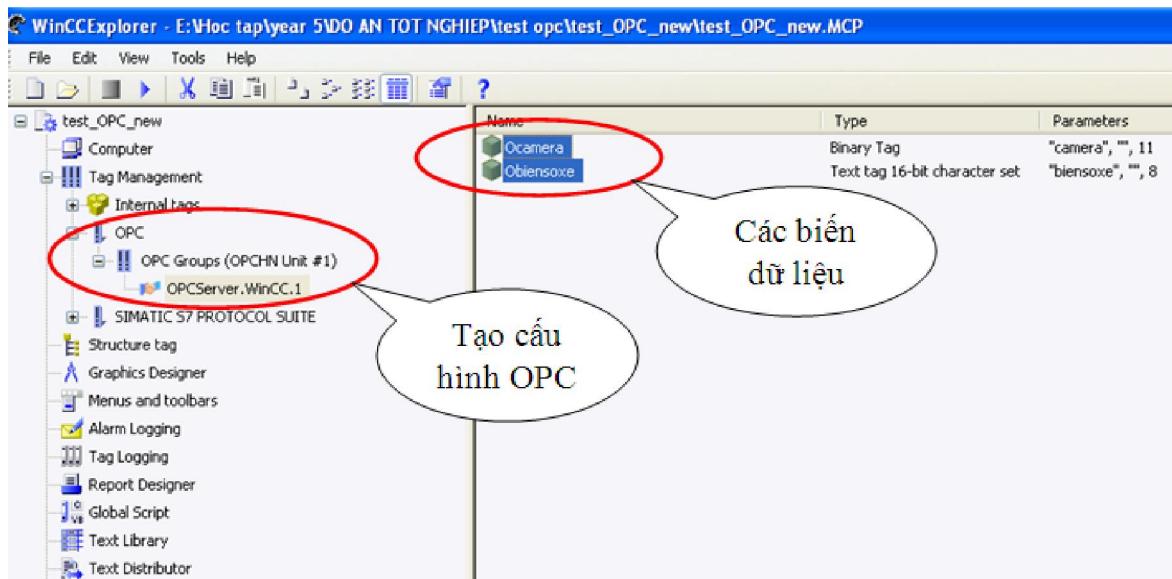
Để kết nối được với Matlab ta cần cấu hình chức năng OPC trong WinCC.

Add new driver OPC cho Tag Management (Hình 4.2).



Hình 4.2 Tạo OPC cho WinCC

Tạo OPC Groups, OPC Server WinCC và các biến dữ liệu (Hình 4.3)



Hình 4.3 Tạo OPC server và biến dữ liệu

### 4.3 OPC Matlab

#### 4.3.1 Tạo kết nối OPC trong Matlab

Khác với OPC của WinCC, để tạo OPC trong Matlab ta cần thực hiện các dòng lệnh trong như sau:

```
da = opcda('localhost','OPCServer.WinCC.1');
```

```
connect(da);
```

Lệnh “opcda” để tạo kết nối với OPC WinCC.

Lệnh “connect” để kết nối với OPC đã được tạo ra.

Với:

- “localhost” là cú pháp lệnh được định nghĩa sẵn.
- “OPCServer.WinCC.1” là tên OPC đã cấu hình trong phần mềm WinCC.

#### 4.3.2 Tạo nhóm biến dữ liệu liên kết với WinCC

Cú pháp lệnh:

```
nhom_bien = addgroup(da,'ExReadAsync');
biensoxe=additem(nhom_bien,'biensoxe');
camera=additem(nhom_bien,'camera');
```

Lệnh “addgroup” để tạo nhóm biến chung tên là “nhom\_bien”.

Lệnh “additem” tạo liên kết chuyển các biến WinCC thành biến Matlab.

Với:

- “ExReadAsync” là thuộc tính của lệnh addgroup.
- “biensoxe” là tên biến của WinCC.

#### 4.3.3 Đọc và ghi dữ liệu giữa Matlab và WinCC

Để có thể giao tiếp được dữ liệu từ WinCC và Matlab ta cần các lệnh đọc và ghi dữ liệu để truy xuất trạng thái hoặc ghi giá trị các biến của WinCC.

Cú pháp lệnh:

```
wincc_takeimage= read(camera);  
write(biensoxe,word);
```

Lệnh “read” để đọc giá trị biến của WinCC.

Lệnh “write” ghi giá trị cho biến WinCC.

Với:

- “word” là giá trị cần gán cho biến “biensoxe”.

## CHƯƠNG 5

### TỔNG QUAN VỀ NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG

#### 5.1 Tổng quan

Trong những năm gần đây với sự phát triển kinh tế của Việt Nam, số lượng phương tiện giao thông đã tăng một cách nhanh chóng. Phương tiện cá nhân tăng lên, đòi hỏi diện tích đất dành cho bãi đậu xe cũng phải tăng theo. Tuy nhiên, tại các thành phố lớn như Hồ Chí Minh, Hà Nội, Đà Nẵng, Cần Thơ việc đáp ứng yêu cầu đất này ngày càng tỏ ra không khả thi do giá trị đất đang tăng nhanh, và nhu cầu đất cho các mục đích quan trọng khác cũng đang thiếu. Giải pháp "chữa cháy" là sử dụng một phần diện tích mặt đường làm chỗ đậu xe chỉ mang tính chất tạm thời và cũng không đáp ứng đủ nhu cầu.

Hiện nay tại khu vực trung tâm thành phố Hồ Chí Minh, số ôtô thường xuyên dừng đậu, phần lớn dừng đậu trên vỉa hè, lòng đường gây cản trở giao thông (Hình 5.1).



Hình 5.1 Ô tô đậu dày đặc trên vỉa hè và lòng đường tại TP Hồ Chí Minh

Để giải quyết vấn đề chỗ đậu xe trong đô thị, nhiều nước trên thế giới sử dụng hệ thống nhà đậu xe nhiều tầng tự động, và đã trở thành phổ biến như Nhật Bản, Hàn Quốc, Ấn Độ, Singapore, Trung Quốc, Mỹ và các nước Châu Âu. Tại các

nước này có nhiều công ty chuyên kinh doanh bãi đậu ô tô nhiều loại, trong đó hệ thống đậu nhiều tầng tự động được sử dụng rất phổ biến. Các công ty sản xuất hệ thống đậu xe tự động là các nhà chế tạo, không trực tiếp kinh doanh bãi đậu xe mà chỉ cung cấp và lắp đặt thiết bị cho các nhà đầu tư. Ngoài ra, còn các hệ thống các công ty sản xuất các thiết bị phụ trợ như: hệ thống lấy vé tự động đọc thẻ, trả tiền tự động.

So với các bãi đỗ xe kiểu truyền thống, những lợi ích của một bãi đỗ xe tự động không chỉ là nhanh chóng và tiện lợi. Bạn sẽ không còn phải lo lắng chiếc xe của mình bị xô xát, va đụng, trầy xước bởi nơi đậu xe được cách ly hoàn toàn với bên ngoài.

Vào đầu năm 2006, Ủy Ban Nhân Dân thành phố Hồ Chí Minh đã mời các tổ chức, cá nhân tham gia đầu tư xây dựng 7 bãi đậu xe ngầm tại khu vực trung tâm thành phố. Bãi đậu xe ngầm thứ 8 tại công viên Lê Văn Tám được giao cho Công ty cổ phần Đầu tư phát triển không gian ngầm IUS làm chủ đầu tư.

#### **Các bãi đậu xe nói trên được thiết kế theo công nghệ xếp xe tự động gồm:**

- Bãi đậu xe ngầm tại số 116 Nguyễn Du có diện tích xây dựng trên mặt đất là  $560\text{ m}^2$ , diện tích xây dựng dưới mặt đất  $3.950\text{ m}^2$  gồm 8 tầng ngầm.
- Bãi đậu xe ngầm bờ sông Sài Gòn có tổng diện tích xây ngầm  $45.540\text{ m}^2$ , xây trên mặt đất  $900\text{ m}^2$ , gồm 5 tầng, có thể chứa 5.000 ô tô, 5.000 xe máy.
- Bãi tại sân bóng đá Tao Đàn có tổng diện tích  $40.000\text{ m}^2$  gồm 4 tầng ngầm.
- Bãi tại công viên Chi Lăng có tổng diện tích ngầm  $3.560\text{ m}^2$ , diện tích xây trên mặt đất  $210\text{ m}^2$  gồm 7 tầng.
- Bãi đậu tại công viên Bách Tùng Diệp gồm 5 tầng,  $5.200\text{ m}^2$  xây ngầm và  $300\text{ m}^2$  xây trên mặt đất.
- Bãi đậu ngầm tại công trường Lam Sơn gồm 8 tầng,  $2.110\text{ m}^2$  xây ngầm và  $230\text{ m}^2$  xây trên mặt đất.

- Bãi đậu xe tại sân vận động Hoa Lư gồm 5 tầng, tổng diện tích  $49.838m^2$ .
- Dự án bãi đậu xe tại công viên Lê Văn Tám gồm 5 tầng ngầm diện tích sàn là  $72.321 m^2$ , với mức đầu tư 1.748 tỉ đồng.

Tuy nhiên các bãi đỗ xe ngầm trên hiện nay vẫn chưa khởi công xây dựng. Vì thế để giải quyết nhu cầu đậu xe hiện nay ở khu vực trung tâm, TPHCM đang nghiên cứu xây dựng sớm các bãi đậu xe nổi nhiều tầng sử dụng băng thiết bị tự động trong khi chờ xây dựng các bãi đậu xe ngầm.

Đề nghị trên được ông Phan Thanh Nam, Giám đốc Công ty Đầu tư phát triển công nghiệp và vận tải (Tracodi) đưa ra trong buổi báo cáo về các dự án bãi đậu xe ở TPHCM ngày 23-3 tại Sở Giao thông vận tải TPHCM. Ông Nam cũng đề xuất với UBND TPHCM và Sở Giao thông vận tải nghiên cứu xây dựng 5 bãi đậu xe ở khu vực trung tâm (trừ các bãi đậu xe ngầm) nơi tập trung mật độ phương tiện cao, bao gồm:

- Công viên 23 – 9.
- Công trường Quách Thị Trang.
- Bến phà Thủ Thiêm.
- Điểm đậu ở đường Tôn Đức Thắng và Ngô Văn Năm.
- Điểm đậu ở cuối đường Nguyễn Huệ gần sông Sài Gòn.

## 5.2 Các giải pháp nhà giữ xe ô tô tự động

Sau đây là một số giải pháp đã được triển khai ở các nước khác:

### 5.2.1 Giải pháp “Xếp chồng” (Hình 5.2)

Sử dụng một hệ thống thủy lực để nâng tối đa bốn ôtô xếp cạnh nhau lên một tầm cao, để dành chỗ cho những xe khác ở bên dưới. Tuy nhiên, giải pháp này có hiệu quả kinh tế không cao, chỉ phù hợp với qui mô nhỏ (một hoặc vài hộ gia đình). Ưu điểm: hệ thống gọn nhẹ, dễ vận chuyển, lắp đặt nhanh.



Hình 5.2 Giải pháp đậu xe “xếp chồng”

### 5.2.2 Giải pháp “Nhà gửi xe nhiều tầng” (Hình 5.3)

Một nhà gửi xe nhiều tầng với các đường dốc để khách tự lái xe vào và ra khỏi khu gửi xe. Mức độ tự động hóa tương đối không cao (thường chỉ gồm các máy bán vé và hệ thống đóng/ mở cổng tự động). Giải pháp này tuy phổ biến nhưng chưa thỏa đáng lắm về mặt sử dụng không gian, cũng như đối với một số yêu cầu khác (an toàn cho xe và người, ô nhiễm vì khói thải từ ôtô...).



Hình 5.3 Mô hình nhà gửi xe nhiều tầng

### 5.2.3 Giải pháp “Nhà gửi xe tự động lộ thiên” (Hình 5.4)

Đây là một bước cải tiến so với giải pháp nhà gửi xe nhiều tầng. Sức chứa có thể tăng gấp hai lần so với kiểu drive-in parking có cùng diện tích sàn nhờ: loại bỏ các đường dốc và lối chạy ôtô trong nhà. Bố trí các xe sát nhau và thu hẹp khoảng cách giữa các tầng (Hình 5.4).



Hình 5.4 Giải pháp nhà gửi xe tự động

Sở dĩ làm được như vậy là nhờ các khâu nhận, bảo quản và trả xe hoàn toàn được tự động hóa. Việc gửi và nhận xe cũng vì thế mà đơn giản hơn trước. Tùy thuộc thiết kế, tiến trình có thể thay đổi đôi chút, nhưng nói chung khách gửi xe không phải tự mình (hoặc nhờ nhân viên bãi xe) lái xe vào khu vực đậu xe. Tại ngõ vào khách được nhận thẻ gửi xe. Sau khi cho thẻ vào máy đọc, khách lái xe đến đậu vào một cabin. Các màn hình video sẽ kiểm tra xe đã đậu đúng vị trí chưa.

Khách tắt máy và rời khỏi xe. Sau đó cửa cabin đóng lại, các máy tính ra lệnh cho hệ thống băng tải và thang máy đưa xe vào một vị trí đậu xe còn trống thích hợp ở các tầng. Thông tin về vị trí này được máy tính ghi nhận. Khi người khách trở lại, chỉ cần thanh toán ở quầy thu tiền và cho thẻ vào

máy đọc. Chiếc xe sẽ được hệ thống tự động định vị và giao trả ở ngõ ra trong thời gian ngắn nhất (khoảng 2 phút).

Ngoài ra có thể thiết kế theo dạng hình trụ để tăng thêm diện tích để xe và tiết kiệm được thời gian lấy xe ra vào (Hình 5.5).



Hình 5.5 Mô hình nhà gửi xe tự động hình trụ

➤ **Ưu điểm của giải pháp này là:**

Tận dụng tối đa không gian do ôtô được dịch chuyển bằng thang máy theo phương thẳng đứng, không cần các đường dốc và lối dành cho ôtô chạy. Khả năng chứa xe được nâng lên tối đa do không gian cần thiết để đậu ôtô có thể giảm, khoảng cách giữa các tầng để xe được thu hẹp.

Thuận tiện đối với khách gửi xe: toàn bộ việc gửi và nhận xe diễn ra ở tầng trệt với thời gian tối thiểu. Xe không bị va chạm và trầy xước.

Lợi ích đối với chủ sở hữu nhà xe: chi phí duy tu, bảo quản thấp do không có nhu cầu lái xe và đi lại trong khu vực đậu ôtô, nên có thể giảm chiếu sáng đến mức tối thiểu ở đây, do các xe đều được tắt máy nên yêu cầu thông gió để giải phóng khí thải độc hại là không đáng kể. Hơn nữa,

có thể ngăn chặn trình trạng trộm cắp, giảm đáng kể chi phí cho hệ thống an ninh. Hiệu quả đầu tư cao nhờ diện tích sử dụng đất thấp. Thời gian và chi phí xây dựng mô hình này ít hơn so với xây dựng mô hình nhà giữ xe tầng ngầm. Nhanh chóng đưa vào hoạt động, đáp ứng được nhu cầu đỗ xe hiện nay.

#### **5.2.4 Giải pháp “Nhà gửi xe tự động dạng ngầm” (Hình 5.6)**

Tương tự như giải pháp nhà để xe tự động hóa lộ thiên, nhưng sử dụng cấu trúc giếng tạo thành một hệ thống đậu ôtô nhiều tầng đặt ngầm dưới đất (Hình 5.6) và (Hình 5.7).



Hình 5.6 Mô hình nhà gửi xe tự động dạng ngầm

Khách lái xe vào điểm tiếp nhận, tắt máy và ra khỏi xe. Chiếc xe sẽ được chuyển vào khu đậu xe có cấu trúc dạng giếng bằng một thang máy quay  $360^{\circ}$  di chuyển theo phương thẳng đứng, để xếp vào một vị trí an toàn.

Các hệ thống phòng cháy, ngập nước, thông gió và bảo vệ đều được theo dõi bằng máy tính từ một trung tâm điều hành. Khách được phát một thẻ để nhận lại xe sau này và trả tiền gửi xe. Thời gian nhận xe tối đa là một phút.



Hình 5.7 Quá trình xây dựng một nhà gửi xe tự động dạng ngầm

➤ **Ưu điểm :**

Các ưu điểm của giải pháp này cũng tương tự như giải pháp nhà gửi xe tự động lộ thiên, ngoài ra có hai điểm khác nổi bật là:

- Tác động của hệ thống đối với môi trường xung quanh rất ít, do 80% diện tích xung quanh công trình có thể bố trí làm công viên, bảo đảm vẻ mỹ quan của thành phố (Hình 5.8).
- Có thể bố trí nhà gửi xe ngầm trong khu vực đông đúc, gần các tòa nhà hiện hữu, bên dưới hoặc gần các công trình mới.



Hình 5.8 Cảnh quan phía trên của nhà gửi xe tự động dạng ngầm

➤ **Nhược điểm:**

Nếu muốn xây dựng mô hình này cho các mặt bằng đã có sẵn công trình xây dựng và hơi khó khăn, vì phải tốn công sức và tiền của giải tỏa mặt bằng, đèn bù như thế thời gian để có thể thực thi được mô hình này sẽ kéo dài hơn. Vì thế mô hình này thích hợp với các công trình xây mới hoàn toàn, như thế sẽ tiết kiệm được khoảng đất trống trên bề mặt để dùng vào các mục đích khác nhau.

#### **5.2.5 Nhược điểm chung của hệ thống đỗ xe tự động gấp phai**

Những ưu điểm của hệ thống giữ xe tự động đã được đề cập thông qua các giải pháp nhà giữ xe đã trình bày phía trên. Ngoài những ưu điểm tích cực trên thì khi thực thi mô hình ta cũng cần phải quan tâm đến các nhược điểm mà hệ thống gấp phai nhằm đưa ra giải pháp khắc phục hiệu quả nhất. Một số nhược điểm như sau:

- Cần xem xét đến thời gian lấy xe, thời gian lấy xe tùy thuộc vào từng loại hệ thống. Đối với loại hệ thống 100 xe thông thường thì thời gian lấy xe lâu nhất khoảng gần 2 phút/xe, nhanh nhất 0,5 phút/xe, bình quân 1,5 phút/xe. Đối với các công trình nhà ở, siêu thị, các bãi xe công cộng... thì thông thường người sử dụng ít khi gửi xe hoặc lấy xe cùng một khoảng thời gian nên thời gian lấy trả xe 1,5 phút/xe không là vấn đề, thậm chí còn nhanh hơn so với bãi xe tự lái. Tuy nhiên đối với các công trình văn phòng, rạp hát, hội nghị... thì việc mọi người ò ạt đến gửi xe trong phôang vài phút trước giờ làm việc, giờ khai mạc, và ò ạt lái xe trong khoảng vài phút sau giờ tan sở sẽ gây ra ùn tắc cục bộ, và người lái xe phải chờ thời gian khá dài để lái xe so với bãi xe tự lái. Do đó, với các công trình có đặc điểm này, nếu muốn lắp đặt hệ thống tự động thì phải có nhiều cửa ra vào khác nhau với nhiều thang nâng để giảm thiểu thời gian lấy xe.
- Vấn đề sự cố về mất điện. Đối với bãi xe thông thường, dù tòa nhà mất điện thì vẫn có thể lái xe ra khỏi bãi xe. Tuy nhiên với hệ thống tự động, không xe

nào có thể ra khỏi hệ thống khi mất điện. Do đó, máy phát điện dự phòng cho hệ thống phải được trang bị.

- Về phòng cháy chữa cháy: ngoài việc tuân thủ các tiêu chuẩn phòng cháy chữa cháy trong xây dựng chung cho nhà cao tầng và tầng hầm, cần thiết phải lắp đặt riêng hệ thống điều khiển báo và chữa cháy tự động cho khu vực đỗ xe. Bề mặt kết cấu cần được sơn chống cháy đặc biệt và phải định kì sơn lại hoặc thay lớp khác theo thời hạn sử dụng. Các hệ thống chữa cháy tự động các nước đang lắp đặt cho bãi đỗ xe tự động đều là hệ thống chữa cháy bằng CO<sub>2</sub>. Ngoài ra, hệ thống cần được thiết kế để thoát khói thoát khí cháy.
- Các hầm chứa xe cũng phải bố trí bơm nước tự động để thoát nước khi xảy ra ngập. Đối với các hệ thống tự động đặt trong các công trình nhà ở, bệnh viện, việc tính toán các thiết kế cách âm và chống rung là rất cần thiết để giảm thiểu tiếng ồn.
- Khi xảy ra sự cố về hư hỏng thiết bị khi vận hành thì việc nhận và trả xe cũng bị tạm hoãn trong thời gian khắc phục sự cố. Vì thế cần có đội ngũ kỹ sư chuyên nghiệp khắc phục sự cố trong thời gian sớm nhất. Để giảm thiểu sự cố này thì khi thiết kế cần đầu tư chi phí để xây dựng và lắp ráp thiết bị hoạt động tốt nhất, độ tin cậy cao. Tránh xảy ra các sự cố đáng tiếc khi vận hành.
- Hơn nữa, để tránh tình trạng bị làm chuột bạch để thí nghiệm, các nhà đầu tư cũng không nên lựa chọn sử dụng các hệ thống vừa mới được phát minh, đang trong quá trình thử nghiệm sử dụng... Hơn nữa, khi hệ thống bị sự cố dù là nhỏ nhất thì không có xe nào có thể lấy ra khỏi hệ thống vì thế việc lựa chọn nhà cung cấp có đại lý bảo hành uy quyền sẵn tại Việt Nam là cần thiết.

### 5.3 Lựa chọn phương án nhà gửi xe ô tô tự động

Tóm lại, xây dựng nhà để xe tự động là giải pháp tốt nhất cho bài toán chỗ đậu xe tại các thành phố lớn của nước ta. Với điều kiện chung của Việt Nam và tính khả thi của các dự án khi áp dụng tại Việt Nam, để lựa chọn giữa 2 phương án: Nhà gửi xe tự động dạng ngầm và Nhà gửi xe tự động lộ thiên.

Điều kiện chung của Việt Nam và thành phố Hồ Chí Minh trong vấn đề xây dựng công trình ngầm:

- Luật quy hoạch đô thị chưa đề cập đầy đủ các quy định về công trình ngầm, các đô thị Việt Nam chưa có quy hoạch không gian ngầm, quy chuẩn kỹ thuật cho các loại công trình ngầm chưa thống nhất.
- Nguồn nhân lực về công trình ngầm của nước ta còn yếu. Trong thời gian vừa qua ở thành phố Hồ Chí Minh đã xảy ra nhiều sự cố kỹ thuật trong quá trình thi công các công trình nhà cao tầng có tầng ngầm. Chẳng hạn, công trình xây dựng cao ốc Pacific đã làm sập toàn bộ Viện Khoa học Xã hội và Nhân văn vùng Nam bộ, công trình xây dựng tòa cao ốc Sài Gòn Residences làm nứt chung cư số 5 Nguyễn Siêu, quận 1.
- Việc xây các bãi đậu xe ngầm phía dưới công viên khó có phương án bảo tồn cây xanh hiệu quả, ảnh hưởng đến hiệu quả xã hội.
- Thời gian thi công và chi phí đầu tư tầng hầm cao hơn nhiều so với công trình nổi trên mặt đất.
- Chủ trương của Ủy Ban Nhân Dân thành phố Hồ Chí Minh giao Sở Giao thông vận tải quy hoạch các bãi đậu xe ngầm và nghiên cứu, đề xuất chuyển hướng xây dựng bãi đậu xe ngầm sang hình thức cao tầng trên mặt đất.

Trên cơ sở phân tích tính khả thi của các dự án khi áp dụng tại Việt Nam hiện nay, nhóm làm luận văn lựa chọn thi công mô hình nhà gửi xe tự động lô thiên hình trụ (Hình 5.9).



Hình 5.9 Nhà gửi xe tự động hình trụ

## CHƯƠNG 6

### ĐIỀU KHIỂN ĐỘNG CƠ BƯỚC

#### 6.1 Động cơ bước

Động cơ bước được sử dụng rất phổ biến trong công nghiệp. Nó được ứng dụng trong điều khiển số vị trí, nhất là ở các hệ thống điều khiển vòng hở. Có thể tìm thấy động cơ bước trong các ứng dụng điều khiển van, máy công cụ CNC, tay máy công nghiệp...

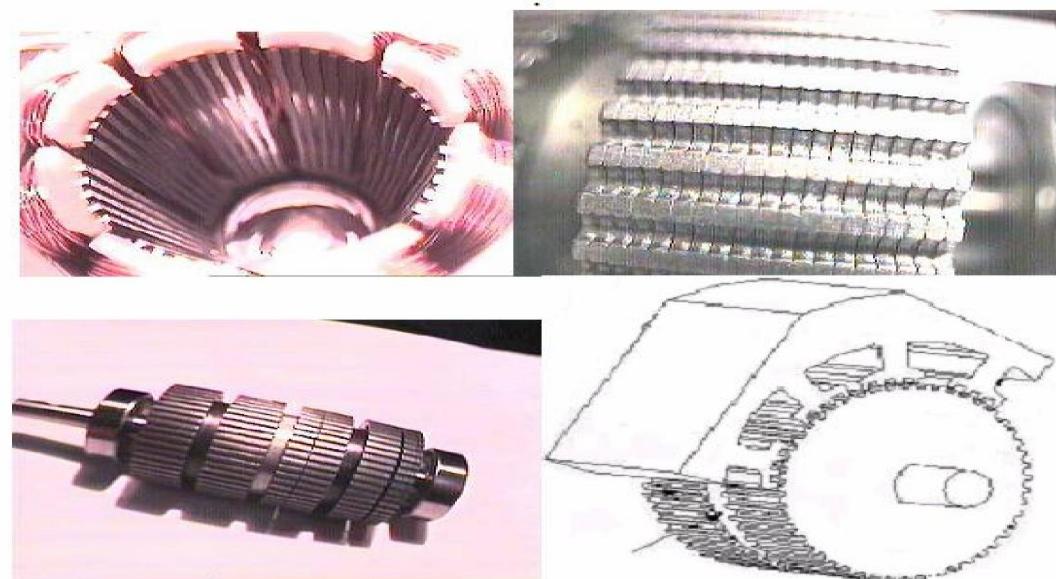
##### 6.1.1 Phân loại và cấu tạo

Động cơ bước (Stepper Motor) được chia làm 3 loại chủ yếu là động cơ bước nam châm vĩnh cửu, động cơ bước biến từ trờ và động cơ bước hỗn hợp.

###### ❖ **Động cơ bước nam châm vĩnh cửu** (Hình 6.1)

Còn gọi là động cơ bước kiểu tác dụng, các cực từ được chế tạo dạng móng.

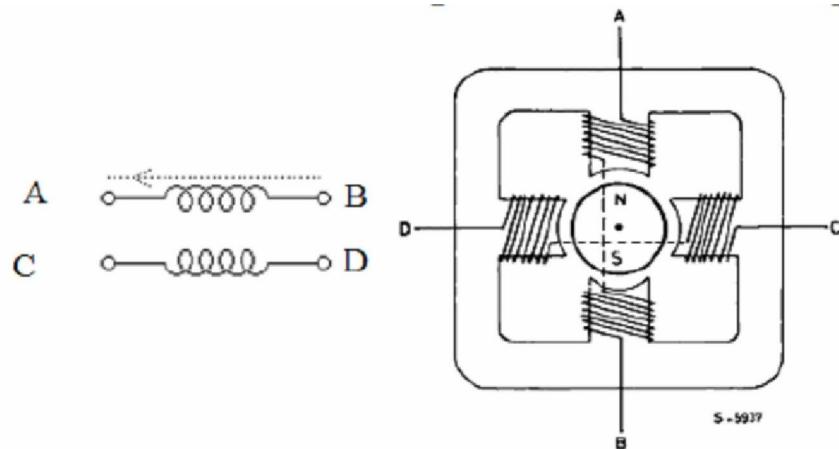
- Stator gồm nhiều cuộn dây, được quấn lên các cực từ làm bằng các lá sắt mỏng dập nhiều hình móng, ghép sát nhau.
- Rotor làm từ nhiều lá nam châm vĩnh cửu, cũng được dập rãnh để ghép lại thành hình trụ có nhiều rãnh tạo nên các cực từ.
- Trên mỗi đầu cực có thể xé nhiều rãnh hơn nữa để tăng số cực từ của động cơ, làm cho góc bước nhỏ đi nhiều lần. Loại động cơ nam châm vĩnh cửu phổ biến hiện nay có thể cho góc bước nhỏ đến 1.8 hay 3.2 độ/bước.



Hình 6.1 Cấu tạo động cơ bước nam châm vĩnh cửu

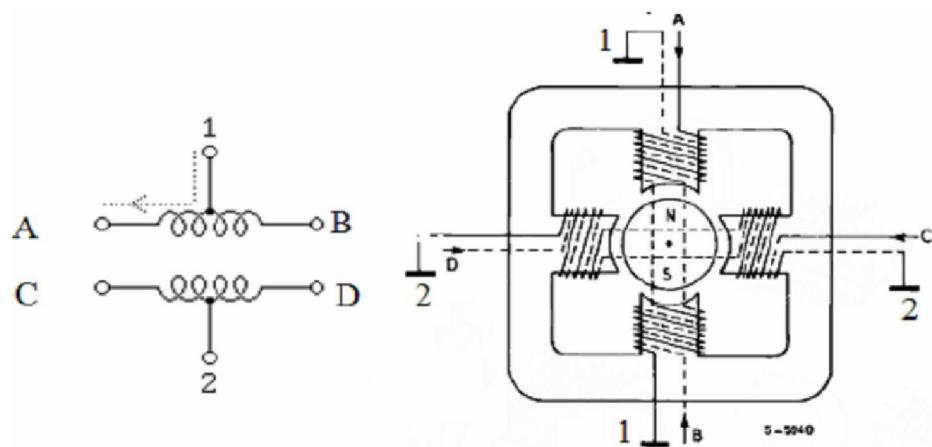
➤ Tùy thuộc vào cách nối dây Stator có thể phân biệt các loại động cơ bước nam châm vĩnh cửu:

- Đầu dây stator động cơ bước lưỡng cực (Hình 6.2)



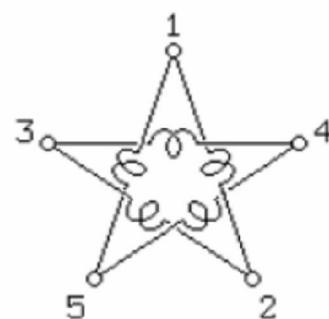
Hình 6.2 Đầu dây động cơ bước lưỡng cực

- Đầu dây stator Động cơ bước đơn cực (Hình 6.3)



Hình 6.3 Đầu dây stator động cơ bước đơn cực

- Đầu dây stator động cơ nhiều pha 3-Phase, 5-pha (Hình 6.4)

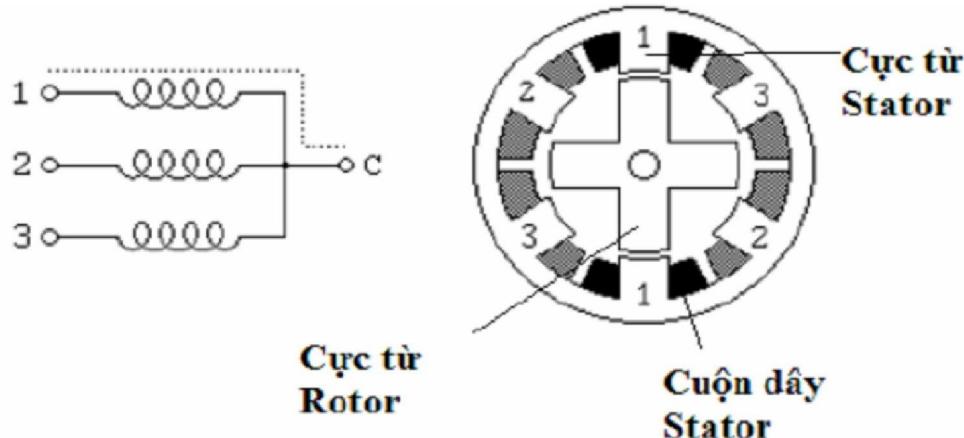


Hình 6.4 Đầu dây stator động cơ nhiều pha

Là loại ít thông dụng hơn 2 loại trên, động cơ kiểu này thường được chế tạo với các cuộn dây được quấn nối tiếp thành 1 vòng kín. Thiết kế phổ biến nhất là sử dụng dây nối 3 pha và 5 pha.

Loại này cho moment xoắn lớn nhất trong 3 loại động cơ nam châm vĩnh cửu nhưng cũng điều khiển phức tạp nhất vì cần có nhiều mạch cầu để lái các cuộn dây mà mỗi cuộn sẽ có 2 cực tính.

❖ **Động cơ bước từ trờ thay đổi** (Hình 6.5)



Hình 6.5 Động cơ bước từ trờ thay đổi

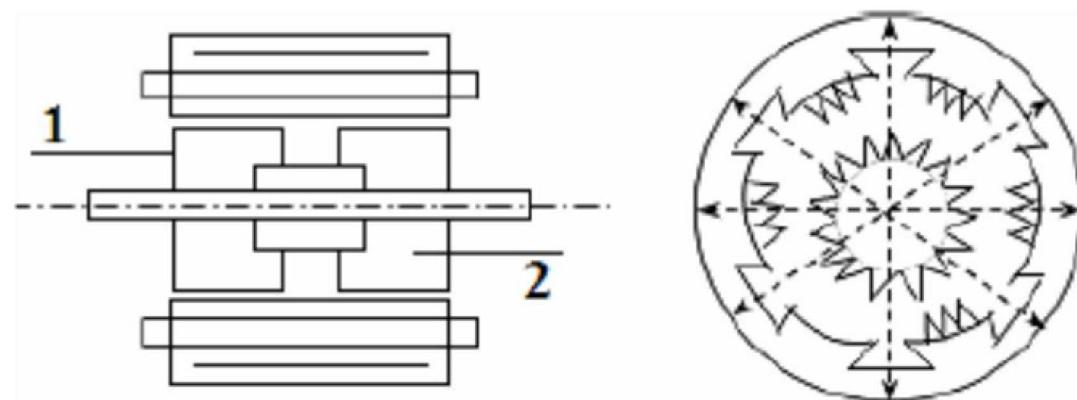
Đặc điểm của động cơ này là có số cực từ Rotor ít hơn Stator. Stator động cơ có 3 cuộn dây 1, 2 và 3 với 3 đầu nối chung với nhau tại điểm ra C. Mỗi cuộn dây quấn trên 2 hoặc 4 cực từ đối diện nhau. Rotor làm bằng vật liệu dẫn từ, là sắt non chưa bão hòa được dập rãnh và ghép lại. Từ trờ trên rotor sẽ thay đổi theo mỗi góc quay. Cả Stator và rotor đều có nhiều rãnh trên mỗi mặt cực cho phép đạt đến góc quay rất nhỏ trong mỗi bước.

Đặc điểm của loại động cơ này là chuyển động êm, tốc độ quay lớn, số (bước/vòng) lớn và tần số làm việc khá cao do không bị ảnh hưởng bởi từ trường vĩnh cửu Rotor.

❖ **Động cơ bước hồn hợp** (Hình 6.6)

Loại này là sự kết hợp giữa động cơ bước nam châm vĩnh cửu và động cơ bước từ trờ thay đổi nhằm có được đặc tính tốt nhất của 2 loại trên là moment lớn và số (bước/vòng) lớn.

Loại này có kết cấu Stator tương tự như động cơ bước biến từ trờ, với 2, 4 hoặc 5 pha. Rotor gồm có 2 tầng, tầng 1 có nhiều răng như rotor loại biến từ trờ, tầng 2 là loại nam châm vĩnh cửu.



Hình 6.6 Động cơ bước hỗn hợp

Sự kết hợp này làm cho Rotor được kích thích mạnh hơn, và kết cấu phân tầng cho phép đạt được góc bước rất bé : khoảng 0.18 – 0.27 độ/bước.

### 6.1.2 Đặc tính của động cơ bước

Động cơ bước thực chất là động cơ đồng bộ hoạt động dưới tác động của các xung dòng/áp rời rạc và kế tiếp nhau. Khi có 1 xung dòng/áp đặt vào Stator động cơ thì Rotor sẽ quay 1 góc nhất định gọi là bước của động cơ. Và nếu các xung này đặt vào Stator liên tục thì Rotor sẽ quay liên tục.

Bước của động cơ càng nhỏ thì độ chính xác khi điều khiển càng cao. Bước động cơ phụ thuộc vào số pha (cuộn dây stator), số cực từ - số răng của Stator, Rotor và phương pháp điều khiển: bước đủ, nửa bước hay vi bước.

Chiều quay của động cơ không phụ thuộc vào chiều dòng điện chạy trong phần ứng mà phụ thuộc vào thứ tự các cuộn dây được cấp xung điều khiển. Vị trí của trục động cơ được xác định bằng số bước (hay số xung đặt vào Stator) và vận tốc của động cơ thì tỉ lệ thuận với tần số xung (Số bước/giây).

Tính năng làm việc được đặc trưng bởi bước thực hiện được, đặc tính góc, tần số xung giới hạn mà khi thực hiện được 1 bước, các quá trình quá độ phải tắt đi để có thể thực hiện bước mới. Tính năng mở máy thể hiện bằng số xung cấp cực

đại khi khởi động mà không làm cho động cơ bị mất đồng bộ (bỏ bước). Tùy theo kết cấu từng động cơ mà tần số xung giới hạn có thể tiếp nhận là từ 10 – 10.000 KHz.

➤ **Các thông số động cơ bước:**

- Điện áp làm việc định mức (V).
- Dòng định mức (A).
- Điện trở cuộn dây (Ohm).
- Điện cảm cuộn dây (H).
- Góc bước (độ/bước): góc quay của động cơ ứng với 1 bước điều khiển.
- Moment hẫm (đối với động cơ nam châm vĩnh cửu): moment trên trực rotor do nam châm gây ra khi không có xung điều khiển.
- Moment duy trì: moment lớn nhất do cuộn dây kích thích tạo ra trên trực động cơ.



Hình 6.7 Động cơ bước

## 6.2 Các phương pháp điều khiển động cơ bước

Dù là loại động cơ nào thì phương pháp điều khiển cơ bản vẫn là cấp các xung điều khiển đến các cuộn dây Stator theo 1 thứ tự xác định. Tần số xung qui định tốc độ quay và thứ tự cấp qui định chiều quay của động cơ.

Lấy ví dụ động cơ lưỡng cực minh họa các phương pháp điều khiển sau đây:

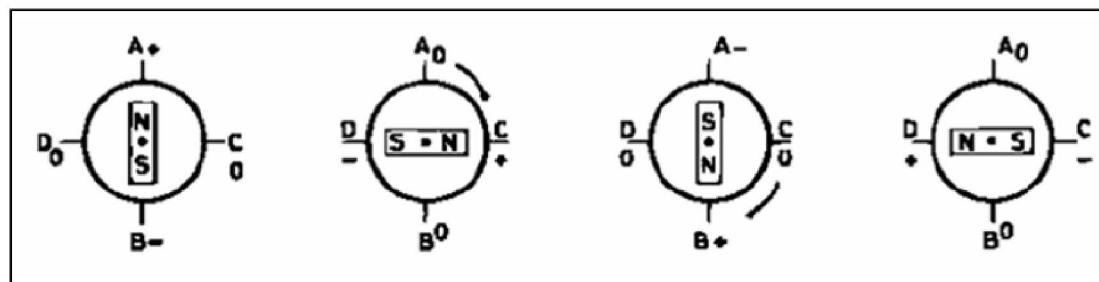
### 6.2.1 Chế độ điều khiển bước đú kiểu 1 pha

Mã xung điều khiển như sau (Hình 6.8):

Thời điểm	Cực A	Cực B	Cực C	Cực D
1	1	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	0	0	1

Hình 6.8 Mã xung điều khiển bước đú kiểu 1 pha

Hoạt động điều khiển đú bước kiểu 1 pha (Hình 6.9)



Hình 6.9 Biểu diễn hoạt động điều khiển đú bước kiểu 1 pha

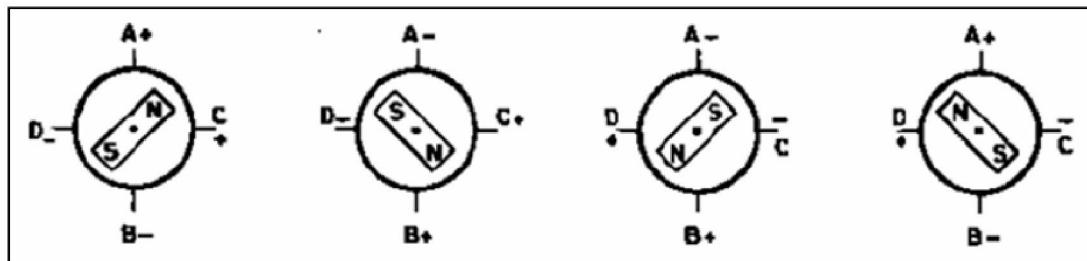
### 6.2.2 Chế độ điều khiển bước đú kiểu 2 pha

Mã xung điều khiển như sau: (Hình 6.10)

Thời điểm	Cực A	Cực B	Cực C	Cực D
1	1	0	1	0
2	0	1	1	0
3	0	1	0	1
4	1	0	0	1

Hình 6.10 Mã xung điều khiển bước đú kiểu 2 pha

Hoạt động điều khiển đủ bước kiểu 2 pha (Hình 6.11)



Hình 6.11 Biểu diễn hoạt động điều khiển đủ bước kiểu 2 pha

Chế độ điều khiển này về số bước là giống như chế độ điều khiển 2 pha nhưng vị trí bước của 2 kiểu là không giống nhau. Mặc khác kiểu 2 pha tốn năng lượng nhiều hơn kiểu 1 pha vì chỉ cần cấp điện cho 1 cực, còn kiểu 2 pha thì cần phải cấp điện cho 2 cực.

### 6.2.3 Chế độ điều khiển nửa bước

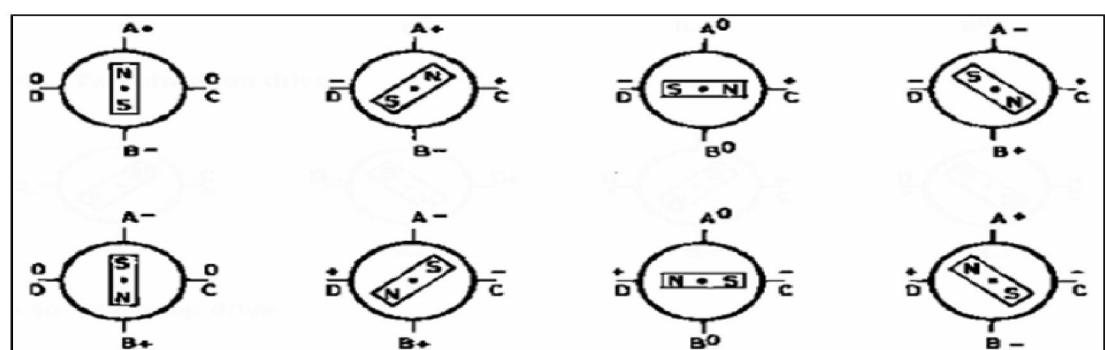
Là sự kết hợp giữa chế độ điều khiển đủ bước kiểu 1 pha và 2 pha

Mã xung điều khiển như sau: (Hình 6.12)

Thời điểm	Cực A	Cực B	Cực C	Cực D
1	1	0	0	0
2	1	0	1	0
3	0	0	1	0
4	0	1	1	0
5	0	1	0	0
6	0	1	0	1
7	0	0	0	1
8	1	0	0	1

Hình 6.12 Mã xung điều khiển kiểu nửa bước

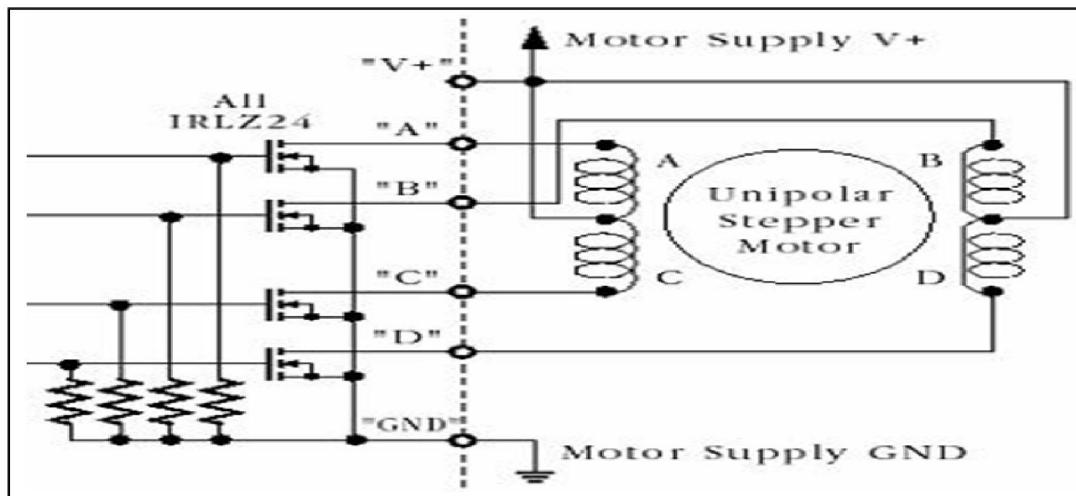
Hoạt động điều khiển nửa bước (Hình 6.13)



Hình 6.13 Biểu diễn hoạt động điều khiển kiểu nửa bước

### 6.3 Mạch điều khiển động cơ bước

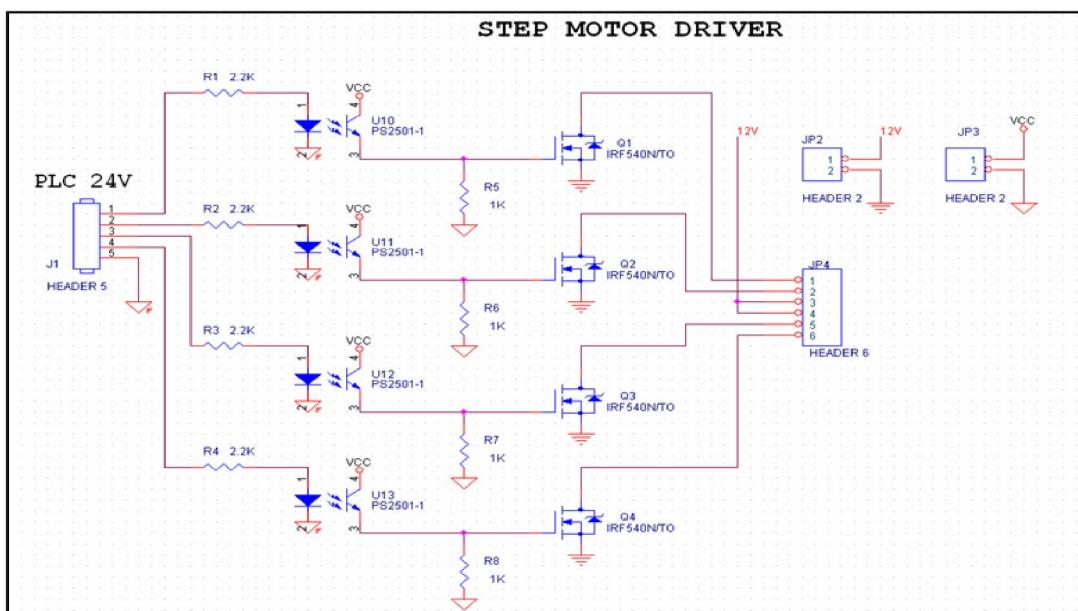
#### 6.3.1 Sơ đồ nguyên lý (Hình 6.14)



Hình 6.14 Sơ đồ nguyên lý mạch điều khiển động cơ bước

Sơ đồ nguyên lý trình bày rõ cấu tạo cuộn dây bên trong của động cơ bước, từ đó giúp ta hiểu rõ hơn khi tín hiệu điều khiển đi vào 4 dây nhận tín hiệu của động cơ bước sẽ làm cho động cơ hoạt động. Hai dây cấp nguồn 5V hoặc 12V (tùy vào loại động cơ) sẽ duy trì hoạt động của động cơ.

#### 6.3.2 Mạch lái động cơ bước (Hình 6.15)



Tín hiệu điều khiển từ PLC (24V) sẽ được đưa vào opto để cách ly điện áp. Điện áp đầu ra của OPTO 5V sẽ kích cho FET dẫn. Nguồn điện 5V hoặc 12V nuôi động cơ sẽ đi vào 2 chân chung của động cơ step và lần lượt đi qua các cuộn dây ABCD của động cơ, qua FET về mass.

Bốn tín hiệu xung điều khiển xuất ra từ PLC truyền đến 4 dây điều khiển của động cơ bước theo quy luật đã trình bày ở trên. Mỗi lần truyền các xung điều khiển ta sẽ cho trì hoãn một khoảng thời gian định trước, sau đó tiếp tục truyền tín hiệu điều khiển sẽ làm cho động cơ Step hoạt động.

## CHƯƠNG 7

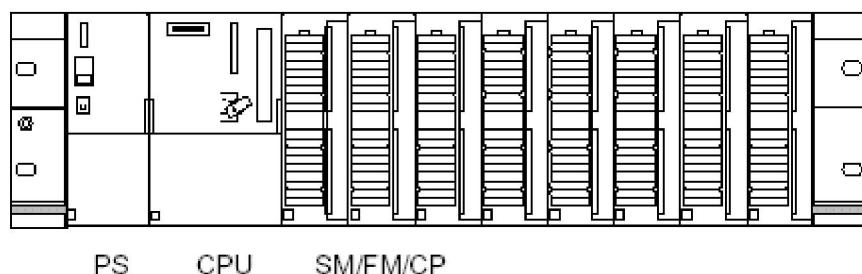
### BỘ ĐIỀU KHIỂN KHẢ LẬP TRÌNH PLC S7-300

#### 7.1 Tổng quan PLC S7 – 300

PLC S7-300 cấu trúc dạng module gồm các thành phần sau:

- CPU các loại khác nhau: 312IFM, 312C, 313, 313C, 314, 314IFM, 314C, 315, 315-2 DP, 316-2 DP...
- Module tín hiệu SM xuất nhập tín hiệu tương đồng: SM321, SM322, SM323, SM331, SM332, SM334, SM338, SM374.
- Module chức năng FM.
- Module truyền thông CP.
- Module nguồn PS307 cấp nguồn 24VDC cho các module khác, dòng 2A, 5A, 10A.
- Module ghép nối IM: IM360, IM361, IM365.

Các module được gắn trên thanh ray như (Hình 7.1), tối đa 8 module SM/FM/CP ở bên phải CPU, tạo thành một rack, kết nối với nhau qua bus connector gắn ở mặt sau của module. Mỗi module được gán một số slot tính từ trái sang phải, module nguồn là slot 1, module CPU slot 2, module kế mang số 3...



Hình 7.1 Các module được gắn trên cùng 1 thanh ray

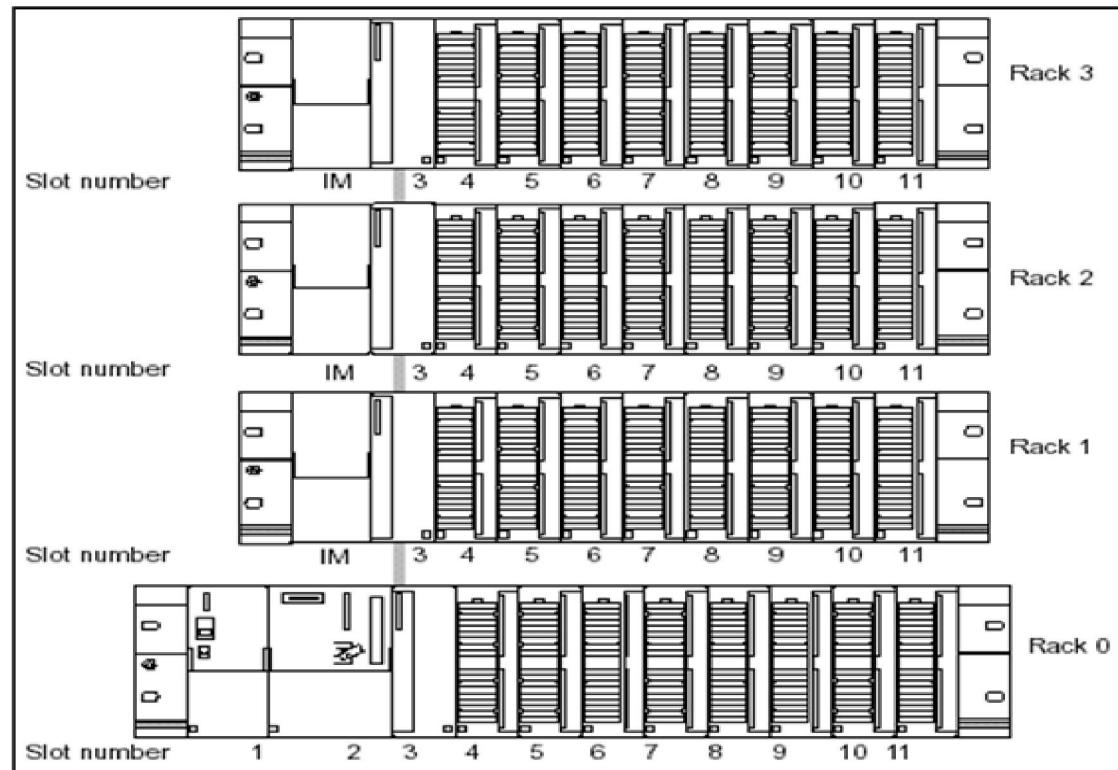
Nếu có nhiều module thì bố trí thành nhiều rack (trừ CPU312IFM và CPU313 chỉ có một rack), CPU ở rack 0, slot 2, kế đó là module phát IM360, slot 3, có nhiệm vụ kết nối rack 0 với các rack 1, 2, 3, trên mỗi rack này có module kết nối thu IM361, bên phải mỗi module IM là các module SM/FM/CP (Hình 7.3). Cáp nối hai module IM dài tối đa 10m. Các module được đánh số theo slot và dùng làm cơ sở để đặt địa chỉ đầu cho các module ngõ vào ra tín hiệu. Đối với CPU 315-2DP, 316-2DP, 318-2 có thể gán địa chỉ tùy ý cho các module.

Rack	Module Start Addresses	Slot Number										
		1	2	3	4	5	6	7	8	9	10	11
0	Digital Analog	PS	CPU	IM	0 256	4 272	8 288	12 304	16 320	20 336	24 352	28 368
1 <sup>1</sup>	Digital Analog	—	—	IM	32 384	36 400	40 416	44 432	48 448	52 464	56 480	60 496
2 <sup>1</sup>	Digital Analog	—	—	IM	64 512	68 528	72 544	76 560	80 576	84 592	88 608	92 624
3 <sup>1</sup>	Digital Analog	—	—	IM	96 640	100 656	104 672	108 688	112 704	116 720	120 736	124 <sup>2</sup> 752 <sup>2</sup>

1 Not with the CPU 312 IFM/313

2 Not with the CPU 314 IFM

Hình 7.2 Định nghĩa địa chỉ cho các module



Hình 7.3 Hình các module được gắn trên nhiều rack

Mỗi địa chỉ tương ứng với một byte. Với các module số địa chỉ một ngõ vào hay ra là x.y, x là địa chỉ byte, y có giá trị từ 0 đến 7. Ví dụ module SM321 DI 32 có 32 ngõ vào gắn kê CPU slot 4 có địa chỉ là I0.y, I1.y, I2.y, I3.y, I là ký hiệu chỉ ngõ vào số. Module analog có địa chỉ theo word, ví dụ module SM332 AO4 có 4 ngõ ra analog gắn ở slot 5 rack 1 có địa chỉ PQW400, PQW402, PQW404, PQW406, ngõ ra số có ký hiệu là Q còn ngõ vào analog ký hiệu là PIW.

Các CPU 312IFM, 314 IFM, 31xC có tích hợp sẵn một số module mở rộng:

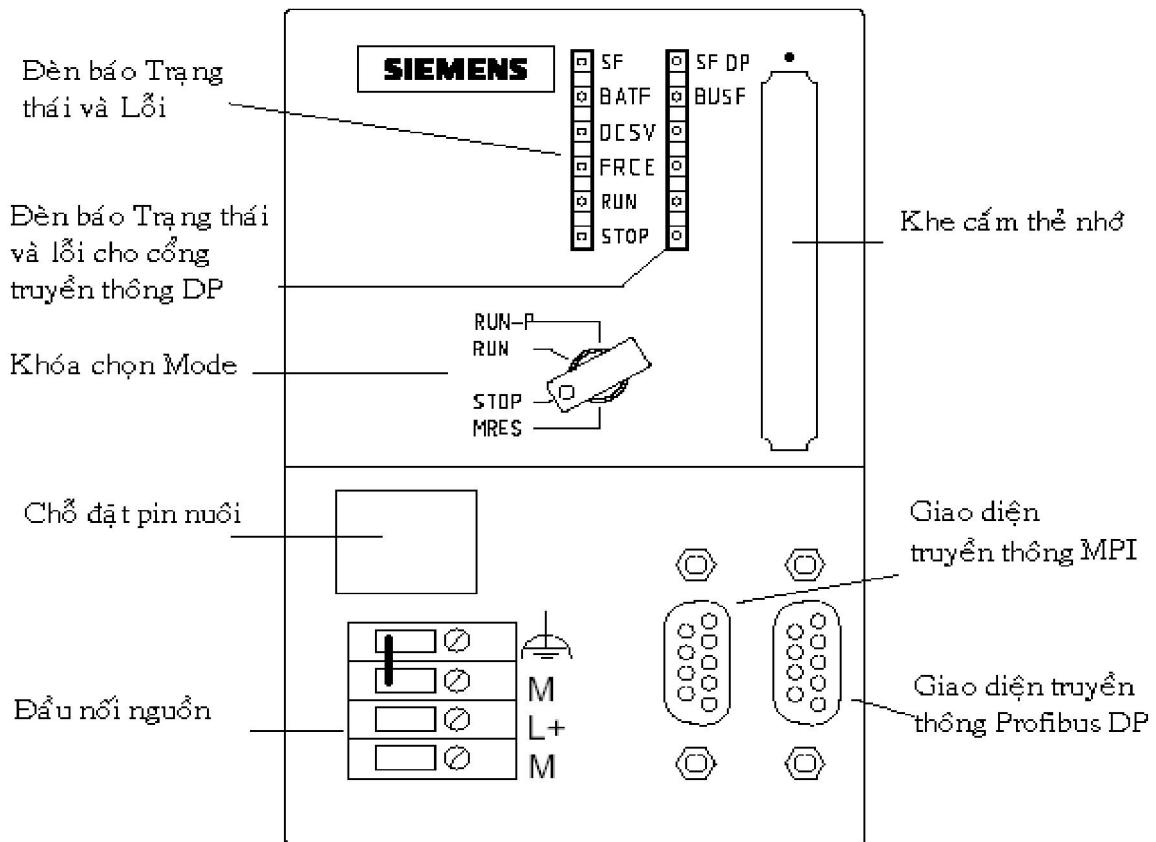
- CPU 312IFM, 312C: 10 ngõ vào số địa chỉ I124.0 ... I124.7, I125.1; 6 ngõ ra số Q124.0...Q124.5.
- CPU 313C: 24 DI I124.0..126.7, 16DO Q124.0..125.7, 5 ngõ vào tương đồng AI địa chỉ 752..761, hai ngõ ra AO 752..755.

- CPU 314IFM: 20 ngõ vào số I124.0 ... I126.3; 16 ngõ ra số Q124.0 ...Q125.7; 4 ngõ vào tương đồng PIW128, PIW130, PIW132, PIW134; một ngõ ra tương đồng PQW128.

## 7.2 Module CPU (Hình 7.4)

Các module CPU khác nhau theo hình dạng chức năng, vận tốc xử lý lệnh. Loại 312IFM, 314IFM không có thẻ nhớ. Loại 312IFM, 313 không có pin nuôi. Loại 315-2DP, 316-2DP, 318-2 có cổng truyền thông DP. Các đèn báo có ý nghĩa sau:

- SF ... (đỏ) ... lỗi phần cứng hay mềm.
- BATF ... (đỏ) ... lỗi pin nuôi.
- DC5V ... (lá cây) ... nguồn 5V bình thường.
- FRCE ... (vàng) ... force request tích cực.
- RUN ... (lá cây) ... CPU mode RUN ; LED chớp lúc start-up tần suất 1 Hz; mode HALT tần suất 0.5 Hz.
- STOP mode ... (vàng) ... CPU mode STOP hay HALT hay start-up; LED chớp khi memory reset request.



Hình 7.4 Module CPU

**Khóa mode có 4 vị trí:**

- RUN-P chế độ lập trình và chạy.
- RUN chế độ chạy chương trình.
- STOP ngừng chạy chương trình.
- MRES reset bộ nhớ.

Thẻ nhớ có thể có dung lượng từ 16KB đến 4MB, chứa chương trình từ PLC chuyển qua và chuyển chương trình ngược trở lại cho CPU.

Pin nuôi giúp nuôi chương trình và dữ liệu khi bị mất nguồn (tối đa 1 năm), ngoài ra còn nuôi đồng hồ thời gian thực. Với loại CPU không có pin nuôi thì cũng có một phần vùng nhớ được duy trì.

Thông qua cổng truyền thông MPI (MultiPoint Interface) có thể nối: máy tính lập trình, màn hình OP (Operator panel), các PLC có cổng MPI (S7-300, M7-300, S7-400, M7-400, C7-6xx), S7-200, vận tốc truyền đến 187.5kbps (12Mbps)

với CPU 318-2, 10.2 kbps với S7-200) . Cổng Profibus –DP nối các thiết bị trên theo mạng Profibus với vận tốc truyền lên đến 12Mbps.

#### Các vùng nhớ của PLC:

**Vùng nhớ chương trình** (load memory) chứa chương trình người dùng (không chứa địa chỉ ký hiệu và chú thích) có thể là RAM hay EEPROM trong CPU hay trên thẻ nhớ.

**Vùng nhớ làm việc** (working memory) là RAM, chứa chương trình do vùng nhớ chương trình chuyển qua, chỉ các phần chương trình cần thiết mới được chuyển qua, phần nào không cần ở lại vùng nhớ chương trình.

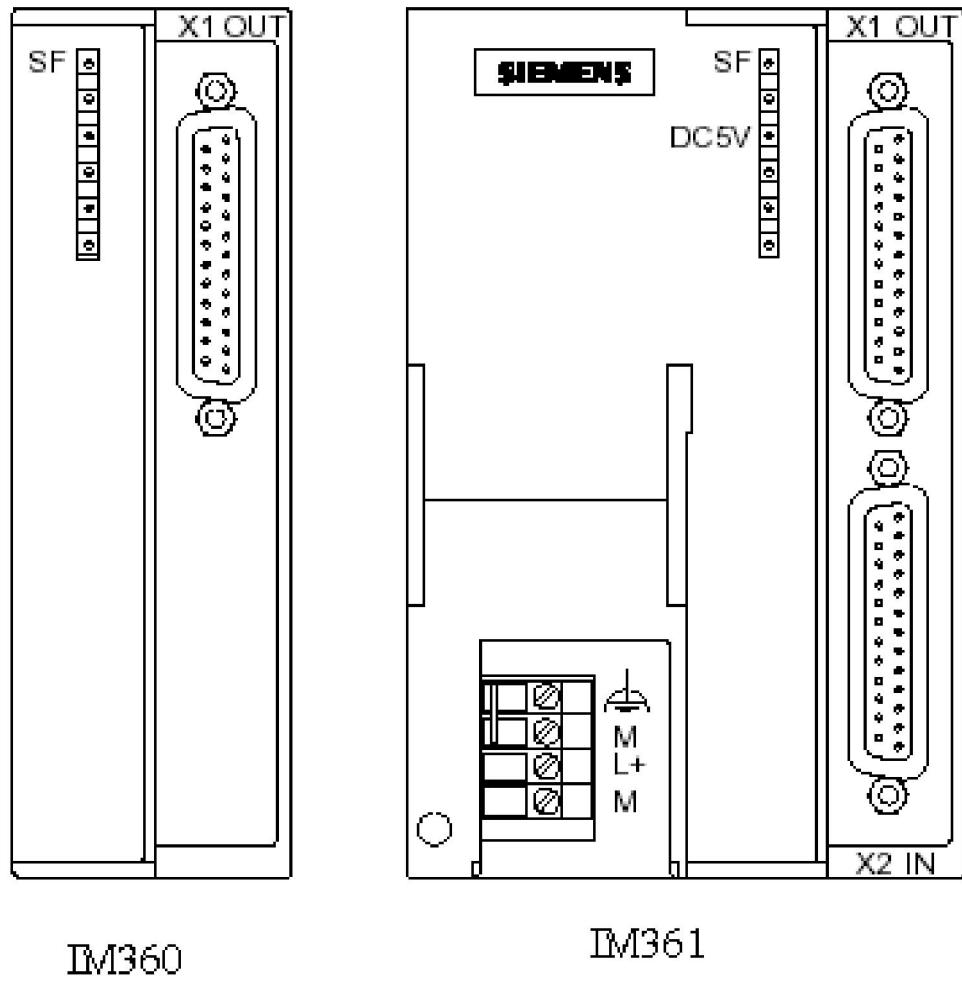
**Vùng nhớ hệ thống** (system memory) phục vụ cho chương trình người dùng, bao gồm timer, counter, vùng nhớ dữ liệu M, bộ nhớ đệm xuất nhập...

Trên CPU 312IFM và 314 IFM vùng nhớ chương trình là RAM và EEPROM; các CPU khác có pin nuôi, vùng nhớ chương trình là RAM và thẻ nhớ. Khi mất nguồn hay ở chế độ MRES ( reset bộ nhớ) RAM sẽ bị xóa. Một số vùng nhớ của RAM (timer, counter, vùng nhớ M, khôi dữ liệu...) có thể khai báo là lưu giữ (retentive) bằng phần mềm S7 để chuyển các vùng này sang bộ nhớ lưu giữ (NVRAM non volatile) dù không có pin nuôi, kích thước cụ thể tùy loại CPU.

Thông số	CPU 312IFM	CPU 313	CPU 314	CPU 314IFM
Working memory	6KB	12KB	24KB	32KB
Load memory	20KBRAM 20KBEEPROM	20KBRAM up to 4MB FEPROM (memory card)	40KB up to 4MB FEPROM (memory card)	48KB RAM 48KBEEPROM
Vận tốc	0.7ms/1000 lệnh nhị phân	0.7ms/1000 lệnh nhị phân	0.3ms/1000 lệnh nhị phân	0.3ms/1000 lệnh nhị phân
Data Memory	1KB Retentivity adjustable MB0..MB71 Preset MB0..MB15	2KB Retentivity adjustable MB0..MB71 Preset MB0..MB15	2KB Retentivity adjustable MB0..MB255 Preset MB0..MB15	2KB Retentivity adjustable MB0..MB143 Preset MB0..MB15
Counter	adjustable Retentivity C0..C31 Preset C0..C7	adjustable Retentivity C0..C63 Preset C0..C7	adjustable Retentivity C0..C63 Preset C0..C7	Adjustable Retentivity C0..C63 Preset C0..C7
Timer	T0..T63 no retentivity	T0..T127 Adjustable Retentivity T0..T31 Preset: no	T0..T127 Adjustable Retentivity T0..T127 Preset: no	T0..T127 Adjustable Retentivity T0..T71 Preset: no
Digital inputs	10 integrated + 128	128	512	496 + 20 integrated
Digital outputs	6 integrated + 128	128	512	496 + 16 integrated
Analog inputs	32	32	64	64 + 4 integrated
Analog outputs	32	32	64	64 + 1 integrated
Process image input	I0.0.. I127.7	I0.0.. I127.7	I0.0.. I127.7	
Process image output	Q0.0 ..Q127.7	Q0.0 ..Q127.7	Q0.0 ..Q127.7	

Hình 7.5 Các thông số của các vùng nhớ của PLC

### 7.3 Module IM (Hình 7.6)



Hình 7.6 Module IM

Module IM360 gắn ở rack 0 kế CPU dùng để ghép nối với module IM361 đặt ở các rack 1, 2, 3 giúp kết nối các module mở rộng với CPU khi số module lớn hơn 8. Cáp nối giữa hai rack là loại 368. Trong trường hợp chỉ có hai rack, ta dùng loại IM365.

## 7.4 Module tín hiệu

### Module vào số có các loại sau:

- SM 321; DI 32 \_ 24 VDC.
- SM 321; DI 16 \_ 24 VDC.
- SM 321; DI 16 \_ 120 VAC, 4\*4 nhóm.
- SM 321; DI 8 \_ 120/230 VAC, 2\*4 nhóm.
- SM 321; DI 32 \_ 120 VAC 8\*4 nhóm.

### Module ra số:

- SM 322; DO 32 \_ 24 VDC/0.5 A, 8\*4 nhóm.
- SM 322; DO 16 \_ 24 VDC/0.5 A, 8\*2 nhóm.
- SM 322; DO 8 \_ 24 VDC/2 A, 4\*2 nhóm.
- SM 322; DO 16 \_ 120 VAC/1 A, 8\*2 nhóm.
- SM 322; DO 8 \_ 120/230 VAC/2 A, 4\*2 nhóm.
- SM 322; DO 32 \_ 120 VAC/1.0 A, 8\*4 nhóm.
- SM 322; DO 16 \_ 120 VAC ReLay, 8\*2 nhóm.
- SM 322; DO 8 \_ 230 VAC Relay, 4\*2 nhóm.
- SM 322; DO 8 \_ 230 VAC/5A Relay, 1\*8 nhóm.

### Module vào/ ra:

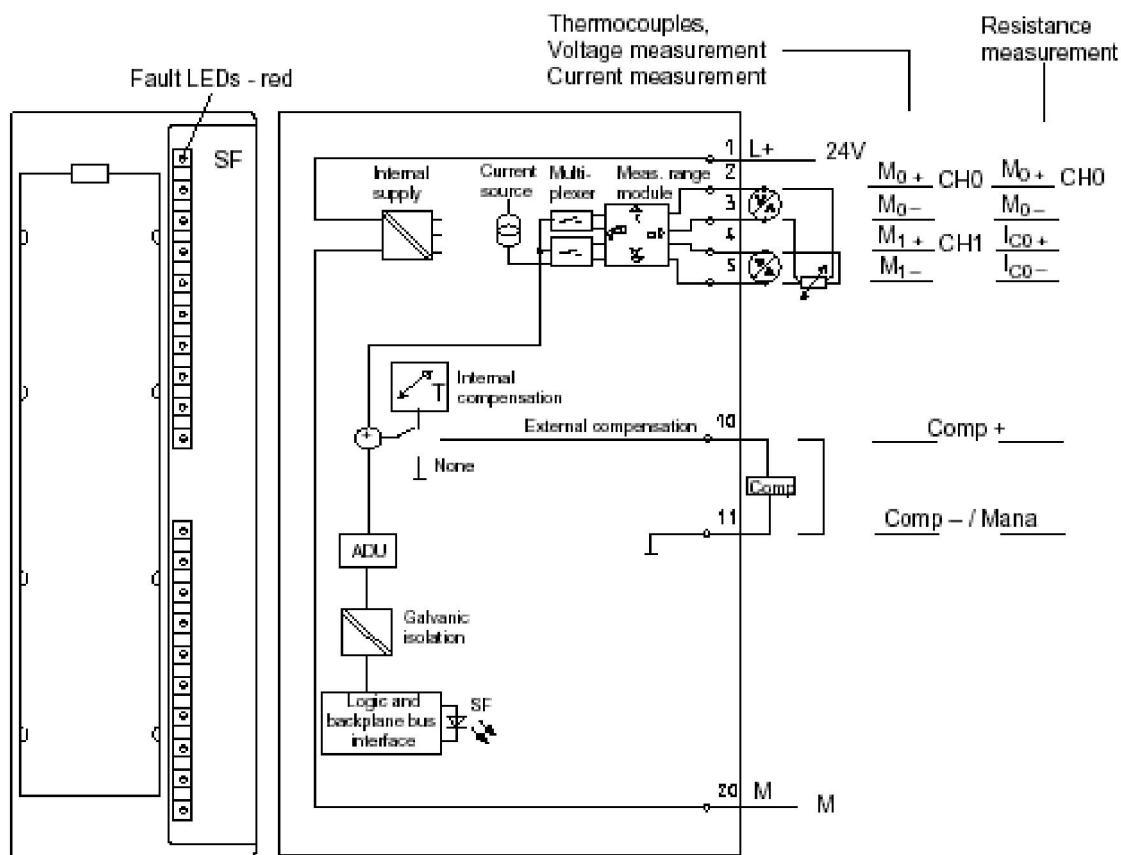
- SM 323; DI 16/DO 16 \_ 24 VDC/0.5 A.
- SM 323; DI 8/DO 8 \_ 24 VDC/0.5 A.

### Module Analog in

Module analog in có nhiều ngõ vào dùng để đo điện áp, dòng điện , điện trở. Có nhiều tầm đo với độ phân giải, thời gian chuyển đổi khác nhau. Cài đặt thông số hoạt động cho Module bằng phần mềm S7 Simatic 300 Station Hardware.

- **SM331 AI 2\*12:** module chuyển đổi hai kênh vi sai áp hoặc dòng, hoặc một kênh điện trở 2/3/4 dây, dùng phương pháp tích phân, thời gian chuyển

đổi từ 5ms đến 100ms, độ phân giải 9, 12, 14 bit + dấu, các tầm đo như sau:  $\pm 80$  mV;  $\pm 250$  mV;  $\pm 500$  mV;  $\pm 1000$  mV;  $\pm 2.5$  V;  $\pm 5$  V;  $1 \dots 5$  V;  $\pm 10$  V;  $\pm 3.2$  mA;  $\pm 10$  mA;  $\pm 20$  mA;  $0 \dots 20$  mA;  $4 \dots 20$  mA. Điện trở  $150 \Omega$ ;  $300 \Omega$ ;  $600 \Omega$ ; Đo nhiệt độ dùng cặp nhiệt E, N, J, K, L, nhiệt kế điện trở Pt 100, Ni 100. Các thông số mặc định đã được cài sẵn trên module, kết hợp với đặt vị trí của module tầm đo (bốn vị trí A, B, C, D) nếu không cần thay đổi thì có thể sử dụng ngay.



Hình 7.7 Cấu tạo của Module SM331 AI 2\*12

Measuring Range Module Setting	Measuring Method	Measuring Range
A	Voltage	$\pm 1000 \text{ mV}$
B	Voltage	$\pm 10 \text{ V}$
C	Current, 4-wire transducer	4 to 20 mA
D	Current, 2-wire transducer	4 to 20 mA

Hình 7.8 Tầm đo và phương pháp đo cho các loại module

- **SM331, AI 8\*12 bit**, 8 kênh vi sai chia làm hai nhóm, độ phân giải 9 (12,14) bit + dấu.
- **SM331, AI 8\*16 bit**, 8 kênh vi sai chia làm 2 nhóm, độ phân giải 15 bit + dấu.

### Module Analog Out

Cung cấp áp hay dòng phụ thuộc số nhị phân phụ hai

- SM332 AO 4\*12 bit: 4 ngõ ra dòng hay áp độ phân giải 12 bit, thời gian chuyển đổi 0.8 ms.
- SM332 AO 2\*12 bit.
- SM332 AO 4\*16 bit.

### Module Analog In/Out

- SM 334; AI 4/AO 2 \* 8 Bit.
- SM334; AI 4/AO 2\* 12 Bit.

### Module chức năng FM

- FM350-2 : đếm xung tám kênh.
- FM351, 353, 354, 357-2 : điều khiển định vị.
- FM352: bộ điều khiển cam điện tử.
- FM355: bộ điều khiển hệ kín.

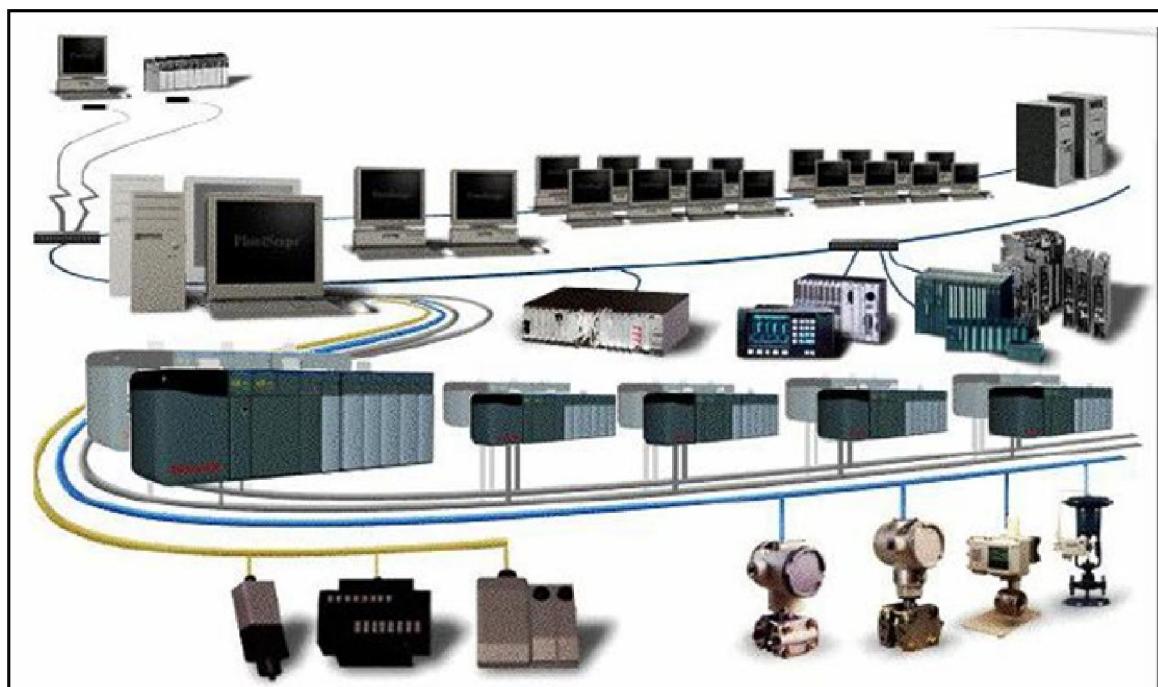
## CHƯƠNG 8

### TỔNG QUAN VỀ HỆ SCADA

#### 8.1 Khái quát hệ thống SCADA (Hình 8.1)

Trong thời đại công nghệ sản xuất hiện đại ngày nay, hệ thống điều khiển giám sát và thu thập dữ liệu được áp dụng ngày càng rộng rãi trong các nhà máy, xí nghiệp sản xuất tự động. Hệ thống SCADA có thể kết nối với rất nhiều thiết bị đo đặc và điều khiển các thiết bị chấp hành từ khoảng cách rất xa (lên đến vài ngàn cây số).

SCADA (Supervisory Control And Data Acquisition) là một hệ thống điều khiển giám sát và thu thập dữ liệu. Để có thể điều khiển và giám sát từ xa thì hệ SCADA phải có hệ thống giao thức truy cập, truyền tải dữ liệu cũng như hệ giao diện giữa người và máy (HMI – Human Machine Interface).



Hình 8.1 Hệ thống giám sát và thu thập dữ liệu SCADA

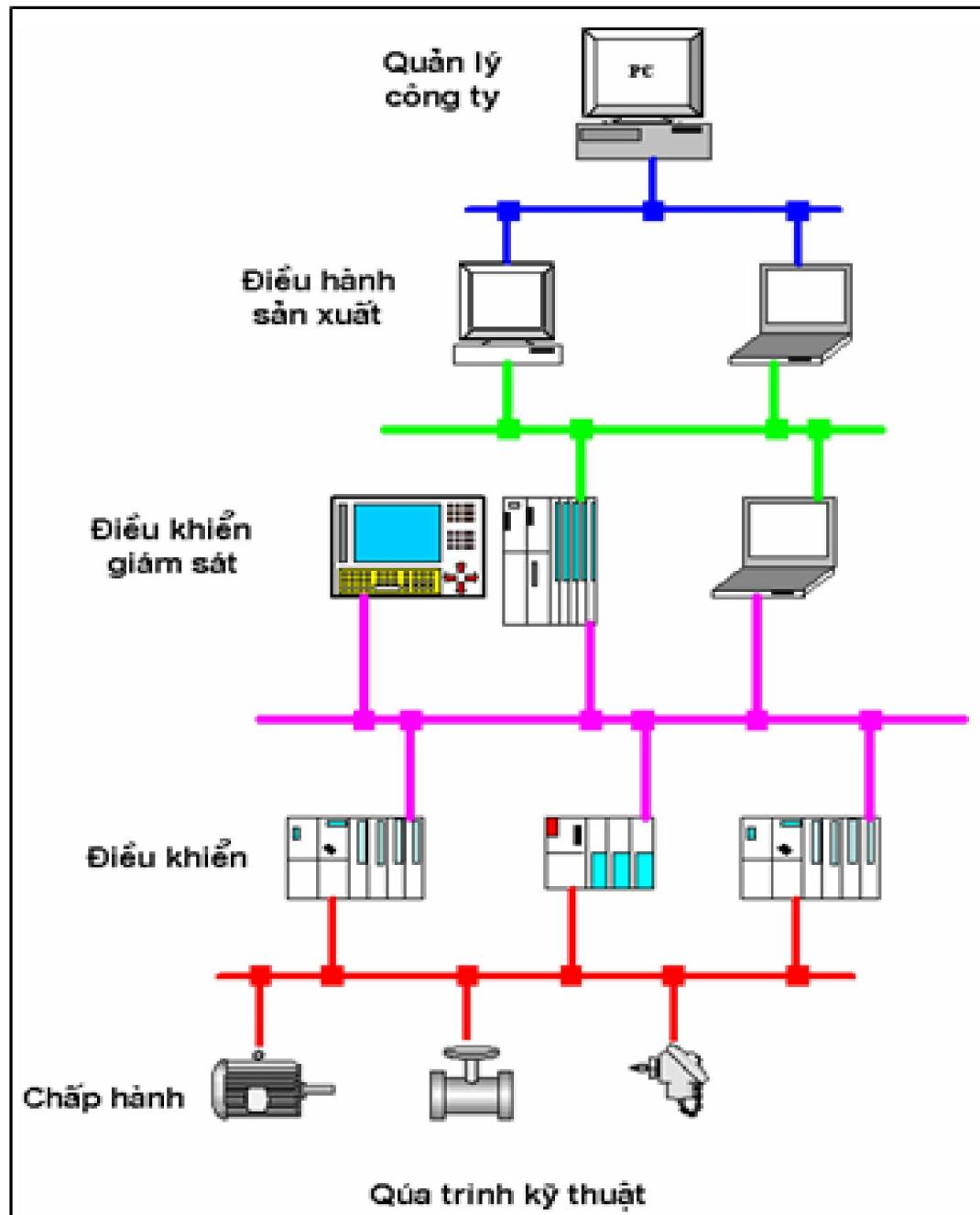
Trong hệ thống điều khiển giám sát thì giao diện người và máy là một thành

phần quan trọng để phục vụ cho việc quan sát và thao tác vận hành ở cấp điều khiển cục bộ. Vì lý do giá thành, đặc điểm kỹ thuật nên các màn hình vận hành (OP – Operator Panel), bảng điều khiển cảm ứng (TP – Touch Panel), Multi Panel ... chuyên dụng được sử dụng nhiều.

Có nhiều quá trình sử dụng hệ thống SCADA như: thủy điện, các khâu xử lý và phân phối nước, khí tự nhiên... Các hệ thống SCADA cho phép các nơi xa liên lạc với phương tiện điều khiển và cung cấp dữ liệu điều khiển cần thiết cho các quá trình điều khiển. Khi khoảng cách đến các nơi xa tăng thì càng khó truy cập hơn, khi đó SCADA trở thành sự chọn tốt nhất cho người điều hành để điều chỉnh và quan sát.

## 8.2 Mô hình phân cấp chức năng

### 8.2.1 Mô hình phân cấp (Hình 8.2)



Hình 8.2 Mô hình phân cấp chức năng của hệ thống SCADA

Để sắp xếp, phân loại các chức năng tự động hóa của một hệ thống điều khiển

và giám sát người ta thường sử dụng mô hình như trên. Ở cấp dưới thì các chức năng càng đòi hỏi yêu cầu cao hơn về độ nhanh nhẹy, thời gian phản ứng. Một chức năng ở áp trên được thực hiện dựa trên các chức năng ở cấp dưới nhưng ngược lại lượng thông tin cần trao đổi và xử lý lại lớn hơn nhiều.

**Cấp chấp hành:** Các chức năng chính của cấp chấp hành là đo lường, dẫn động và chuyển đổi tín hiệu trong trường hợp cần thiết. Thực tế, đa số các thiết bị cảm biến hay chấp hành cũng có phần điều khiển riêng cho việc thực hiện đo lường/ truyền động được chính xác và nhanh nhẹy. Các thiết bị thông minh (có bộ vi xử lý riêng) cũng có thể đảm nhận việc xử lý và chuẩn bị thông tin trước khi đưa lên cấp trên điều khiển.

**Cấp điều khiển:** Nhiệm vụ chính của cấp điều khiển là nhận thông tin từ các bộ cảm biến, xử lý các thông tin đó theo một thuật toán nhất định và truyền đạt lại kết quả xuống các bộ chấp hành. Máy tính đảm nhận việc theo dõi các công cụ đo lường, tự thực hiện các thao tác như ấn nút mở/ đóng van, điều chỉnh cần gạt, núm xoay,... Đặc tính nổi bật của cấp điều khiển là xử lý thông tin. Cấp điều khiển và cấp chấp hành hay được gọi chung là cấp trường (Field level) chính vì các bộ điều khiển, cảm biến và chấp hành được cài đặt trực tiếp tại hiện trường gần kề với hệ thống kỹ thuật.

**Cấp điều khiển giám sát:** có chức năng giám sát và vận hành một quá trình kỹ thuật, có nhiệm vụ hỗ trợ người sử dụng trong việc cài đặt ứng dụng, thao tác theo dõi và xử lý những tình huống bất thường. Ngoài ra trong một số trường hợp, cấp này còn thực hiện các bài toán điều khiển cao cấp như điều khiển phối hợp, điều khiển trình tự và điều khiển theo công thức.

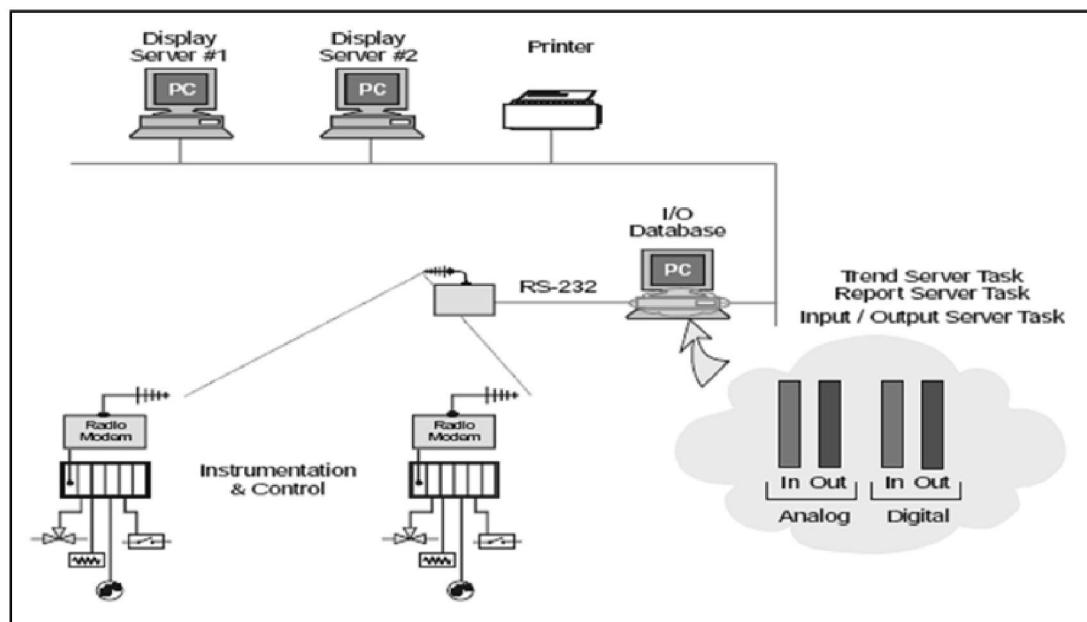
Thông thường người ta chỉ coi ba cấp dưới thuộc phạm vi của một hệ thống điều khiển và giám sát. Tuy nhiên biểu thị hai cấp trên cùng (Quản lý công ty

và Điều hành sản xuất) sẽ giúp ta hiểu thêm một mô hình lý tưởng cho cấu trúc chức năng tổng thể cho các công ty sản xuất công nghiệp.

### 8.2.2 Chức năng nhiệm vụ

Chức năng của mỗi cấp SCADA cung cấp những dịch vụ sau:

- Thứ nhất: thu thập từ xa các số liệu về sản xuất và tổ chức việc lưu giữ nhiều loại cơ sở dữ liệu (số liệu lịch sử về sản xuất, sự kiện thao tác, báo động...)
- Thứ hai: cung cấp các dịch vụ về điều khiển, giám sát hệ sản xuất.
- Thứ ba: hiển thị báo cáo tổng kết về quá trình sản xuất (trang màn hình, trang đồ thị, trang sự kiện, trang báo động, trang báo cáo sản xuất...).
- Thứ tư: điều khiển từ xa quá trình sản xuất (đóng/cắt các máy móc thiết bị ...).
- Thứ năm: thực hiện các dịch vụ về truyền số liệu trong hệ và ra ngoài (đọc viết số liệu PLC/RTU, gửi trả lời các bản tin yêu cầu của cấp trên về số liệu, về thao tác hệ).



Hình 8.3 Một hệ thống SCADA điển hình

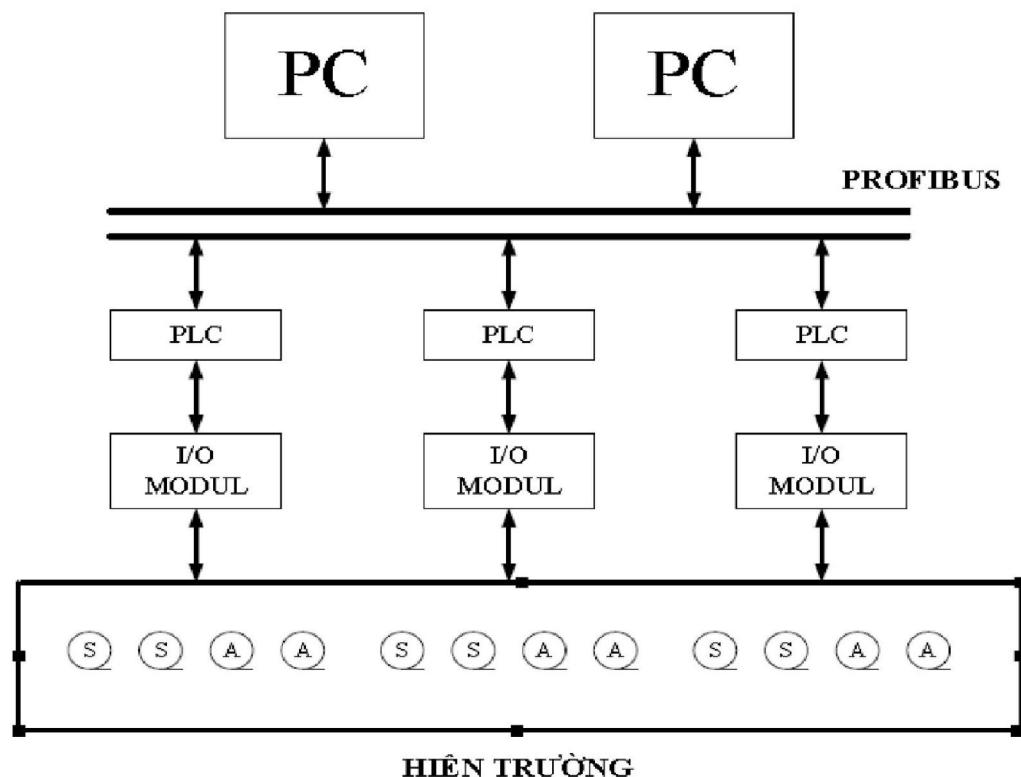
Các chức năng đó mỗi thứ đều có những yêu cầu đặc biệt đối với các bộ phận phần cứng, phần mềm, phần chuyên trách của SCADA. Cụ thể là phần đo, giám sát từ xa cần bảo đảm thu thập dữ liệu hiển thị in án đủ những số liệu cần cho quản lý kỹ thuật. Phần điều khiển tinh xa phải đảm bảo được việc kiểm tra đóng cắt an toàn, đúng đắn. Phần truyền tin xa phải quy định rõ các nhiệm vụ truyền số liệu hiện trường, đặc biệt là thủ tục truyền với các SCADA cấp trên.

Ngày nay, hầu hết các hệ SCADA còn có khả năng liên kết với các hệ thống thương mại có cấp độ cao hơn, cho phép đọc viết theo cơ sở dữ liệu chuẩn như Oracle, Access, Microsoft SQL...

### 8.3 Cấu trúc hệ SCADA (Hình 8.4)

#### 8.3.1 Cấu trúc

Cấu trúc chung của hệ SCADA được minh họa như sau: (Hình 8.4)

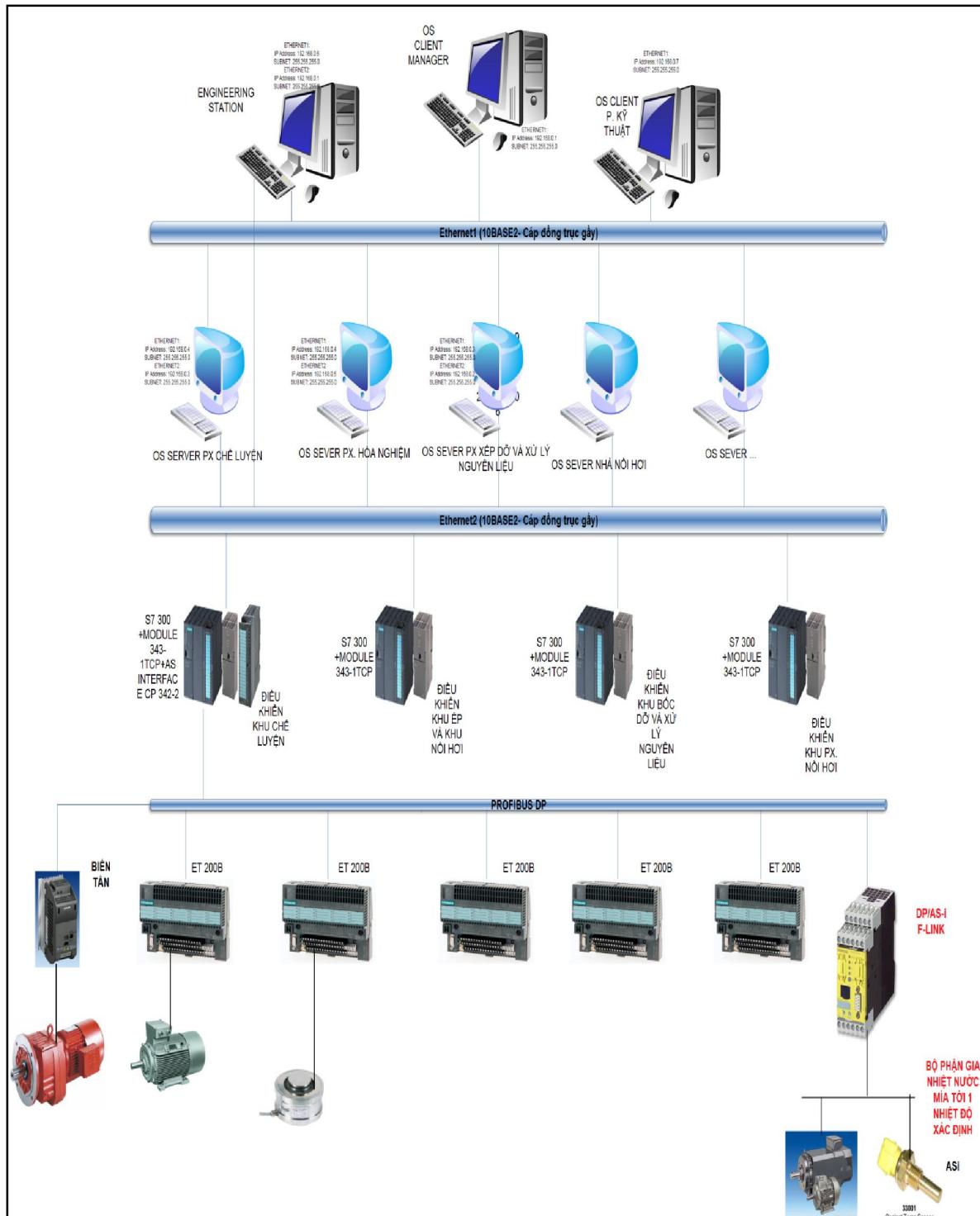


Hình 8.4 Cấu trúc chung của một hệ scada

**Các thành phần chính của hệ thống SCADA bao gồm:**

- Giao diện quá trình: bao gồm các cảm biến, thiết bị đo, thiết bị chuyển đổi và các cơ cấu chấp hành.
- Thiết bị điều khiển tự động: gồm các bộ điều khiển khả trình PLC (Programmable Logic Controller), các thiết bị điều chỉnh số đơn lẻ CDC (Compact Digital Controller) và máy tính với các phần mềm điều khiển tương ứng.
- Hệ thống điều khiển giám sát: gồm các phần mềm và giao diện giữa người và máy HMI, trạm vận hành, giám sát và điều khiển cao cấp.
- Hệ thống truyền thông: bus cảm biến/ chấp hành, bus trường, bus hệ thống.
- Hệ thống bảo vệ, cơ chế thực hiện chức năng an toàn.

➤ Ví dụ cho một hệ thống scada cho nhà máy đường: (Hình 8.5)

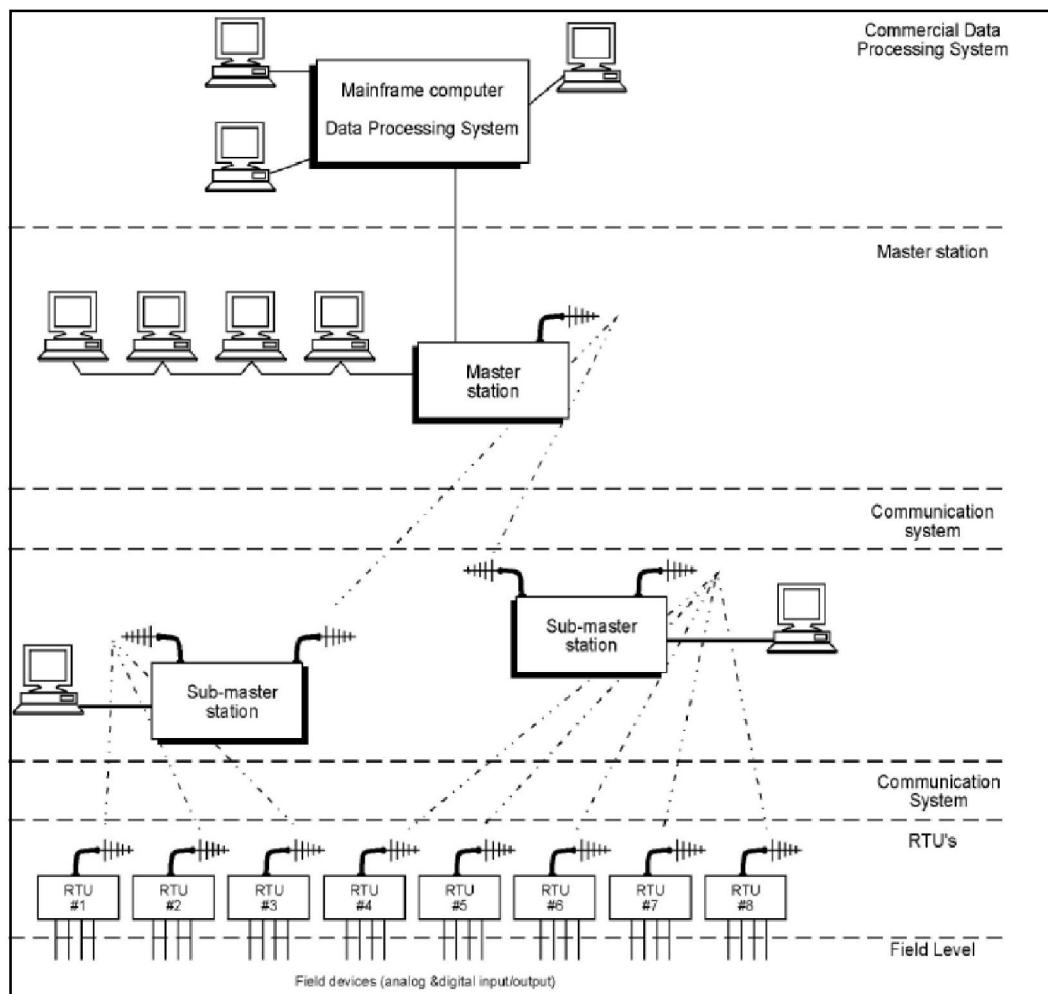


Hình 8.5 Hệ thống Scada thiết kế cho nhà máy đường

### 8.3.2 Phần cứng

Hệ thống SCADA bao gồm các RTU (Remote Terminal Unit) gọi là thiết bị đầu cuối, nhiệm vụ thu thập dữ liệu và truyền dữ liệu đến các trạm chủ thông qua các chuẩn truyền thông. Trạm chủ hiển thị dữ liệu thu được, và cho phép người vận hành thực hiện tác vụ điều khiển từ xa.

Dưới đây là mô hình của một hệ thống SCADA (Hình 8.6)



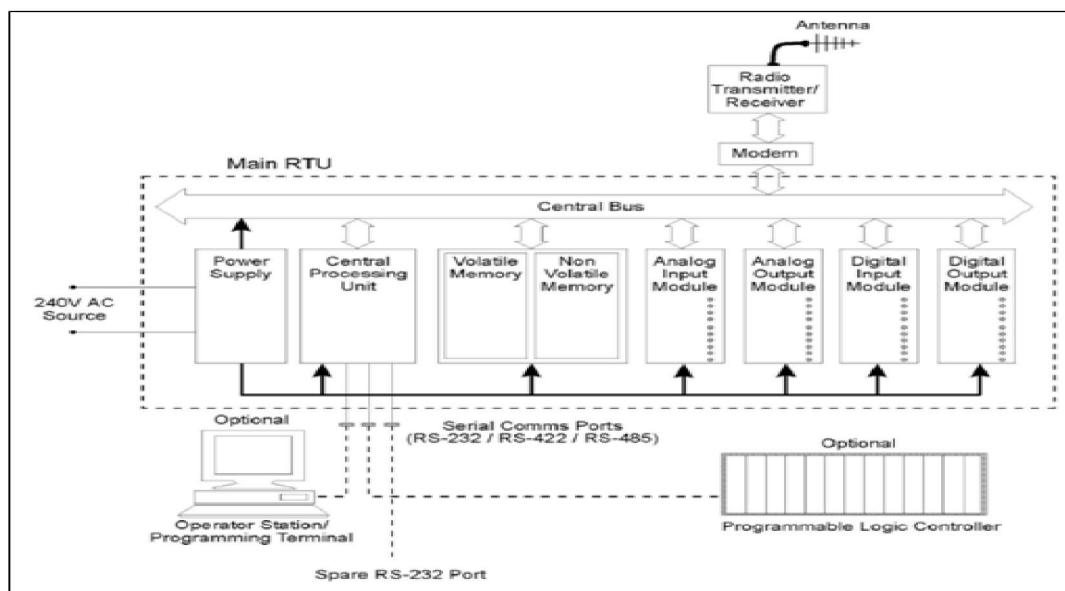
Hình 8.6 Mô hình một hệ SCADA

**Ứng với mô hình phân cấp chức năng, một hệ SCADA sẽ có 5 cấp thiết bị phần cứng sau:**

- Các thiết bị đo lường và chấp hành ở hiện trường.
- Các thiết bị đầu cuối RTU.
- Hệ thống truyền thông.
- Các trạm chủ.
- Hệ thống máy tính phân tích xử lý số liệu.

Truyền thông giữa trạm chủ và các khu vực ở xa thông qua các loại phương tiện truyền như cáp đồng trực, cáp đôi dây xoắn, cáp quang, sóng radio, vi sóng và có thể sử dụng vệ tinh.

Trạm chủ hay trạm phụ (submaster) tập hợp dữ liệu từ các RTU khác nhau và cung cấp cho người vận hành giao diện để hiển thị thông tin và điều khiển tới các vị trí ở xa. Trong một hệ thống đo lường từ xa rộng lớn, các trạm phụ tổng hợp thông tin từ các vị trí ở xa và gửi trở về để điều khiển trạm chủ. (Hình 8.7)



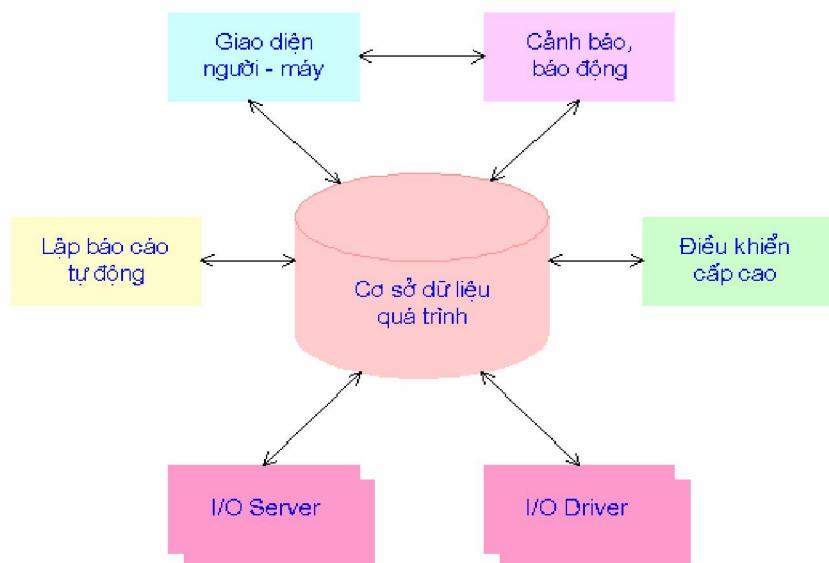
Hình 8.7 Cấu trúc của thiết bị đầu cuối (RTU)

Các RTU (remote terminal unit) gọi là thiết bị đầu cuối, nhiệm vụ thu thập dữ liệu và truyền dữ liệu đến các trạm chủ, cung cấp giao tiếp các tín hiệu số và tương tự tại mỗi vị trí từ xa.

### 8.3.3 Phần mềm

Phần mềm cho hệ thống SCADA có 2 dạng: có bản quyền và mã nguồn mở. Phần mềm có bản quyền được các nhà sản xuất bán cùng với hệ thống phần cứng. Các phần mềm mã nguồn mở được sử dụng nhiều hơn. Vì phần mềm có bản quyền sẽ phải mua giá cao và phần mềm mã nguồn mở có thể dùng được với nhiều hệ thống của nhiều nhà sản xuất khác nhau, khả năng tương thích của nó khá tốt.

Citect and WonderWare là 2 phần mềm mã nguồn mở được dùng phổ biến trong scada. Một vài phần mềm chứa thông tin quản lý tài nguyên của hệ thống SCADA.



Hình 8.8 Cấu trúc phần mềm của hệ thống SCADA

➤ **Một phần mềm SCADA bao gồm:** (Hình 8.8)

- Cơ sở dữ liệu.
- Giao diện người dùng.
- Hiển thị dạng đồ họa.
- Chức năng cảnh báo.
- Lập báo cáo tự động.
- Truy xuất dữ liệu.
- Hệ thống mạng.
- Xử lý phân tán.

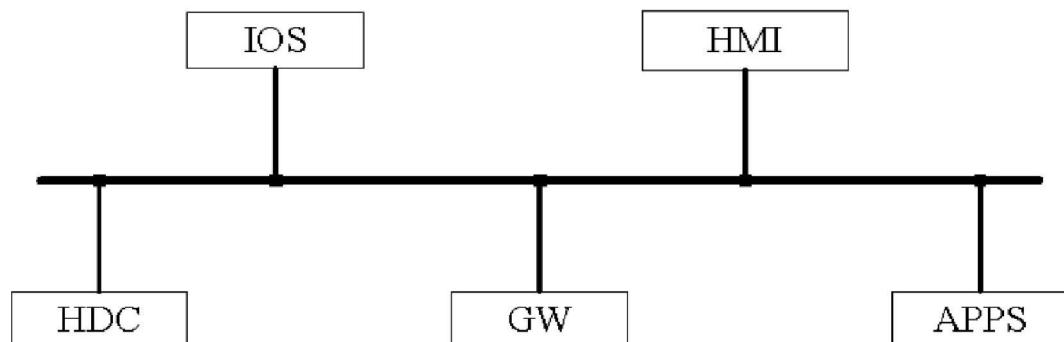
#### 8.4 Đặc tính chính của hệ thống SCADA

##### 8.4.1 Kiến trúc hệ thống

Hiện nay các hệ SCADA thế hệ mới được xây dựng theo kiến trúc phân bố, trong đó công suất của máy chủ được phân bổ trên một số bộ xử lý được nối cùng với nhau bằng mạng cục bộ (LAN). Trong đó mỗi bộ xử lý có một trách nhiệm nhất định như: thu thập và xử lý, tạo hiển thị, thiết lập báo cáo v.v... và một số bộ xử lý dùng để dự phòng.

Nhờ vào các công nghệ mạng điện rộng (WAN), hệ thống SCADA có thể được phân bố trên một miền địa lý rộng.

Hệ thống được thiết kế theo hệ thống mở và cơ cấu Server\_Client.



Hình 8.9 Kiến trúc phân bố tiêu biểu của một hệ SCADA

Trong đó:

IOS	Data Input/Output Module.
HMI	Human Machine Interface.
HDC	Historical Data Collection and Storage Module.
GW	Gateway for inter LAN communications.
APPS	Application Calculations and Processing Module.
HMI	HumanMachine Interface.

SCADA trang bị các VDU (Visual Display Unit - Đơn vị hiển thị) đầy đủ đồ họa có kèm mouse, bàn phím... Các nút điều khiển nối cứng được thay thế bằng các biểu tượng trên màn hình được tác động bằng chuột hay bàn phím. Các phương tiện này giúp cho người điều hành thực hiện:

- Nhanh chóng hoán đổi giữa các màn hình hiển thị.
- Nhanh chóng xem được chi tiết các thông tin.
- Tạo và sửa đổi các màn hình hiển thị trực tiếp ở hệ thống.
- Có những hiệu ứng đặc biệt để tăng khả năng phân biệt rõ dữ liệu (ví dụ các màu khác nhau cho các lô khác nhau trong một đường ống dẫn dầu).

Với các VDU hoạt động trong môi trường Windows, người điều hành cũng có thể:

- Xem trên cùng một VDU nhiều mảng thông tin mà thường trải ra trên nhiều màn hình hiển thị.
- Trên cùng màn hình, truy cập được các dữ liệu nằm rải rác theo vùng hoặc các dữ liệu thuộc các cơ sở dữ liệu khác nhau.

Do đó, những phương tiện HMI hiện đại cho phép người điều hành truy cập số lượng thông tin lớn dễ hơn và nhanh hơn.

#### **8.4.2 Các đơn vị đo lường từ xa RTU (Remote Terminal Unit)**

Các RTU không còn là những thiết bị thu động nữa, mà mục đích chính của nó là thu thập và lưu trữ dữ liệu vùng. Nhiều mức xử lý dữ liệu và điều khiển được thực hiện ở các RTU.

Nhiều loại thiết bị có thể được nối vào các RTU như: PLC, máy đo lưu lượng, thiết bị lấy chuẩn và đo bồn chứa...

Các RTU có thể được sắp xếp theo kiến trúc phân bố hoặc phân cấp. Dữ liệu của các RTU được xử lý tại trạm chủ.

#### **8.4.3 Cơ sở dữ liệu**

Dữ liệu được lưu trữ không những là dữ liệu đo đặc từ xa, được tính toán mà còn có các thông số báo về các sự kiện và các cảnh báo. Do bản chất phân bố của hệ SCADA, cơ sở dữ liệu cũng phân bố.

Về vấn đề bảo mật thì cho nhiều cấp truy cập sử dụng: điều hành viên, giám sát viên, huấn luyện, kỹ sư.

Cơ sở dữ liệu cũng liên hệ với hệ thống tin quản trị MIS (Geographic Information System).

#### **8.4.4 Truyền thông máy tính**

Truyền thông LAN (Local area network) / WAN (wide area network) giữa các máy tính dựa trên một số tiêu chuẩn công nghiệp phù hợp với mô hình OSI (Open System Interconnection).

Hiện nay đã có các công nghệ: Fast Internet, FDDI (Fiber Distributed Data Interface), Frame Relay...

## 8.5 Ứng dụng của hệ thống SCADA

SCADA được ứng dụng rộng rãi trong các lĩnh vực như: tàu biển, công nghiệp nặng, trong các tòa nhà hiện đại...

### 8.5.1 Ưu điểm của hệ thống SCADA

- Tăng khả năng điều khiển và xử lý của hệ thống.
- Hệ thống vận hành nhanh hơn, chính xác hơn.
- Bảo đảm an toàn cho người vận hành.
- Bảo vệ tốt các thiết bị khi xảy ra lỗi trong hệ thống.
- Tăng năng suất, độ tin cậy.
- Tiết kiệm năng lượng.
- Tốn ít chi phí vận hành hơn.

### 8.5.2 Nhược điểm của hệ thống SCADA

- Tốn nhiều chi phí đầu tư xây dựng và thiết kế.
- Trình độ chuyên môn đòi hỏi cao.

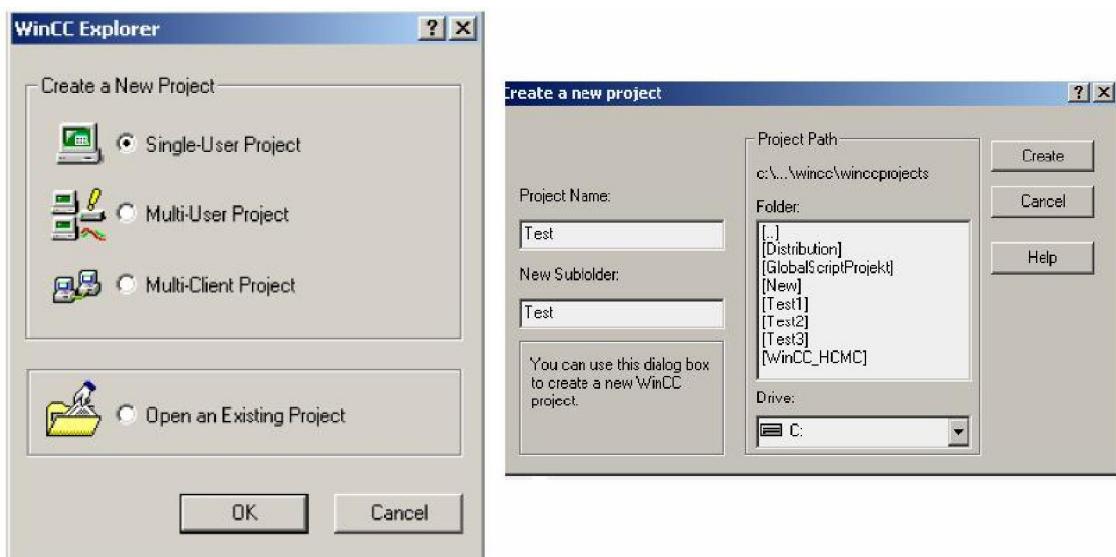
## CHƯƠNG 9

### GIỚI THIỆU PHẦN MỀM WINCC VÀ GIAO DIỆN CỦA ĐỀ TÀI

#### 9.1 Giới thiệu WinCC

WinCC (Windows Control Center) là phần mềm SCADA của hãng Siemens dùng để giám sát, thu thập dữ liệu và điều khiển một hệ thống tự động hóa. WinCC hỗ trợ người lập trình thiết kế giao diện người – máy HMI (Human Machine Interface) trong hệ thống SCADA (Supervisory Control And Data Acquisition) nhằm thực hiện chức năng giám sát, điều khiển và thu thập dữ liệu của quá trình sản xuất.

WinCC 7.0 hoạt động trên nền hệ điều hành Windows 2000, Windows XP, Windows Vista. Các hệ điều hành đều có khả năng thực hiện đa nhiệm vụ, đảm bảo phản ứng nhanh với việc xử lý ngắt và độ an toàn chống mất dữ liệu bên trong ở mức độ cao. WinCC đã và đang là chương trình HMI được sử dụng rộng rãi nhất tại Việt Nam.



Hình 9.1 Tạo một dự án WinCC

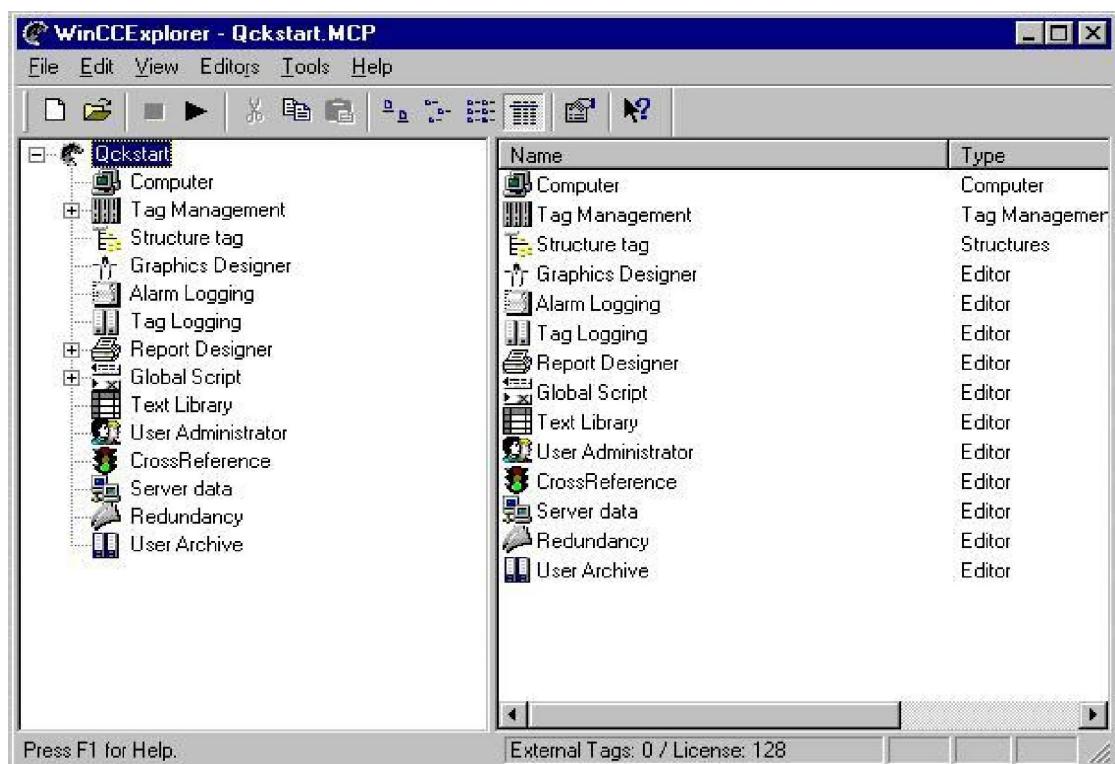
## 9.2 Đặc trưng cơ bản

WinCC đang được ứng dụng trong nhiều lĩnh vực khác nhau, phổ biến nhất là tự động hóa quá trình điều khiển và giám sát quy trình sản xuất. Khi một hệ thống sử dụng chương trình WinCC để điều khiển và thu thập dữ liệu từ quá trình, có thể mô phỏng các sự kiện xảy ra trong quá trình bằng các sơ đồ, hình ảnh trực quan.

Ngoài khả năng thích ứng cho việc xây dựng các hệ thống có qui mô nhỏ và vừa khác nhau, WinCC còn có thể dễ dàng tích hợp với những ứng dụng có qui mô lớn như hệ thống quản lý việc thực hiện sản xuất MES (Manufacturing Execution System) và hệ thống ERP (Enterprise Resource Planning).

## 9.3 Chức năng của WinCC

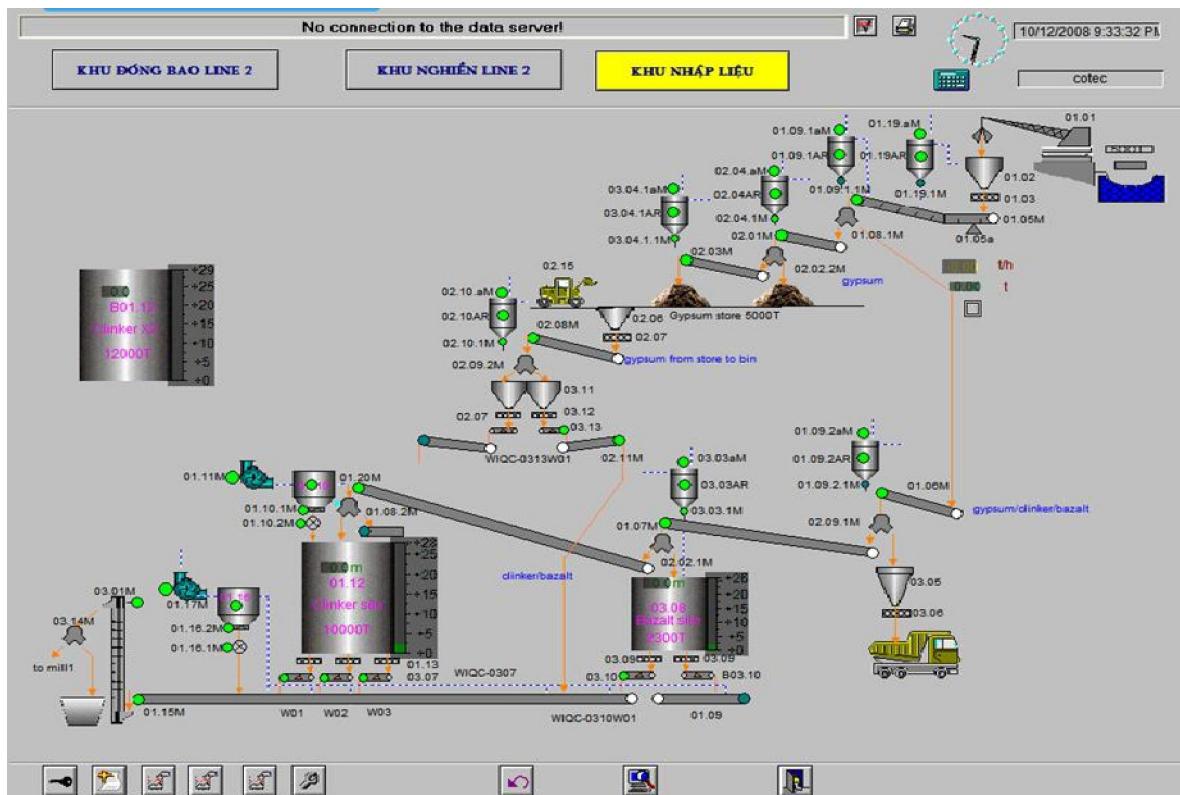
Khi sử dụng WinCC để thiết kế giao diện người- máy HMI và mạng SCADA, WinCC cung cấp các chức năng sau (Hình 9.2)



Hình 9.2 Cửa sổ làm việc WinCC Explorer

- **Graphics Designer:** thực hiện dễ dàng các chức năng mô phỏng thông qua các đối tượng đồ họa của chương trình như Windows, OLE, I/O,... với nhiều thuộc tính động.
- **Alarm Logging:** thực hiện hiển thị các thông báo, báo cáo khi hệ thống vận hành. Nó chứa các chức năng để nhận các thông báo từ quá trình để hiển thị, hồi đáp và lưu trữ. Alarm Logging còn được thiết kế để giúp dễ dàng tìm ra nguyên nhân lỗi.
- **Tag Logging:** cho phép thu thập dữ liệu quan trọng về tiêu chuẩn công nghệ từ các quá trình, lưu trữ, nén các giá trị đo dưới nhiều dạng khác nhau.
- **Report Designer:** có nhiệm vụ tạo các thông báo, báo cáo và các kết quả này được lưu trữ dưới dạng các trang nhật ký sự kiện.
- **User Archives:** cho phép người sử dụng lưu trữ dữ liệu từ chương trình ứng dụng và có khả năng trao đổi với các thiết bị tự động hóa khác. Điều này có nghĩa là các công thức, thông số trong chương trình WinCC có thể được soạn thảo, lưu trữ và sử dụng trong hệ thống.
- **Text Library :** chứa các văn bản tùy thuộc ngôn ngữ do người dùng tạo ra.
- **Global Scripts:** sử dụng ngôn ngữ C-Script để tạo các hàm, các thao tác giúp cho việc xử lý các sự kiện một cách mềm dẻo và linh hoạt.
- **User Administrator:** cho phép phân quyền các nhóm người sử dụng.

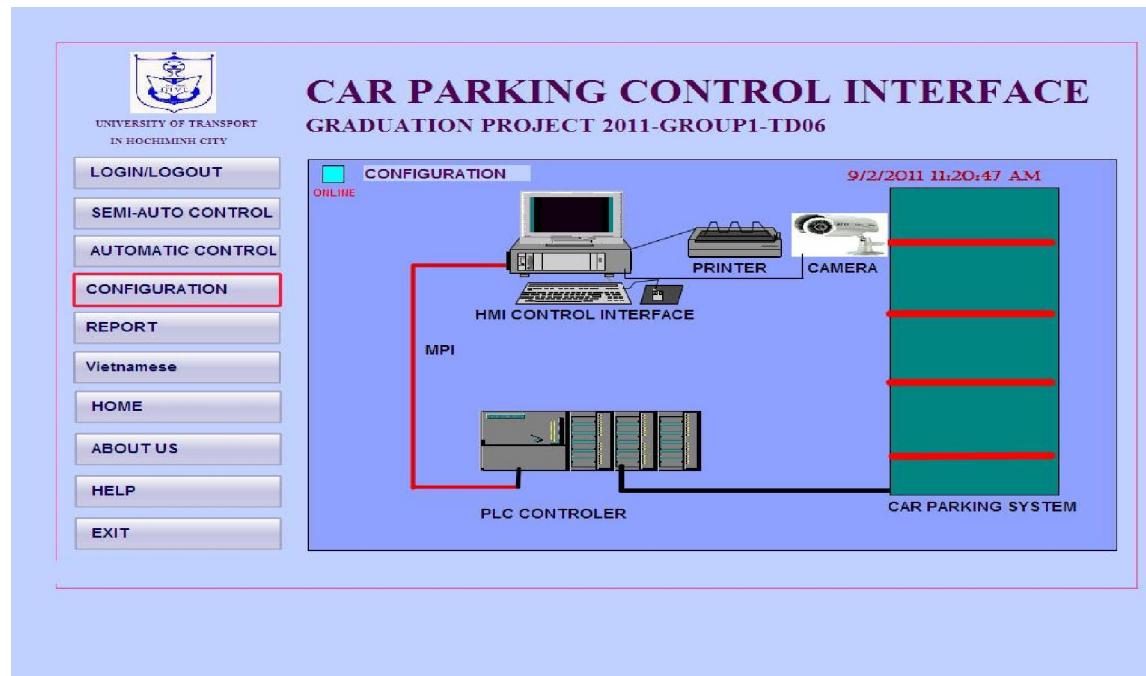
## 9.4 Một dự án mẫu của WinCC



Hình 9.3 Một giao diện HMI được tạo từ WinCC

## 9.5 Giao diện WinCC thiết kế cho đề tài – Bài đồ xe tự động

### 9.5.1 Sơ đồ tổng quan hệ thống nhà giữ xe ô tô tự động (Hình 9.4)



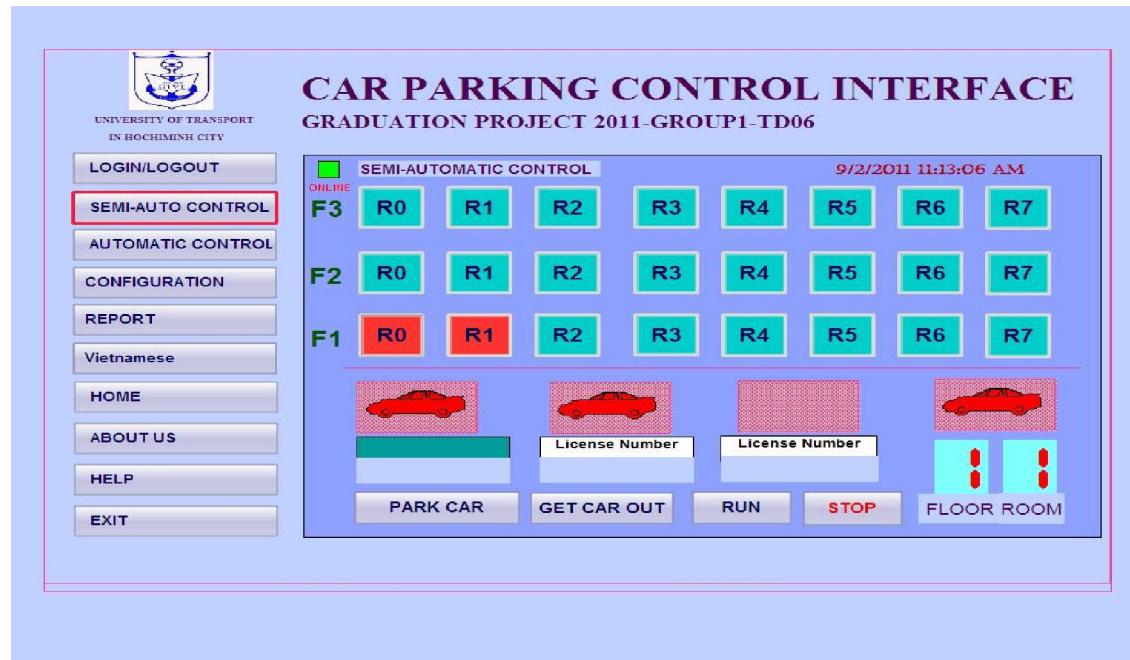
Hình 9.4 Sơ đồ tổng quan hệ thống nhà giữ xe ô tô tự động

### 9.5.2 Giao diện đăng nhập điều khiển hệ thống (Hình 9.5)



Hình 9.5 Giao diện đăng nhập điều khiển hệ thống

### 9.5.3 Giao diện điều khiển hệ thống bán tự động (Hình 9.6)



Hình 9.6 Giao diện điều khiển hệ thống bán tự động

### 9.5.4 Giao diện điều khiển hệ thống tự động (Hình 9.7)

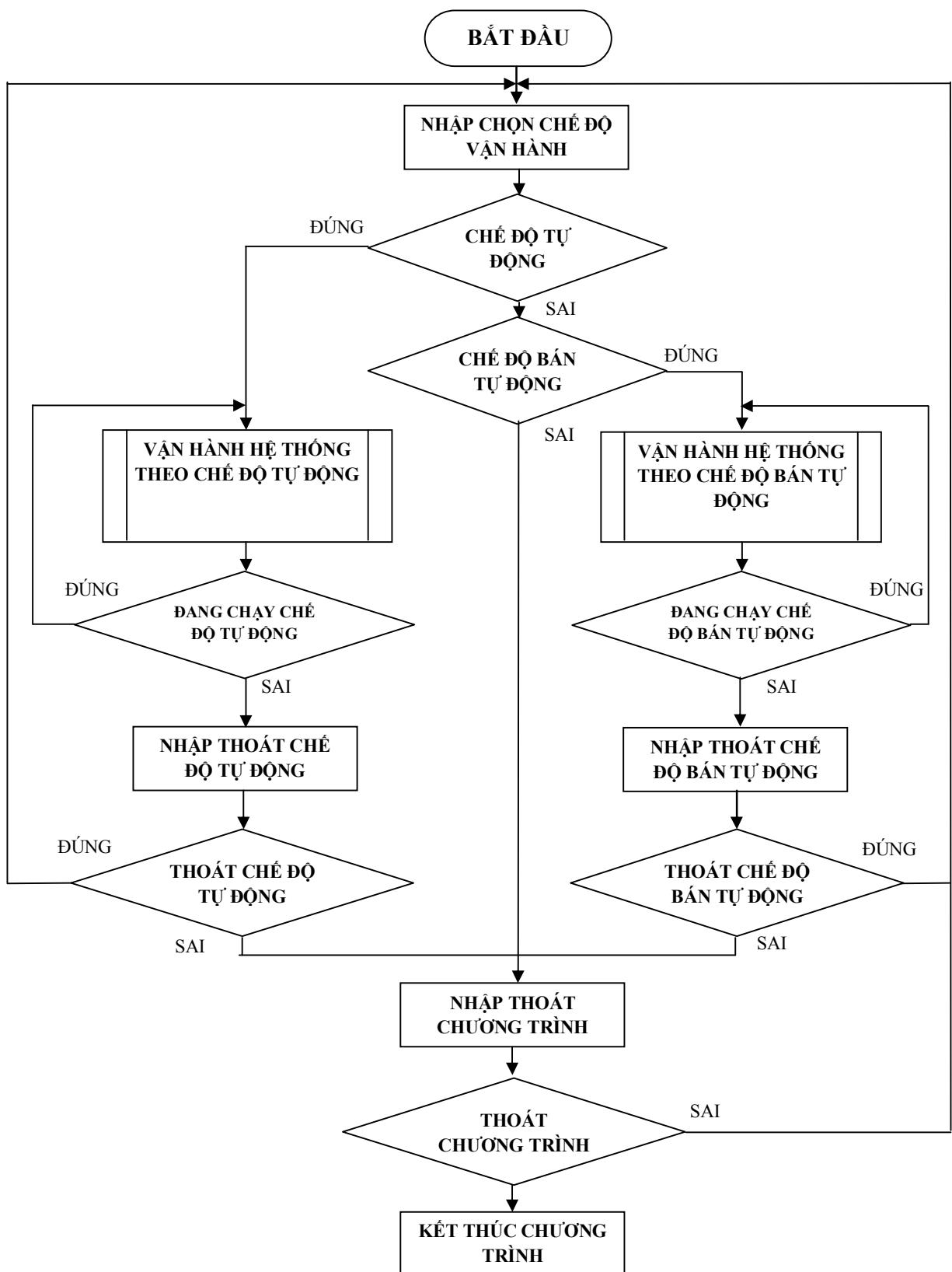


Hình 9.7 Giao diện điều khiển hệ thống bán tự động

## CHƯƠNG 10

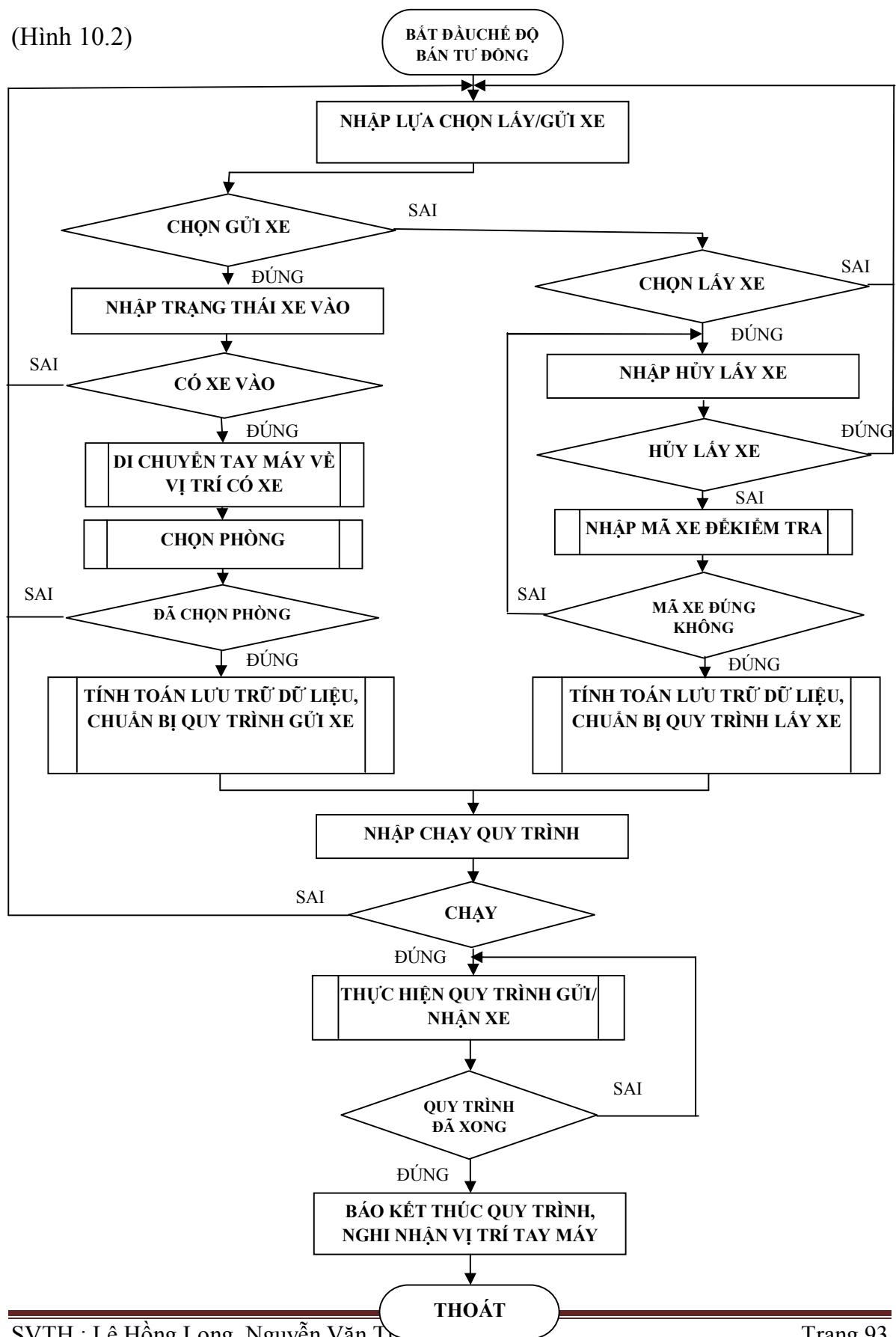
### GIẢI THUẬT ĐIỀU KHIỂN NHÀ GIỮ XE Ô TÔ TỰ ĐỘNG

- 10.1 Giải thuật điều khiển chương trình chính (Hình 10.1)**
- 10.2 Giải thuật chương trình điều khiển bán tự động (Hình 10.2)**
- 10.3 Giải thuật chương trình điều khiển tự động (Hình 10.3)**
- 10.4 Giải thuật xác định vị trí có xe gần nhất để di chuyển tay máy về lấy xe (Hình 10.4)**
- 10.5 Giải thuật đường đi ngắn nhất cho khoang (Hình 10.5)**
- 10.6 Chương trình con điều khiển động cơ bước (Hình 10.6)**

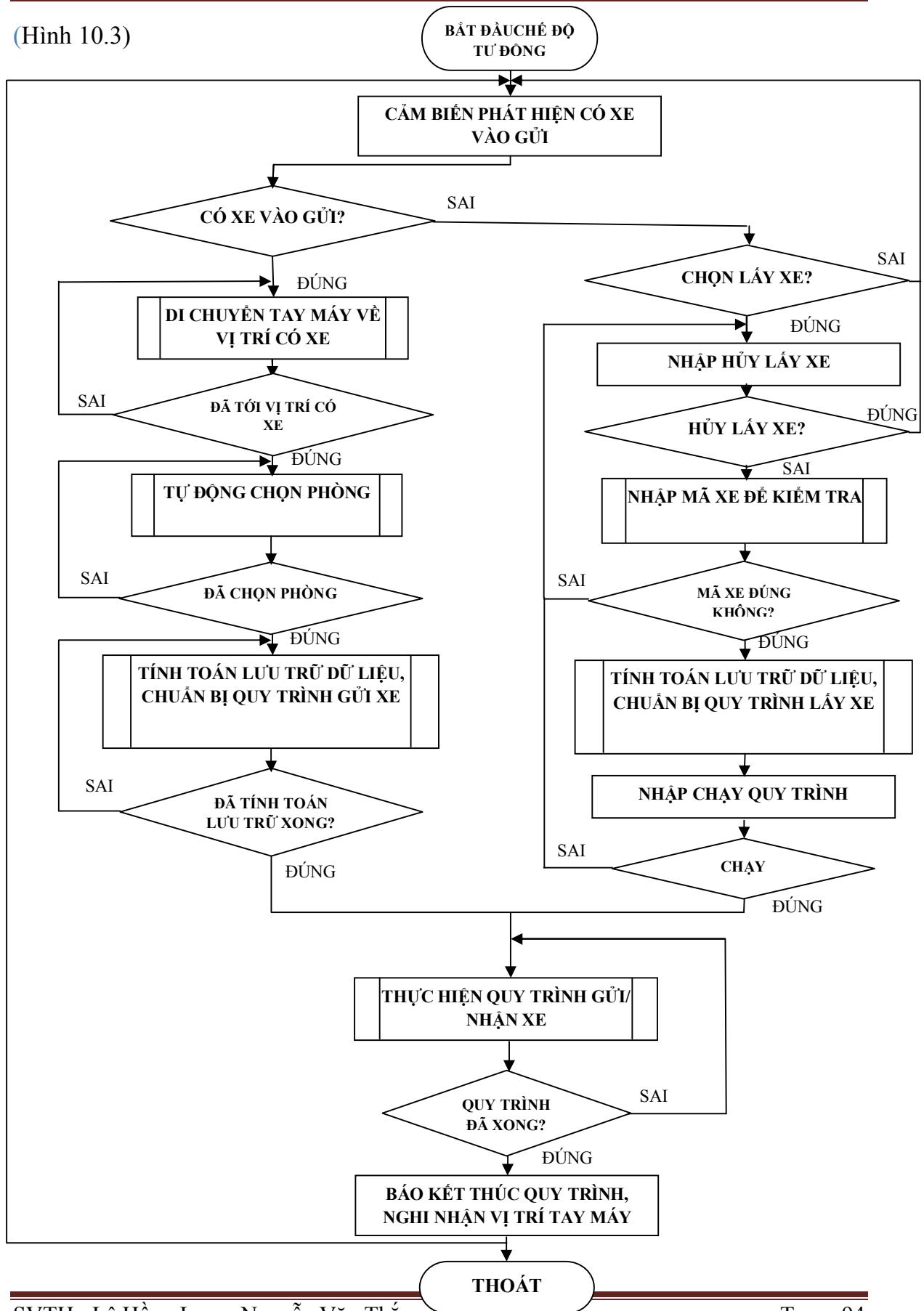


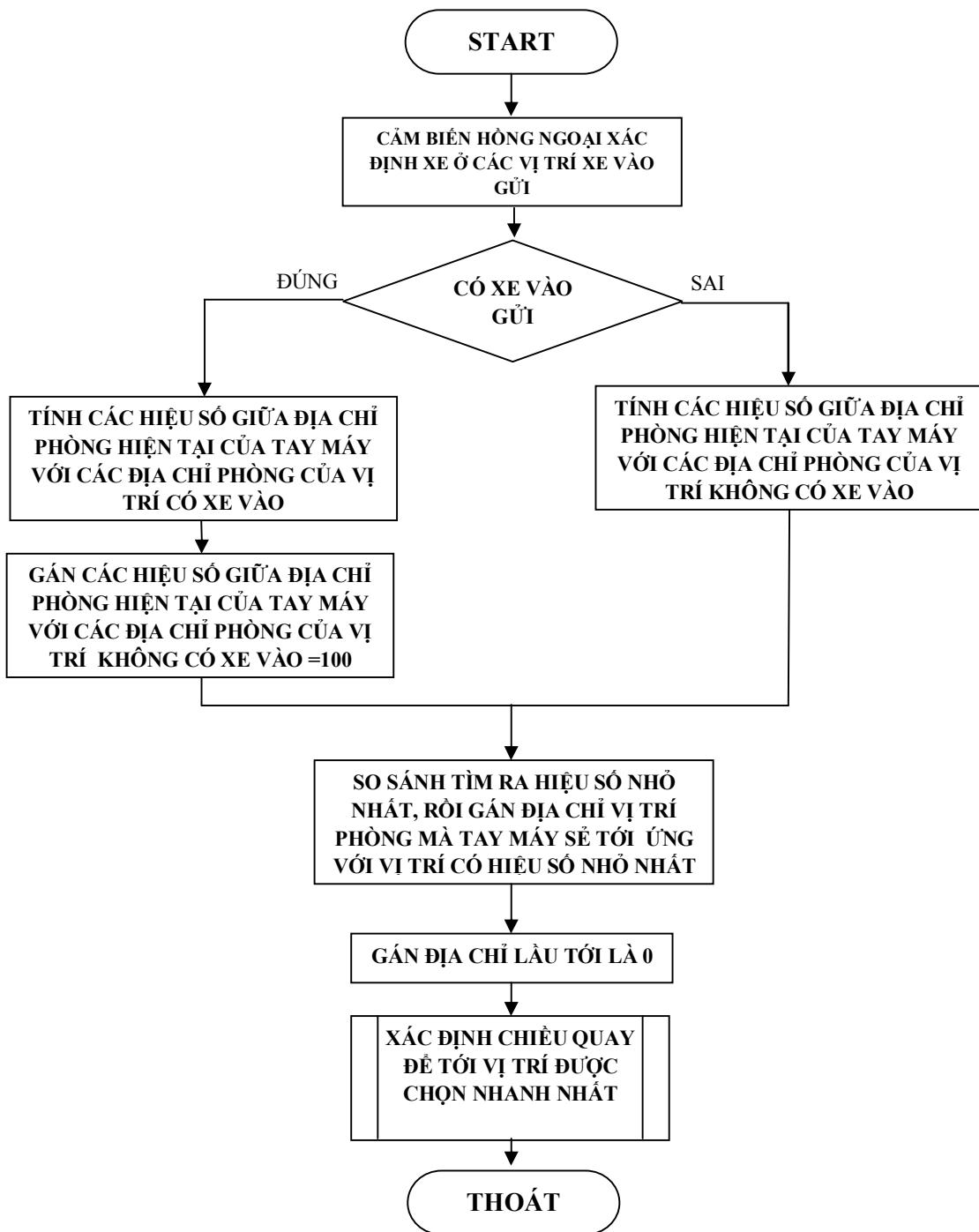
Hình 10.1 Giải thuật chương trình chính điều khiển nhà giữ xe ô tô tự động

(Hình 10.2)



(Hình 10.3)





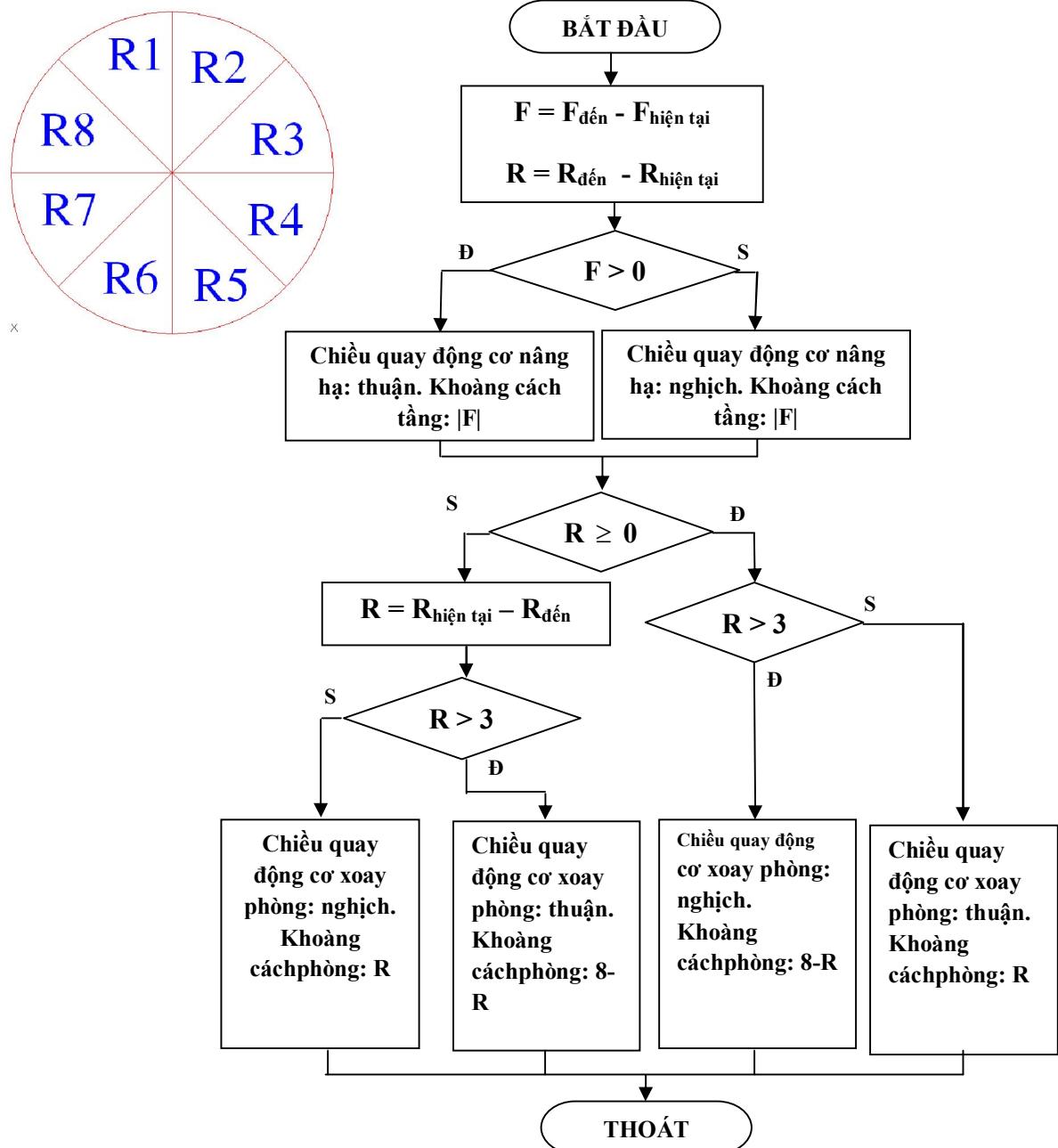
Hình 10.4 Giải thuật xác định vị trí có xe gần nhất để di chuyển tay máy về lấy xe

Với  $F$  là ký hiệu của tầng;  $F_{\text{đến}}$  : là tầng cần điều khiển đi đến;  $F_{\text{hiện tại}}$  : là tầng hiện tại đang đứng.

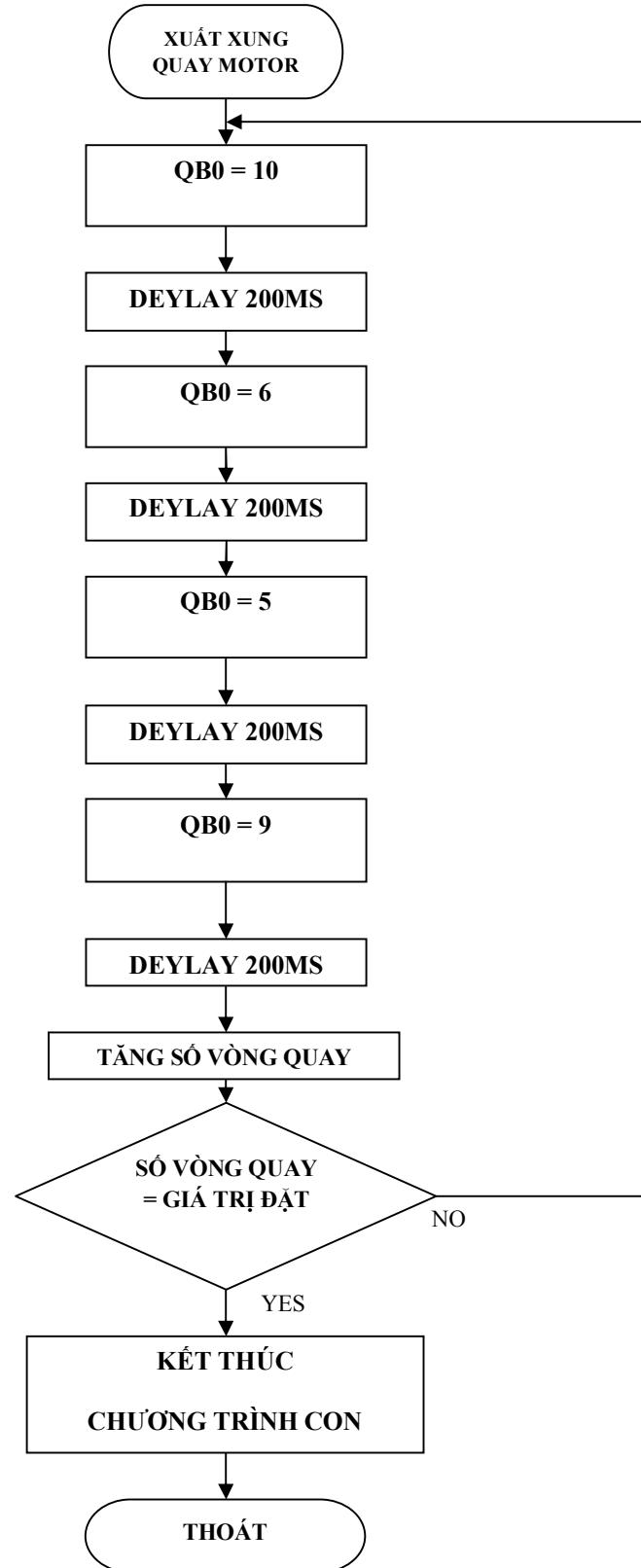
Với  $R$  là ký hiệu của khoang;  $R_{\text{đến}}$  : là khoang cần đi đến;  $R_{\text{hiện tại}}$  : là khoang hiện tại đang đứng.

CW : là ký hiệu của sự quay cùng chiều kim đồng hồ.

NW : là ký hiệu của sự quay ngược chiều kim đồng hồ.



Hình 10.5 Chương trình con chọn đường đi ngắn nhất cho khoang

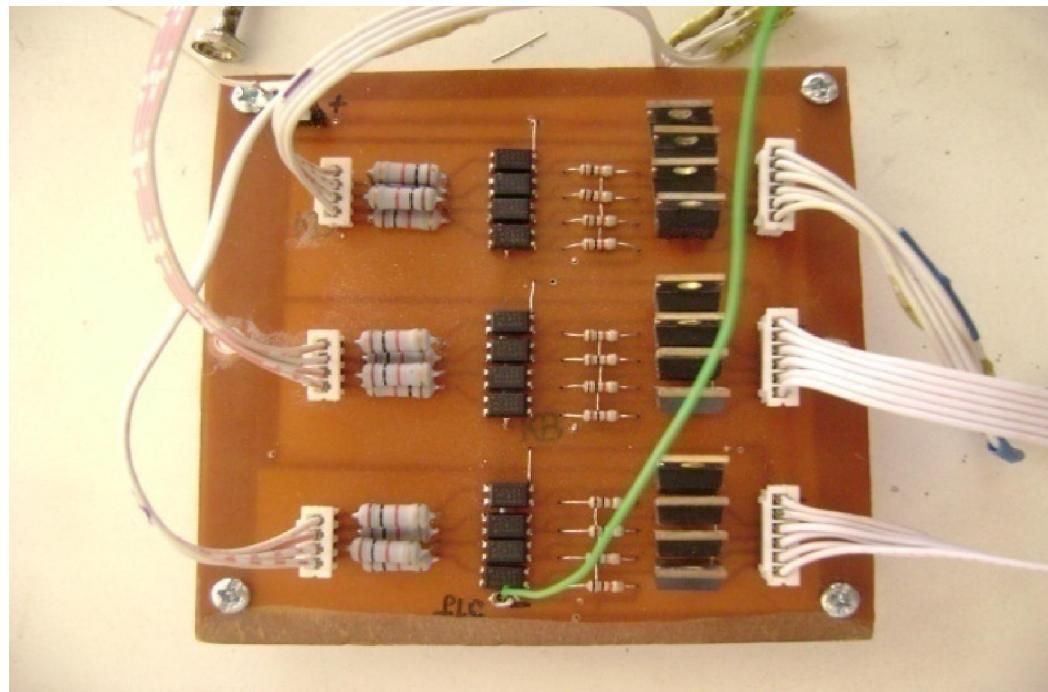


Hình 10.6 Chương trình con điều khiển động cơ bước

## CHƯƠNG 11

### THIẾT KẾ VÀ THI CÔNG MÔ HÌNH PHẦN CỨNG

#### 11.1 Thiết kế mạch lái điều khiển 3 động cơ bước (Hình 11.1)



Hình 11.1 Mạch lái điều khiển 3 động cơ bước

#### 11.2 Thiết kế cánh tay máy lấy xe (Hình 11.2)



Hình 11.2 Mạch lái điều khiển 3 động cơ bước

### 11.3 Mô hình nhà giữ xe ô tô tự động (Hình 11.3)



Hình 11.3 Mô hình nhà giữ xe ô tô tự động

## CHƯƠNG 12

### KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

#### 12.1 Kết quả đạt được

- ❖ Nhận dạng được bảng số xe ô tô ở Việt Nam.
- ❖ Liên kết được Matlab và WinCC thông qua giao thức OPC.
- ❖ Thiết kế và thi công mạch lái điều khiển động cơ bước.
- ❖ Lập trình điều khiển nhà giữ xe ô tô tự động bằng PLC S7 – 300.
- ❖ Thiết kế giao diện SCADA WinCC để điều khiển quá trình hoạt động của hệ thống nhà giữ xe ô tô tự động.
- ❖ Thiết kế và thi công mô hình nhà giữ xe ô tô tự động nhiều ngõ vào.
- ❖ Xuất Excel thống kê báo cáo thu chi của hệ thống nhà giữ xe ô tô tự động.

#### 12.2 Hạn chế của đề tài

Vì kiến thức chuyên môn còn hạn hẹp, thời gian thực hiện đề tài gấp rút nên không tránh những thiếu sót và còn nhiều mặt hạn chế cần khắc phục:

- ❖ Hệ thống nhà giữ xe thiết kế trong đề tài chỉ có một cổng trả xe nên sẽ gây ra tình trạng chờ đợi khi cùng lúc xử lý quá nhiều thao tác trả xe.
- ❖ Chưa tính toán được cơ cấu chi tiết của hệ thống nhà giữ xe ô tô thực tế.
- ❖ Thiếu thiết bị camera chuyên dụng có độ phân giải cao dùng để chụp ảnh đầu vào nhận dạng biển số xe.

### 12.3 Hướng phát triển đề tài

Để hệ thống có thể ứng dụng tốt trong thực tiễn ta cần phải phát triển thêm một số đặc điểm sau:

- ❖ Xây dựng hệ thống có nhiều cửa trả xe khi trường hợp thực hiện nhiều thao tác trả xe ngay cùng một thời điểm.
- ❖ Thay thế mã lấy xe bằng một hệ thống in ấn thẻ từ - mã hóa thông tin xe nhằm nâng cao tính bảo mật của hệ thống. Bên cạnh đó cần tích hợp luôn việc thanh toán phí gửi xe hoàn toàn tự động. Cụ thể là kết hợp công nghệ RFID (Radio Frequency Identification – công nghệ nhận dạng tần số Radio) để triển khai.
- ❖ Tính toán thông số kỹ thuật và chọn thiết bị cho một hệ thống nhà giữ xe ô tô tự động trong thực tế.

## TÀI LIỆU THAM KHẢO

### Tiếng việt

[01] Đoàn Hiệp – Các phương pháp điều khiển động cơ bước

[02] Vũ Đức Hậu – Luận văn tốt nghiệp: SCADA cho Nhà giữ xe ô tô tự động

[03] GS.NguyenVanNgo – Giáo trình xử lý ảnh

[04] Tài liệu hướng dẫn thiết kế mạch in Eagle

[05] Tran Duc Tuan, Tran Le Hong Du, Tran Vinh Phuoc, Nguyen Viet Hoang – Building an Automatic Vehicle License - Plate recognition System (Bài báo khoa học).

[06] [www.picvietnam.com](http://www.picvietnam.com) & [www.dientuvietnam.com](http://www.dientuvietnam.com)

### Tiếng Anh

[01] Siemens – Manual PLC S7 – 300

[02] Siemens – WinCC V 7.0

[03] Gonzalez – Digital Image Processing Using Matlab

[04] Help Matlab

[05] Help WinCC – VBS for Excel

## PHỤ LỤC

### CHƯƠNG TRÌNH MATLAB

#### CHƯƠNG TRÌNH CHÍNH

```

function varargout = RECOGNITION_VLP(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',     mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @RECOGNITION_VLP_OpeningFcn, ...
    'gui_OutputFcn', @RECOGNITION_VLP_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function RECOGNITION_VLP_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
global xungcanhlen
xungcanhlen=0;
global mark_end check_M_cardoor1 check_M_cardoor2 check_M_cardoor3
mark_end=0;
check_M_cardoor1=0;
check_M_cardoor2=0;
check_M_cardoor3=0;
global close_value camera_run
camera_run=0;
global mark
if close_value==1
    mark=1;
    close_value=0;
end
global enable_OPc
global wincc_takeimage
wincc_takeimage=0;

```

```

enable_OPc=0;
global letter
letter=[];
global m;
m=0;
global Soluong_chuso_dangky
Soluong_chuso_dangky=0;
global take_image_value
take_image_value=0;
global angle_im fl re
angle_im=0;
%im_texto=[];
fl = [];
re = [];

function varargout = RECOGNITION_VLP_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

% --- Executes on button press in Load_image_1.
function Load_image_1_Callback(hObject, eventdata, handles)
% plate one row : dimension 47.5*11
% plate two rows: dimension 28*21
global biensoxe1 biensoxe2 biensoxe3 O_cardoor1 O_cardoor2 O_cardoor3
global enable_OPc
if enable_OPc==1
da = opcda('localhost','OPCServer.WinCC.1');%ket noi WinCC
connect(da);
nhom_bien = addgroup(da,'ExReadAsync');%Tao nhom bien WinCC
biensoxe1=additem(nhom_bien,'O_BSX1');
biensoxe2=additem(nhom_bien,'O_BSX2');
biensoxe3=additem(nhom_bien,'O_BSX3');
O_cardoor1=additem(nhom_bien,'O_cardoor1');
O_cardoor2=additem(nhom_bien,'O_cardoor2');
O_cardoor3=additem(nhom_bien,'O_cardoor3');
end
global check_camera camera_run
if camera_run==0
check_camera=0;
end
camera_run=0;
global word_dot;

```

```
word_dot=[];
t='waiting';
set(handles.text13,'string',t);
%word='80N-6789';
%set(handles.text6,'string',word);
axes(handles.axes1);
cla reset ;
axes(handles.axes2);
cla reset ;
axes(handles.axes3);
cla reset ;
axes(handles.axes4);
cla reset ;
axes(handles.axes5);
cla reset ;
axes(handles.axes6);
cla reset ;
global S1 ;
global fillgap_horizontal
global minlength_horizontal
global fillgap_vertical
global minlength_vertical
global options_recognition mark;
global limit_vertical
global limit_horizontal
global number_lines_horizontal
global number_lines_vertical
global length_crop_width_1
global length_crop_width_2
global err_value
err_value=55; %55
length_crop_width_1=25;
length_crop_width_2=15;
number_lines_vertical=2;
number_lines_horizontal=6;%6
limit_horizontal=45;
limit_vertical=100;
fillgap_horizontal=200; %200
minlength_horizontal=147;
fillgap_vertical=30;
minlength_vertical=100;
```

```
if mark==1
    options_recognition=1;
else
    options_recognition=2;
end
[filename, pathname] = uigetfile({'*.jpg'; '*.bmp'; '*.gif'; '*.*'}, 'Pick an Image File');
set(handles.text9,'string',filename);
S= imread([pathname,filename]);
axes(handles.axes1);
imshow(S);
[y,x,z]=size(S);
S1 = imcrop(S,[x/4 y/4 x/2 y/2]);
axes(handles.axes2);
imshow(S1);

function Load_image_2_Callback(hObject, eventdata, handles)
% --- Executes on button press in From_Camera.v
global biensoxe
global enable_OPc
if enable_OPc==1
da = opcda('localhost','OPCServer.WinCC.1');%ket noi WinCC
connect(da);
nhom_bien = addgroup(da,'ExReadAsync');%Tao nhom bien WinCC
biensoxe=additem(nhom_bien,'O_bsx');
end
global check_camera camera_run
if camera_run==0
check_camera=0;
end
camera_run=0;
global word_dot;
word_dot=[];
%word='80N-6789';
global word
set(handles.text6,'string',word);
axes(handles.axes1);
cla reset ;
axes(handles.axes2);
cla reset ;
axes(handles.axes3);
cla reset ;
axes(handles.axes4);
cla reset ;
```

```
axes(handles.axes4);
cla reset ;
axes(handles.axes5);
cla reset ;
axes(handles.axes6);
cla reset ;
global S1 S;
global options_recognition mark;
global fillgap_horizontal
global minlength_horizontal
global fillgap_vertical
global minlength_vertical
global limit_vertical
global limit_horizontal
global number_lines_horizontal
global number_lines_vertical
global length_crop_width_1
global length_crop_width_2
global err_value
err_value=55;
length_crop_width_1=25;
length_crop_width_2=25;
number_lines_vertical=2;
number_lines_horizontal=3;
limit_horizontal=230;
limit_vertical=230;
fillgap_horizontal=30;
minlength_horizontal=30;
fillgap_vertical=30;
minlength_vertical=30;
if mark==1
    options_recognition=1;
else
    options_recognition=2;
end
[filename, pathname] = uigetfile({'*.jpg'; '*.bmp'; '*.gif'; '*.*'}, 'Pick an Image File');
set(handles.text9,'string',filename);
S= imread([pathname,filename]);
axes(handles.axes1);
imshow(S);
if mark==1
    S1=S;
else
```

```
[y,x,z]=size(S);
S1 = imcrop(S,[x/4 y/4 x/2 y/2]);
end
axes(handles.axes2);
imshow(S1);

function From_Camera_Callback(hObject, eventdata, handles)
global nhom_bien
global enable_OPc
if enable_OPc==1
da = opcda('localhost','OPCServer.WinCC.1');%ket noi WinCC
connect(da);
nhom_bien = addgroup(da,'ExReadAsync');%Tao nhom bien WinCC
camera1=additem(nhom_bien,'O_CAMERA');

camera1.Value=0;
camera1.OnChangeFcn='From_Camera_Callback';
camera1.Visible='on';

end
global take_image_value
global check_camera camera_run
check_camera=1;
camera_run=1;
global word_dot;
word_dot=[];
global options_recognition_mark;
global fillgap_horizontal
global minlength_horizontal
global fillgap_vertical
global minlength_vertical
global limit_vertical
global limit_horizontal
global number_lines_horizontal
global number_lines_vertical
global length_crop_width_1
global length_crop_width_2
global err_value
err_value=20;
length_crop_width_1=25;
length_crop_width_2=25;
number_lines_vertical=3;
number_lines_horizontal=3;
limit_horizontal=220;
limit_vertical=200;
fillgap_horizontal=30;
```

```
minlength_horizontal=30;
fillgap_vertical=25;
minlength_vertical=30;
if mark==1
    options_recognition=1;
else
    options_recognition=2;
end
global word;
%word='80N-6789';
set(handles.text6,'string',word);
axes(handles.axes1);
cla reset ;
axes(handles.axes2);
cla reset ;
axes(handles.axes3);
cla reset ;
axes(handles.axes4);
cla reset ;
axes(handles.axes1);
cla reset ;
global webcam
webcam=videoinput('winvideo',2);
figure('Toolbar','none',...
    'Menubar','none',...
    'NumberTitle','Off',...
    'Name','RECOGNITION VLP');
%uicontrol('String', 'Close', 'Callback', 'close(gcf)');
%axes(handles.axes1);
close(gcf);
vidRes = get(webcam, 'VideoResolution');
nBands = get(webcam, 'NumberOfBands');
hImage = image( zeros(vidRes(2), vidRes(1), nBands) );
preview(webcam, hImage);
guidata(hObject, handles);
global wincc_takeimage
%connect OPC WinCC

%%%
off_while=3;

while off_while > 2
    preview(webcam, hImage);
```

```

if enable_OPc==1
    wincc_takeimage=struct2cell(read(camera1));
    wincc_takeimage=wincc_takeimage{2};
    end
global xungcanhlen
    if wincc_takeimage==0
        xungcanhlen=0;
    end
if xungcanhlen==0
    if wincc_takeimage==1 || take_image_value==1
        off_while=1;
        take_image_value=0;
        xungcanhlen=1;

        delete(webcam);
        Load_image_1_Callback(hObject, eventdata, handles)
        %if mark==1
        % S1=S;
        %else
        % [y,x,z]=size(S);
        % S1 = imcrop(S,[x/4 y/4 x/2 y/2]);
        %end
        %axes(handles.axes2);
        %imshow(S1);
        end
    end
% --- Executes on button press in Take_image.
function Take_image_Callback(hObject, eventdata, handles)
global take_image_value
take_image_value=1;

% --- Executes on button press in Export_image.
function Export_image_Callback(hObject, eventdata, handles)
global m; % define global at open window and set m =0;
global S2 ;% export image
m=m+1;
var=strcat('bsx',num2str(m));
var=strcat(var,'.jpg');
imwrite(S2,var,'jpg');

% --- Executes on button press in Extract_vlp.
function Extract_vlp_Callback(hObject, eventdata, handles)

```

```

global BW G;
global S1 typeplate;
global typeplate_write
typeplate='Type of plate';
G=rgb2gray(S1);
xy_long=0;
i=0;
global options_recognition;
global imagen;
are_process=0;
loop_option=0; % vong lap de nhan dang 2 kieu bien so xe
%-----HORIZONTAL-----
while loop_option < 2
loop_option=loop_option+1;
if options_recognition==1 % nhan dang bien so xe theo chieu ngang truoc
BW = edge(G,'sobel',[],'horizontal'); %vertical
%BW=wiener2(BW,[5 5]);
axes(handles.axes3);
imshow(BW);
global number_lines_horizontal
[H,T,R] = hough(BW);
P = houghpeaks(H,number_lines_horizontal);
global fillgap_horizontal
global minlength_horizontal

lines =
houghlines(BW,T,R,P,'FillGap',fillgap_horizontal,'MinLength',minlength_horizontal);%200 - 147
axes(handles.axes2);
imshow(S1); hold on;
i = 0;
max_len = 0;
wid1=0;
hei1=0;
wid2=0;
hei2=0;
xy1=[0 0;0 0];
xy2=[0 0;0 0];
if length(lines) == 1 % Luot qua truong hop ko tim duoc bat ky duong thang nao
else
for k = 1:length(lines)
xyk = [lines(k).point1; lines(k).point2];

```

```

plot(xyk(:,1),xyk(:,2),'LineWidth',2,'Color','green');
% Plot beginnings and ends of lines
plot(xyk(1,1),xyk(1,2),'x','LineWidth',2,'Color','yellow');
plot(xyk(2,1),xyk(2,2),'x','LineWidth',2,'Color','red');
for j=(k+1):length(lines)
    xyj = [lines(j).point1; lines(j).point2];
    lenk(k) = norm(lines(k).point1 - lines(k).point2);
    lenj(j) = norm(lines(j).point1 - lines(j).point2);
    lenkj(k+j)=norm(xyk(1,2) - xyj(1,2)); % (1,2) là tọa độ cua Y, (1,1) là tọa độ
cua X
    tyle1=(lenk(k)/lenkj(k+j));
    tyle2=(lenj(j)/lenkj(k+j));
    global err_value
    err = norm(xyk(1,1) - xyj(1,1));
    if err < err_value
        if (tyle1 > 3.5 && tyle1 <5) %|| (tyle1 > 1.2 && tyle1 <1.8)
            i = i +1 ;
            wid1 = lenk(k);
            hei1 = lenkj(k+j);
            if xyk(1,2)< xyj(1,2)
                xy1 = xyk;
            else
                xy1 = xyj;
            end
        else
            if (tyle2 > 3.5 && tyle2 <5) %|| (tyle2 > 1.2 && tyle2 <1.8)
                i = i +1 ;
                wid1 = lenj(j);
                hei1 = lenkj(k+j);
                if xyk(1,2)< xyj(1,2)
                    xy1 = xyk;
                else
                    xy1 = xyj;
                end
            end
        end
        if i>0
            image_crop=imcrop(S1,[xy1(1,1) xy1(1,2) wid1 hei1]);
        global length_crop_width_1
        global length_crop_width_2
        % modify angle error for one row plate license
        if xy1(1,2)==xy1(2,2)
            angle_im=0;
        end
    end
end

```

```

angle='0';
image_crop_rotate=image_crop;

image_crop_rotate_cut=imcrop(S1,[xy1(1,1)+(1/length_crop_width_1)*wid1)
(xy1(1,2)+(1/8)*hei1) (wid1-(1/length_crop_width_2)*wid1) (hei1-(1/5)*hei1)]);
%Kiem tra loai bang so nen trang hay nen xanh
%image_crop_rotate_cut=imadjust(image_crop_rotate_cut);
level = graythresh(image_crop_rotate_cut);
imagen=im2bw(image_crop_rotate_cut,level);
value0=0;
value1=0;
%tim cac gia tri bang 1 va 0 de so sanh
for ii=1:(size(imagen,1)/4)
    for jj=1:size(imagen,2)
        if imagen(ii,jj)==1
            value1=value1+1;
        else
            value0=value0+1;
        end
    end
end
if value1 > value0
    typeplate='White Background Plate';
    typeplate_write='White';
    imagen=~im2bw(image_crop_rotate_cut,level);%level
else
    typeplate='Blue Background Plate';
    typeplate_write='Blue';
    imagen=im2bw(image_crop_rotate_cut,level); %level
end
%ket thuc viec xac dinh loai bien so xe la nen trang hoac xanh
else
    sin_im=(norm(xy1(1,2) - xy1(2,2)))/(norm(xy1(1,1) - xy1(2,1)));% tinh sin
    goc lech --> chinh do nghieng anh
    angle_im=(asin(sin_im)*180)/3.14; % tinh goc de xoay anh
    if xy1(1,2)>xy1(2,2)
        S1_rotate=imrotate(S1,-angle_im,'bilinear','crop');
        angle=num2str(angle_im);
    else
        S1_rotate=imrotate(S1,angle_im,'bilinear','crop');
        angle=strcat(' ',num2str(angle_im));
    end
    %chuong trinh con de tim cac thong so moi cho xy1

```

```
%chuong trinh con de tim cac thong so moi cho xy1
% ket thuc chuong trinh con tim lai cac thong so moi
% ket thuc chuong trinh con tim lai cac thong so moi
image_crop_rotate=imcrop(S1_rotate,[xy1(1,1) xy1(1,2) wid1 hei1]);

image_crop_rotate_cut=imcrop(S1_rotate,[(xy1(1,1)+(1/length_crop_width_1)*wid
1) (xy1(1,2)+(1/9.5)*hei1) (wid1-(1/length_crop_width_2)*wid1) (hei1-
(1/5)*hei1)]);
%inspect the plate is white or blue
level =graythresh(image_crop_rotate_cut);
imagen=im2bw(image_crop_rotate_cut,level);
value0=0;
value1=0;
for ii=1:(size(imagen,1)/4)
    for jj=1:size(imagen,2)
        if imagen(ii,jj)==1
            value1=value1+1;
        else
            value0=value0+1;
        end
    end
end
if value1 > value0
    typeplate='White Background Plate';
    typeplate_write='White';
    imagen=~im2bw(image_crop_rotate_cut,level);
else
    typeplate='Blue Background Plate';
    typeplate_write='Blue';
    imagen=im2bw(image_crop_rotate_cut,level);
end
%end--inspect the plate is white or blue
end
set(handles.text10,'string',typeplate);
set(handles.text3,'string',angle);
global limit_horizontal
limit=size(imagen);
limit=round((limit(1)*limit(2))/limit_horizontal);% 45; 60 100
imagen = bwareaopen(imagen,limit);
end
end
end
```

```

% tinh chieu dai cua doan thang
len = norm(lines(k).point1 - lines(k).point2);
if( len > max_len)
    max_len = len;
    xy_long = xyk;
end
end
if i>0
    global mark1
    mark1=1;
    break;
else
    if are_process==0
        options_recognition=2;
        loop_option=2; % neu da thuc hien vertical sau do roi thi gan bien de ko lap lai lan
2
    end
end
end
%-----VERTICAL-----
if options_recognition==2 % nhan dang bien so xe theo chieu ngang truoc
    are_process=1; %neu thuc hien truoc thi danh dau lai
BW = edge(G,'sobel',[],'vertical'); %vertical
axes(handles.axes3);
imshow(BW);
global number_lines_vertical
[H,T,R] = hough(BW);
P = houghpeaks(H,number_lines_vertical);

global fillgap_vertical
global minlength_vertical

lines =
houghlines(BW,T,R,P,'FillGap',fillgap_vertical,'MinLength',minlength_vertical);%3
0 - 100
axes(handles.axes2);
imshow(S1); hold on;
i = 0;
max_len = 0;
wid1=0;
hei1=0;
wid2=0;

```

```

hei2=0;
xy1=[0 0;0 0];
xy2=[0 0;0 0];
%-----
if length(lines) == 1 % Luot qua truong hop ko tim duoc bat ky duong thang nao
else
for k = 1:length(lines)
    xyk = [lines(k).point1; lines(k).point2];
    plot(xyk(:,1),xyk(:,2),'LineWidth',2,'Color','green');
    % Plot beginnings and ends of lines
    plot(xyk(1,1),xyk(1,2),'x','LineWidth',2,'Color','yellow');
    plot(xyk(2,1),xyk(2,2),'x','LineWidth',2,'Color','red');
    for j=(k+1):length(lines)
        xyj = [lines(j).point1; lines(j).point2];
        lenk(k) = norm(lines(k).point1 - lines(k).point2);
        lenj(j) = norm(lines(j).point1 - lines(j).point2);
        lenkj(k+j)=norm(xyk(1,1) - xyj(1,1)); % (1,2) la toa do cua Y, (1,1) la toa do
        cua X
        tyle1=(lenkj(k+j)/lenk(k));
        tyle2=(lenkj(k+j)/lenj(j));
        global err_value
        err = norm(xyk(1,2) - xyj(1,2));
        %err1 = norm(xyk(2,2)-xyj(2,2));
        if err < err_value
            %if (err1 < 77)
            if (tyle1 > 1.2 && tyle1 <1.8) %||(tyle1 > 3.5 && tyle1 <5)
                i = i +1 ;
                wid1 = lenkj(k+j);
                hei1 = lenk(k);
                if xyk(1,1)< xyj(1,1)
                    xy1 = xyk;
                else
                    xy1 = xyj;
                end
            else
                if (tyle1 > 1.2 && tyle1 <1.8) %||(tyle1 > 3.5 && tyle1 <5)
                    i = i +1 ;
                    wid1 = lenkj(k+j);
                    hei1 = lenj(j);
                    if xyk(1,1)< xyj(1,1)
                        xy1 = xyk;
                    xy2 = xyj; % to compare angle error
                else

```

```

xy1 = xyj;
xy2 = xyk;
end
end
end
%end
if i>0
image_crop=imcrop(S1,[xy1(1,1) xy1(1,2) wid1 hei1]);
% modify angle error for two rows plate license
if xy1(1,1)==xy1(2,1) % so sanh 2 gia tri hoanh do voi nhau
    angle_im=0;
    angle='0';
    S1_rotate=S1;
    image_crop_rotate=image_crop;
    image_crop_rotate_cut=imcrop(S1_rotate,[(xy1(1,1)+(1/15)*wid1)
(xy1(1,2)+(1/35)*hei1) (wid1-(2/15)*wid1) (hei1-(1/20)*hei1)]);
    %Kiem tra loai bang so nen trang hay nen xanh
    level = graythresh(image_crop_rotate_cut);
    imagen=im2bw(image_crop_rotate_cut,level);
    value0=0;
    value1=0;
%tim cac gia tri bang 1 va 0 de so sanh
for ii=1:(size(imagen,1)/4)
    for jj=1:size(imagen,2)
        if imagen(ii,jj)==1
            value1=value1+1;
        else
            value0=value0+1;
        end
    end
end
if value1 > value0
    typeplate='White Background Plate';
    typeplate_write='White';
    imagen=~im2bw(image_crop_rotate_cut,level);
else
    typeplate='Blue Background Plate';
    typeplate_write='Blue';
    imagen=im2bw(image_crop_rotate_cut,level);
end
%ket thuc viec xac dinh loai bien so xe la nen trang hoac xanh
else

```

```

sin_im=(norm(xy1(1,1) - xy1(2,1)))/(norm(xy1(1,2) - xy1(2,2));% tinh sin
goc lech --> chinh do nghieng anh
%sin_im=(norm(xyk(1,2) - xyj(1,2)))/(norm(xyk(1,1) - xyj(1,1));% tinh sin
goc lech --> chinh do nghieng anh
angle_im=(asin(sin_im)*180)/3.14;% tinh goc de xoay anh
if xy1(1,1)< xy1(2,1)
%if xy2(1,2)< xy1(1,2)
    S1_rotate=imrotate(S1,-angle_im,'bilinear','crop');
    angle=num2str(angle_im);
else
    S1_rotate=imrotate(S1,angle_im,'bilinear','crop');
    angle=strcat('-',num2str(angle_im));
end

%*****successful*****
%chuong trinh con de tim cac thong so moi cho xy1
if abs(angle_im)>4
[xy1, wid1, hei1] = findxy_vertical(S1_rotate, fillgap_vertical,
minlength_vertical, number_lines_vertical);
end
% ket thuc chuong trinh con tim lai cac thong so moi

%*****successful*****
image_crop_rotate=imcrop(S1_rotate,[xy1(1,1) xy1(1,2) wid1
hei1]);%(hei1-(1/20)*hei1)
image_crop_rotate_cut=imcrop(S1_rotate,[(xy1(1,1)+(1/15)*wid1) xy1(1,2)
(wid1-(2/20)*wid1) (hei1-(1/20)*hei1)]);%(xy1(1,2)+(1/35)*hei1)
%inspect the plate is white or blue
level = graythresh(image_crop_rotate_cut);
imagen=im2bw(image_crop_rotate_cut,level);
value0=0;
value1=0;
for ii=1:(size(imagen,1)/4)
    for jj=1:size(imagen,2)
        if imagen(ii,jj)==1
            value1=value1+1;
        else
            value0=value0+1;
        end
    end
end
if value1 > value0
    typeplate='White Background Plate';

```

```

        typeplate_write='White';
        imagen=~im2bw(image_crop_rotate_cut,level);
    else
        typeplate='Blue Background Plate';
        typeplate_write='Blue';
        imagen=im2bw(image_crop_rotate_cut,level);
    end
%end--inspect the plate is white or blue
end
set(handles.text10,'string',typeplate);
set(handles.text3,'string',angle);
global limit_vertical
limit=size(imagen);
limit=round((limit(1)*limit(2))/limit_vertical);%60 100 120
imagen = bwareaopen(imagen,limit);
end
end
end
% tinh chieu dai cua doan thang
len = norm(lines(k).point1 - lines(k).point2);
if ( len > max_len)
    max_len = len;
    xy_long = xyk;
end
end
if i>0
    mark1=2;
    break;
else
    options_recognition=1;
end
end
%-----
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','blue');
if i > 0
axes(handles.axes4);
imshow(image_crop_rotate);
axes(handles.axes5);
imshow(imagen);
end

```

```
%-----TACH VA NHAN DANG KY TU-----
function Recognize_Callback(hObject, eventdata, handles)
global t1
t1=clock;
global imagen l bien_dem bien_dem1 word
global check_bien_dem1
global typeplate_write
global Export_excel
bien_dem =0;
bien_dem1=0;
check_bien_dem1=0;
l=0;
word=[ ];
a=[ ];
re=imagen;
% Load templates
load bsx
global bsx
% Compute the number of letters in template file
num_letras=size(bsx,2);
while 1
    %Fcn 'lines' separate lines in text
    [fl re]=lines(re);
    imgn=fl;
    %Uncomment line below to see lines one by one
    %imshow(fl);pause(0.5)
    %
    % Label and count connected components
    [L Ne] = bwlabel(imgn); % danh dau tung mang (ky tu) diem anh co gia tri bang
    1 theo so thu tu cac mang(ky tu) tu trai qua phai, Ne: number ky tu
    % L la mot anh moi trong do cac diem anh co gia tri 0 va 1 duoc thay
    % the bang cac diem anh co gia tri tu 1 - n dua vao ky thuat danh dau
    % cua bwlabel
    % neu so luong moi line lon hon 6 ky tu thi cho chia line tiep tuc
    for n=1:Ne
        [r,c] = find(L==n); % tim theo ky thuat danh dau theo mang cua label
        % Extract letter
        n1=imgn(min(r):max(r),min(c):max(c)); % n1 la 1 ky tu duoc tach ra
        % check and reject the boundary of plate
        %neu chieu rong cua ky tu nhan dang lon hon 1/2 cua chieu rong cua
        %bien so thi ta bo qua ko nhan dang
        bsx_wide=size(imagen,2);
        letter_wide=size(n1,2);
```

```

bsx_height=size(imagen,1);
letter_height=size(n1,1);
global mark1
filter_value=0;
if mark1==1
    if letter_wide > (bsx_wide/4)||letter_height==bsx_height
        filter_value=1;
    end
else
    if letter_wide > (bsx_wide/4)||letter_height > (bsx_height/2);
        filter_value=1;
    end
end
if filter_value==1
    %nothing
else
    % Resize letter (same size of template)
    img_r=imresize(n1,[42 24]);
    % Call fcn to convert image to text
    [letter]=read_letter(img_r,num_letras);
    % Letter concatenation
    % chen code 2
    global Soluong_chuso_dangky
    global letter letter1
    if bien_dem==0
        word=[word letter];
        if check_bien_dem1==1
            word1=[word1 letter1];
        end
        % Them dau"." vao ky tu bien so xe
        % Giai thuat: dem tong so so dang ky phia sau ky tu CHU CAI. Neu
        % tong so la 4 thi giu nguyen ko thay doi. Nguoc lai neu tong so
        % la 5 thi chung ta thay the "Word co them dau"." thanh Word
        % cuoi cung.
        global word_dot
        if Soluong_chuso_dangky==3
            word_dot=strcat(word,'.');
        end
        if Soluong_chuso_dangky>3
            word_dot=[word_dot letter];
        end
        if Soluong_chuso_dangky==5
            word=word_dot;

```

```

    end
    % Ket thuc viec them dau "."
else
if bien_dem1==1
bien_dem=0;
word1=[word letter1];
word=[word letter];
check_bien_dem1=1;
elseif bien_dem1==2
bien_dem=0;
word=[word1 letter];
else
bien_dem=0;
word=[word letter];
end
end

a=[a img_r];
end
end
if isempty(re) %See variable 're' in Fcn 'Lines'
axes(handles.axes6);
imshow(a);
set(handles.text6,'string',word);
break
end
end
global biensoxe1 biensoxe2 biensoxe3 O_cardoor1 O_cardoor2 O_cardoor3
global enable_OPc
if enable_OPc==1
while(1)
M_cardoor1=struct2cell(read(O_cardoor1));
M_cardoor1=M_cardoor1{2};

M_cardoor2=struct2cell(read(O_cardoor2));
M_cardoor2=M_cardoor2{2};

M_cardoor3=struct2cell(read(O_cardoor3));
M_cardoor3=M_cardoor3{2};
global mark_end check_M_cardoor1 check_M_cardoor2 check_M_cardoor3
if M_cardoor1==0
check_M_cardoor1=0;
end

```

```
if M_cardoor2==0
    check_M_cardoor2=0;
end

if M_cardoor3==0
    check_M_cardoor3=0;
end

if check_M_cardoor1==0
if M_cardoor1==1
    writeasync(biensoxe1,word);
    check_M_cardoor1=1;
    M_cardoor1=0;
    mark_end=1;
end
end

if check_M_cardoor2==0
if M_cardoor2==1
    writeasync(biensoxe2,word);
    check_M_cardoor2=1;
    M_cardoor2=0;
    mark_end=1;
end
end

if check_M_cardoor3==0
if M_cardoor3==1
    writeasync(biensoxe3,word);
    check_M_cardoor3=1;
    M_cardoor3=0;
    mark_end=1;
end
end

if mark_end==1
    mark_end=0;
    break;
end
end
if Export_excel==1
```

```

delete bangsoxe.xls;
xlswrites('bangsoxe.xls',word,typeplate_write,'A1','A2');
end
% Tinh toan thoi gian xu ly
t2=clock;
t=etime(t2,t1);
t=strcat(num2str(t), ' second');
set(handles.text13,'string',t);
% End tinh toan thoi gian xu ly ket qua nhan dang
% Qua Tuyet Voi
global check_camera
if check_camera ==1
    From_Camera_Callback(hObject, eventdata, handles);
end
% Qua tuyet voi
function [fl re]=lines(im_texto)
im_texto=clip(im_texto); % da clip lai lan thu 2 cai line con lai
num_filas=size(im_texto,1);
for s=1:num_filas
    if sum(im_texto(s,:))==0
        nm=im_texto(1:s-1, :); % First line matrix
        rm=im_texto(s:end, :);% Remain line matrix
        %fl = clip(nm);
        fl = nm ;
        %re=clip(rm);
        re= rm ;
        break
    else
        fl=im_texto;%Only one line.
        re=[ ];
    end
end
function img_out=clip(img_in)
[f c]=find(img_in);
img_out=img_in(min(f):max(f),min(c):max(c));%Crops image - cat xen cac dong
va cot chi chua toan gia tri 0
function [letter]=read_letter(imagn,num_letras)
global letter letter1
letter=[];
letter1=[];
global kytu_so
global bsx bien_dem bien_dem1
global l

```



```

function uibuttongroup_options_recognition_SelectionChangeFcn(hObject,
eventdata, handles)
global options_recognition mark;
switch get(eventdata.NewValue,'Tag') % Get Tag of selected object.
    case 'one_row'
        options_recognition=1;
        mark=1;
    case 'two_rows'
        options_recognition=2;
        mark=2;
    otherwise
        % Code for when there is no match.
end
% --- Executes on button press in close.
function close_Callback(hObject, eventdata, handles)
global close_value
close_value=1;
choice = questdlg('Are you sure exit VLP ?',...
    'Close VLP', ...
    'Yes','No','Yes');
% Handle response
switch choice
    case 'Yes'
        close(gcf)
    case 'No'
end

% --- Executes on button press in Excel_export.
function Excel_export_Callback(hObject, eventdata, handles)
global Export_excel
% Hint: get(hObject,'Value') returns toggle state of Excel_export
if (get(hObject,'Value') == get(hObject,'Max'))
    Export_excel = 1;
else
    Export_excel = 0;
end

function OPC_Connect_Callback(hObject, eventdata, handles)
% Hint: get(hObject,'Value') returns toggle state of OPC_Connect
global enable_OPc
if (get(hObject,'Value') == get(hObject,'Max'))
    % Radio button is selected, take appropriate action
    enable_OPc=1;

```

```

else
    % Radio button is not selected, take appropriate action
    enable_OPc=0;
end

function Sampling_Callback(hObject, eventdata, handles)
sampling;

```

## CHƯƠNG TRÌNH SO SÁNH KÝ TỰ MÃU

```

function [letter, kytu_so] = compare_letter(vd)
load data
letter=[];
pass=0;
na=size(data{1,1},2);nb=size(data{1,2},2);nc=size(data{1,3},2);nd=size(data{1,4},2);
ne=size(data{1,5},2);nf=size(data{1,6},2);ng=size(data{1,7},2);nh=size(data{1,8},2);
ni=size(data{1,9},2);nj=size(data{1,10},2);nk=size(data{1,11},2);nl=size(data{1,12},2);
nm=size(data{1,13},2);nn=size(data{1,14},2);no=size(data{1,15},2);
np=size(data{1,16},2);nq=size(data{1,17},2);nr=size(data{1,18},2);ns=size(data{1,19},2);
nt=size(data{1,20},2);nu=size(data{1,21},2);nv=size(data{1,22},2);
nw=size(data{1,23},2);nx=size(data{1,24},2);ny=size(data{1,25},2);nz=size(data{1,26},2);
n1=size(data{1,27},2);n2=size(data{1,28},2);n3=size(data{1,29},2);n4=size(data{1,30},2);
n5=size(data{1,31},2);n6=size(data{1,32},2);n7=size(data{1,33},2);
n8=size(data{1,34},2);n9=size(data{1,35},2);n0=size(data{1,36},2);
if pass==0
for i=1:na
    if vd==data{1,1}(1,i)
        letter='A';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nb
    if vd==data{1,2}(1,i)
        letter='B';
        kytu_so = 0;
        pass=1;

```

```
        break;
    end
end
end

if pass==0
for i=1:nc
    if vd==data{1,3}(1,i)
        letter='C';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nd
    if vd==data{1,4}(1,i)
        letter='D';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:ne
    if vd==data{1,5}(1,i)
        letter='E';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nf
    if vd==data{1,6}(1,i)
        letter='F';
        kytu_so = 0;
```

```
pass=1;
break;
end
end
end

if pass==0
for i=1:ng
    if vd==data{1,7}(1,i)
        letter='G';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nh
    if vd==data{1,8}(1,i)
        letter='H';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:ni
    if vd==data{1,9}(1,i)
        letter='T';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nj
    if vd==data{1,10}(1,i)
        letter='J';
```

```
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nk
if vd==data{1,11}(1,i)
letter='K';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nl
if vd==data{1,12}(1,i)
letter='L';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nm
if vd==data{1,13}(1,i)
letter='M';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nn
if vd==data{1,14}(1,i)
```

```
letter='N';
kytu_so = 0;
pass=1;
break;
end
end
end

if pass==0
for i=1:no
    if vd==data{1,15}(1,i)
        letter='O';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:np
    if vd==data{1,16}(1,i)
        letter='P';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nq
    if vd==data{1,17}(1,i)
        letter='Q';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nr
```

```
if vd==data{1,18}(1,i)
letter='R';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:ns
if vd==data{1,19}(1,i)
letter='S';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nt
if vd==data{1,20}(1,i)
letter='T';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
for i=1:nu
if vd==data{1,21}(1,i)
letter='U';
kytu_so = 0;
pass=1;
break;
end
end

if pass==0
```

```
for i=1:nv
    if vd==data{1,22}(1,i)
        letter='V';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nw
    if vd==data{1,23}(1,i)
        letter='W';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:nx
    if vd==data{1,24}(1,i)
        letter='X';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:ny
    if vd==data{1,25}(1,i)
        letter='Y';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end
```

```
if pass==0
for i=1:nz
    if vd==data{1,26}(1,i)
        letter='Z';
        kytu_so = 0;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n1
    if vd==data{1,27}(1,i)
        letter='1';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n2
    if vd==data{1,28}(1,i)
        letter='2';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n3
    if vd==data{1,29}(1,i)
        letter='3';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end
```

```
if pass==0
for i=1:n4
    if vd==data{1,30}(1,i)
        letter='4';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n5
    if vd==data{1,31}(1,i)
        letter='5';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n6
    if vd==data{1,32}(1,i)
        letter='6';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end

if pass==0
for i=1:n7
    if vd==data{1,33}(1,i)
        letter='7';
        kytu_so = 1;
        pass=1;
        break;
    end
end
```

end

```
if pass==0
for i=1:n8
    if vd==data{1,34}(1,i)
        letter='8';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end
```

```
if pass==0
for i=1:n9
    if vd==data{1,35}(1,i)
        letter='9';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end
```

```
if pass==0
for i=1:n0
    if vd==data{1,36}(1,i)
        letter='0';
        kytu_so = 1;
        pass=1;
        break;
    end
end
end
```

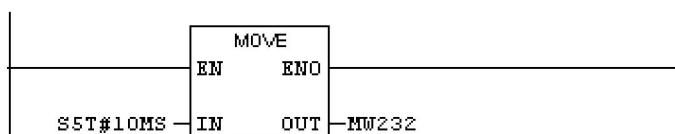
## CHƯƠNG TRÌNH PLC S7 – 300

OB1 : Title:

CHUONG TRINH DIEU KHIEN NHA GIU XE TU DONG BANG GIAO DIEN WINCC

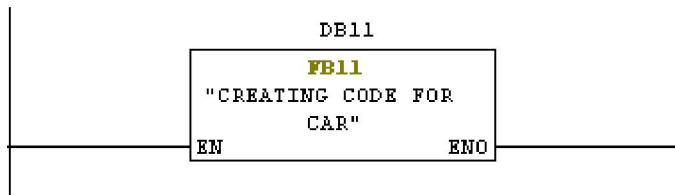
Network 1 : Title:

GAN CHU KY CUA XUNG XUAT RA DIEU KHIEN DONG CO BUOC



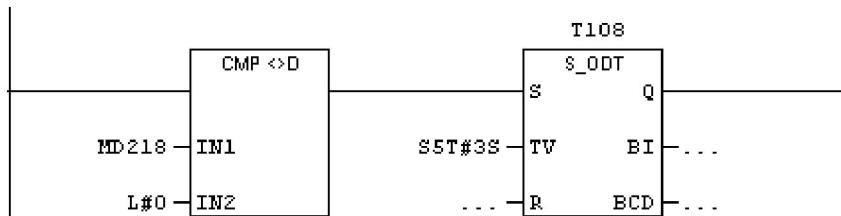
Network 2 : Title:

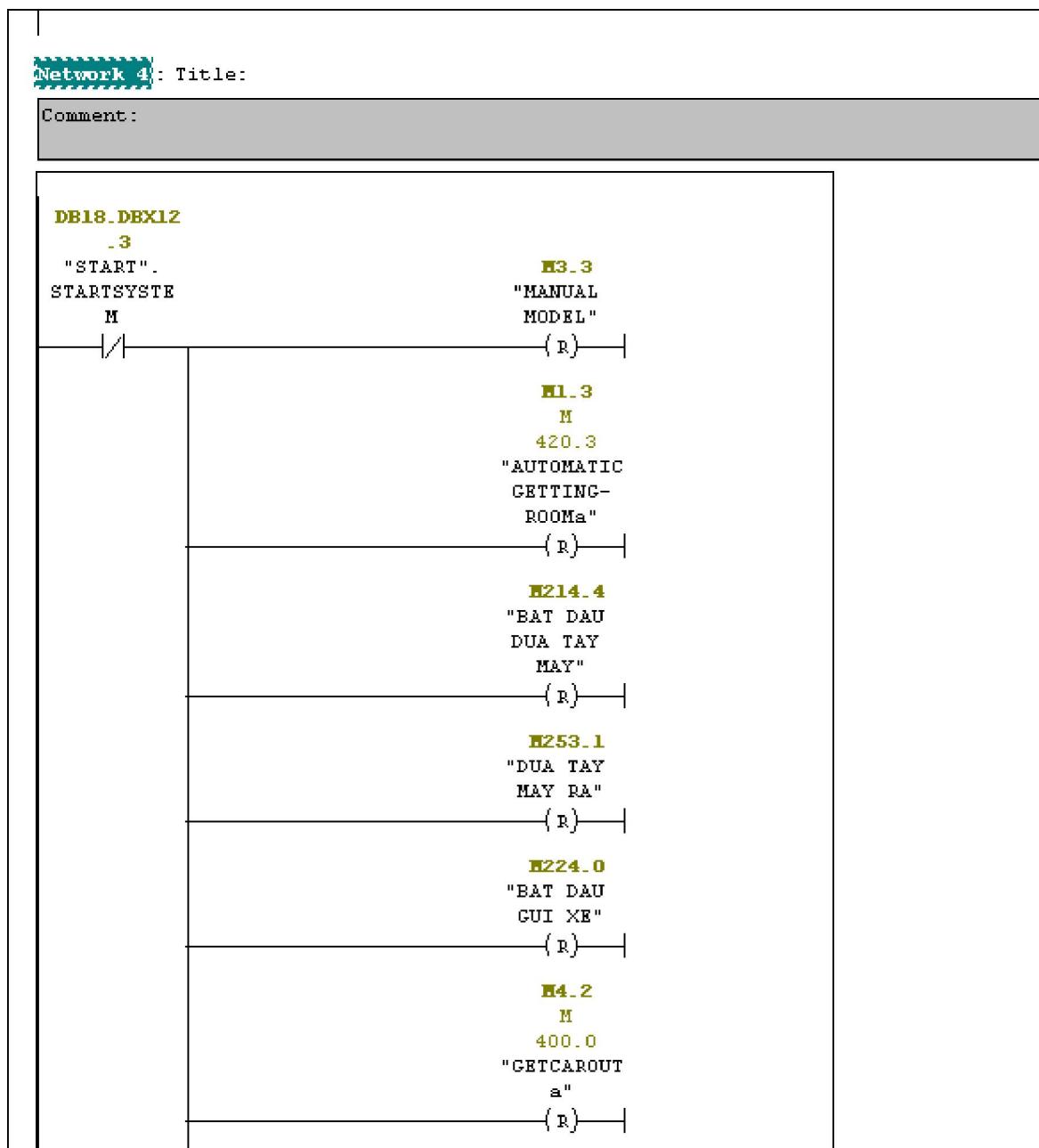
CHUONG TRINH CON TAO MA CHO XE TRUOC KHI GUI XE



Network 3 : Title:

Comment:





		<b>H6.2</b> H 509.3 "KY HIEU LAY XE RAa" ( R )
		<b>H5.6</b> "LAYXE" ( R )
		<b>H1.6</b> H 500.1 "BIT LAY XEA" ( R )
		<b>H252.2</b> H 507.4 "fuyuja" ( R )
		<b>H250.0</b> H 560.0 "PASSa" ( R )
		<b>H6.4</b> H 560.1 "MANUAL TAKING CARa" ( R )
		<b>H1.0</b> "XOAY CABIN TOI P CHON" ( R )

		<b>H0_0</b> "NANG CABIN LEN" ( R )
		<b>H8_0</b> "XOAY VE P. O" ( R )
		<b>H8_2</b> "quayvevit riOnhanxe" ( R )
		<b>H2_5</b> H 560.5 "DA CHON PHONGa" ( R )
		<b>H250_7</b> H 560.6 "TOIUUDELA YXERA" ( R )
		<b>H2_0</b> "SYSTEMST" ( R )
		<b>H250_7</b> H 560.6 "TOIUUDELA YXERA" ( R )
		<b>H0_5</b> H 560.3 "ANH HIEN CHON NHAN"

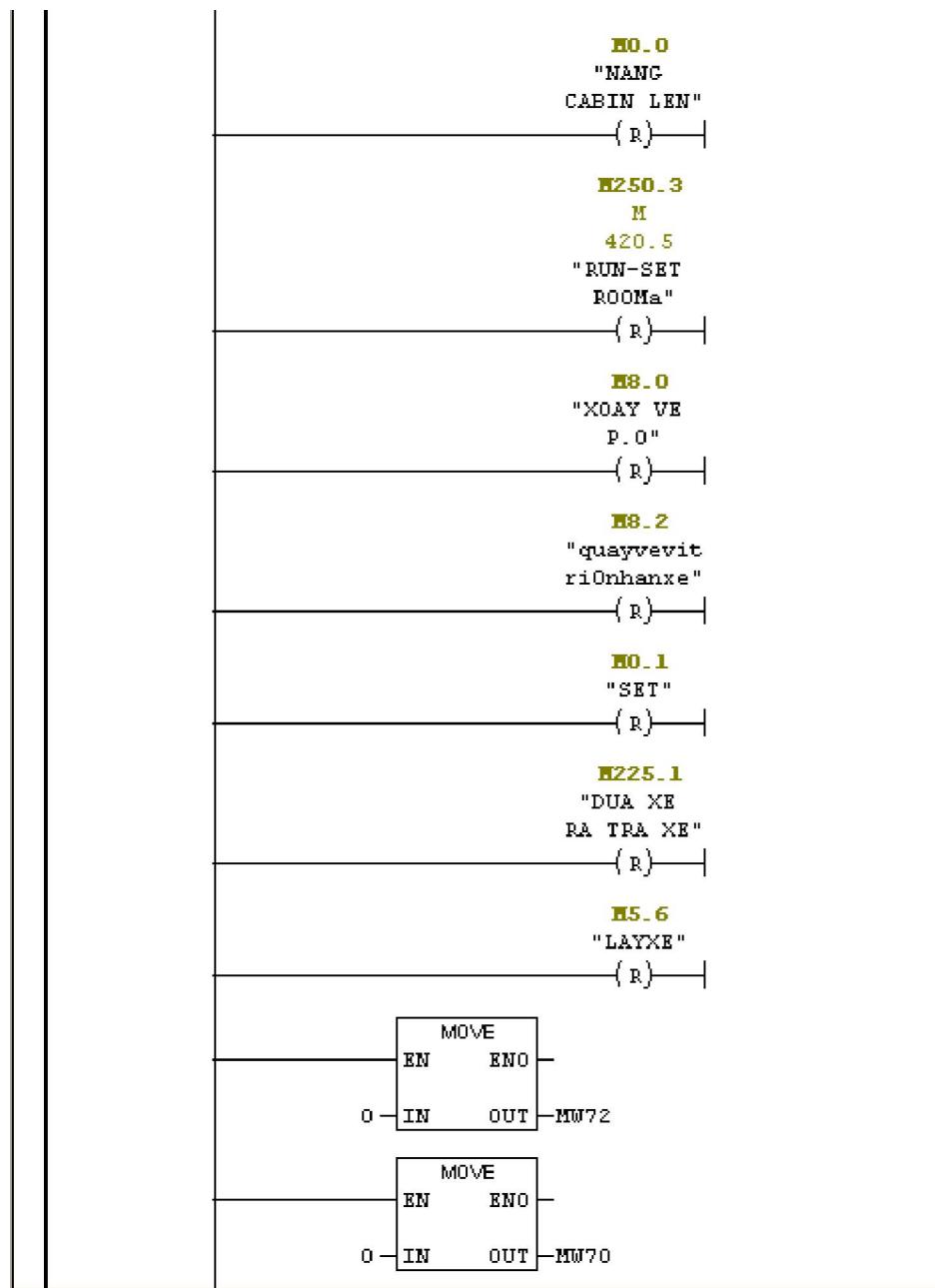
		<b>H0_5</b> M 560.3 "ANH HIEN CHON NHAN PHONG" ( R )
		<b>H2_2</b> M 560.7 "BUSY RUNNING MANUAL MODE" ( R )
		<b>H2_1</b> M 561.0 "BUSY RUNNING AUTO- MODEa" ( R )
		<b>H1_1</b> M 420.1 "AUTOMATIC -CONTROL- MODELa" ( R )
		<b>H3_3</b> "MANUAL MODEL" ( R )
		<b>H250_5</b> M 500.2 "running-- -not-

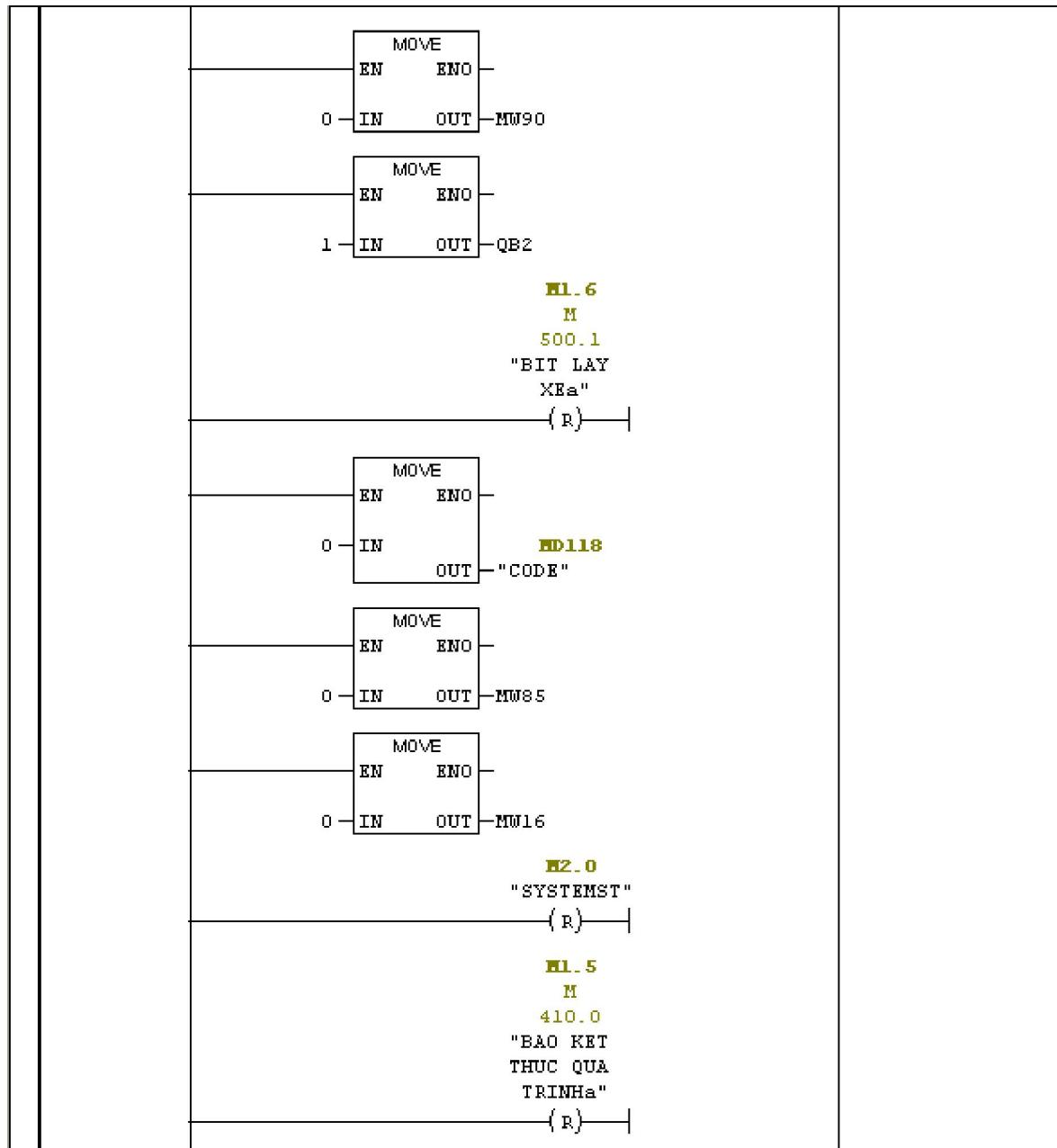
M250.5  
M  
500.2  
"running--  
-not-  
gettingcar  
"  
( R )  
  
C10  
( R )  
  
C11  
( R )  
  
C1  
( R )  
  
C12  
( R )  
  
C103  
( R )  
  
C87  
( R )

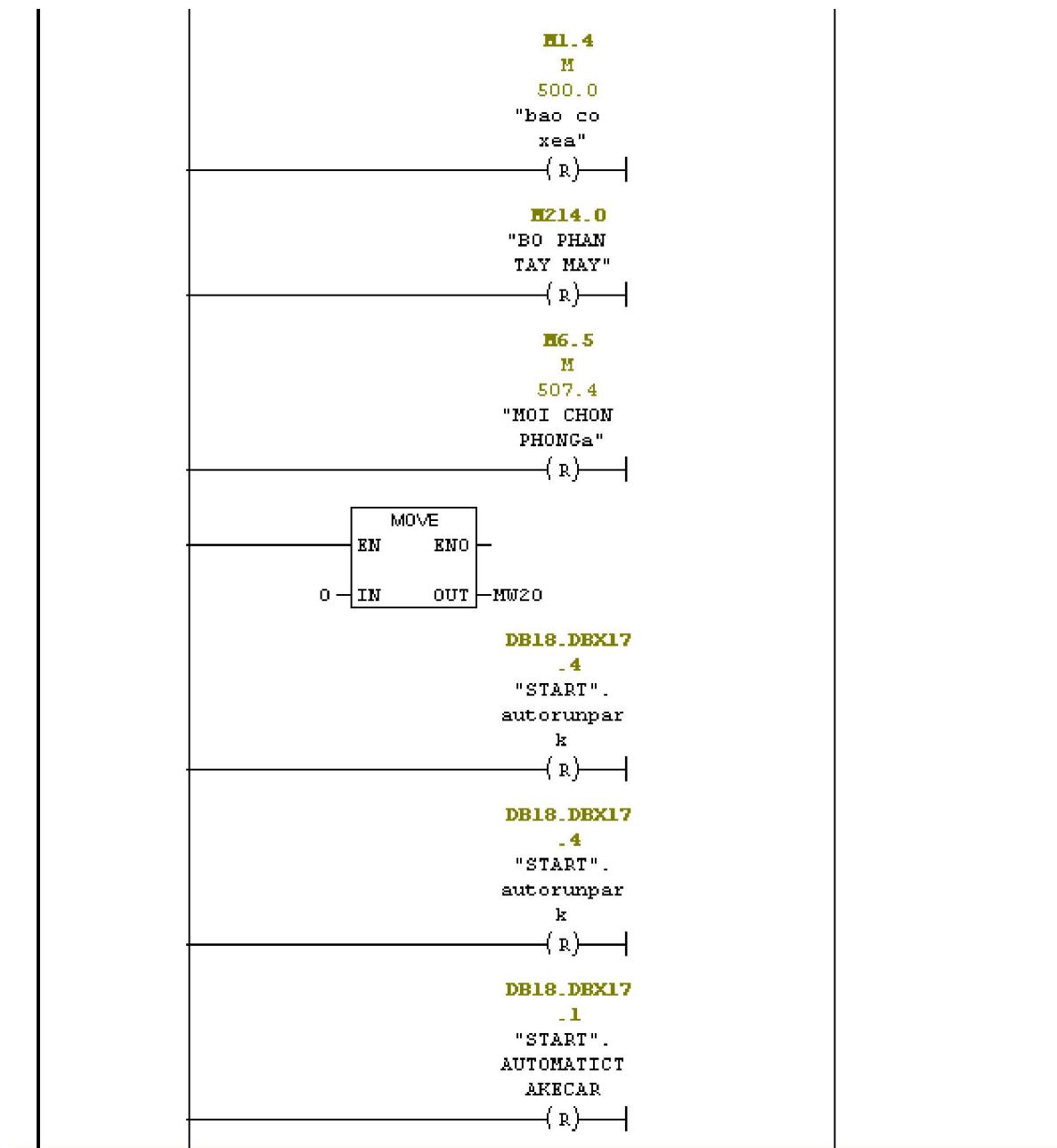
**B6.2**  
**M**  
**509.3**  
"KY HIEU  
LAY XE  
RAs"  
\_\_\_\_\_  
( R )

**H1.0**  
"XOAY  
CABIN TOI  
P CHON"  
( R )

HO. O  
"NANG  
CABIN LEN"

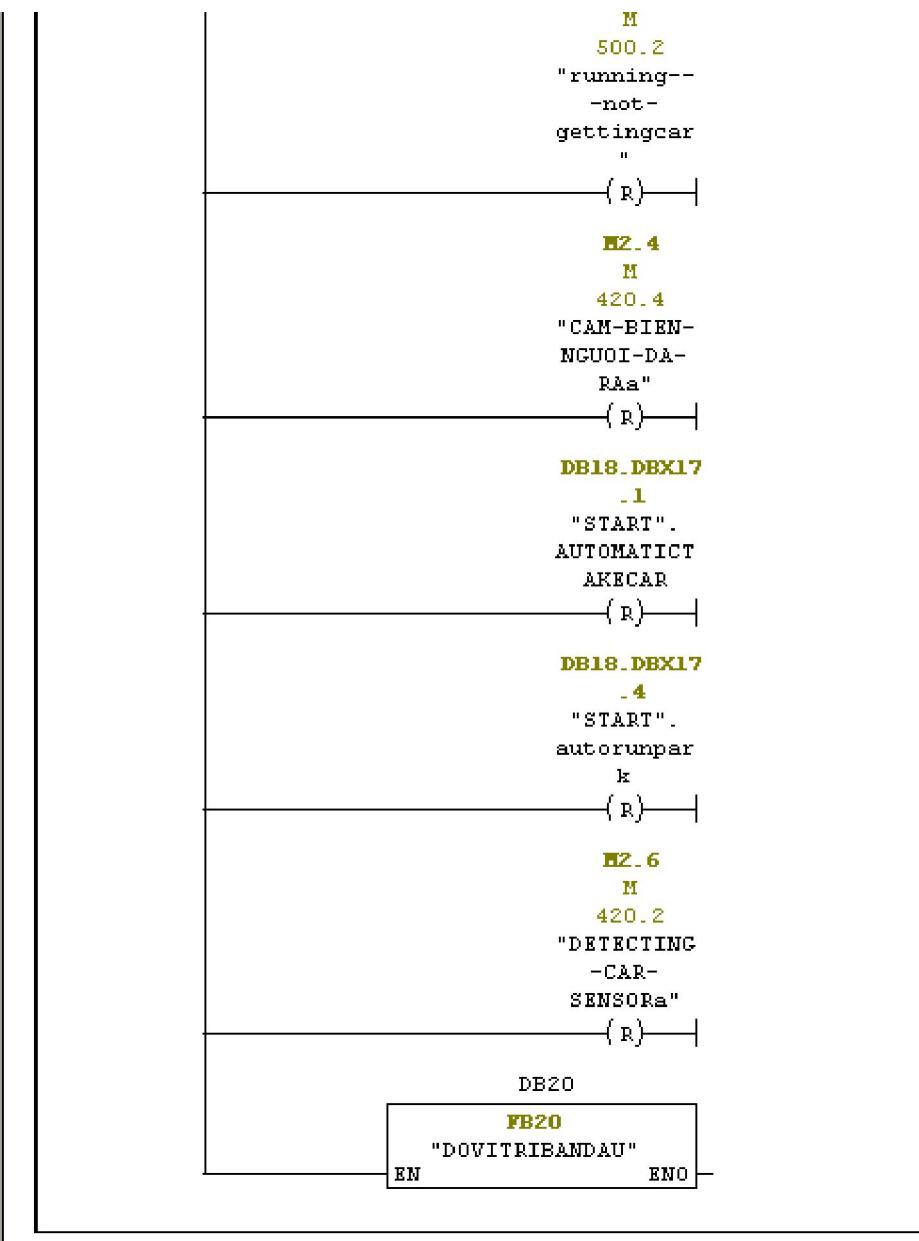






	<b>DB18 . DBX17</b> _1 "START". AUTOMATIC AKECAR ( R )
	<b>M225 . 3</b> "HOAN THANH TRA XE" ( R )
	<b>M3 . 6</b> "Running" ( R )
	<b>M2 . 1</b> M 561.0 "BUSY RUNNING AUTO- MODEa" ( R )
	<b>DB18 . DBX12</b> _0 "START". TT26 ( R )
	<b>M251 . 4</b> M 507.3 "WAITa" ( R )
	<b>M2 . 3</b> M 507.0 "CABIN DUNG 0

		<b>H2.3</b> M 507.0 "CAEIN DUNG 0 PHONG DENa" ( R )
		<b>H250.5</b> M 500.2 "running-- -not- gettingcar " ( R )
		<b>H2.4</b> M 420.4 "CAM-BIEN- NGUOI-DA- RAa" ( R )
		<b>DB18_DBX17</b> <b>1</b> "START". AUTOMATIC AKECAR ( R )
		<b>DB18_DBX17</b> <b>4</b> "START". autorunpar k ( R )
		<b>H2.6</b> M 420.2 "DETECTING



Network 5 : Title:

Comment:



Network 6 : Title:

Comment:



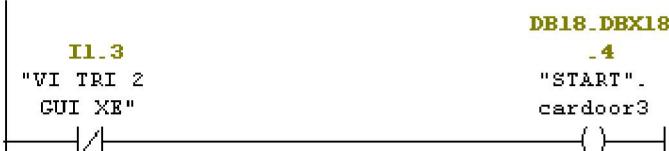
Network 7 : Title:

Comment:



**Network 8 : Title:**

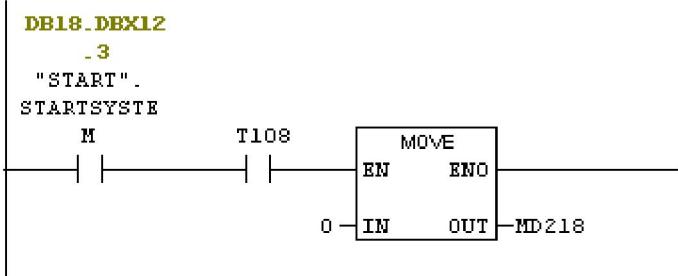
Comment:

**Network 9 : Title:**

Comment:

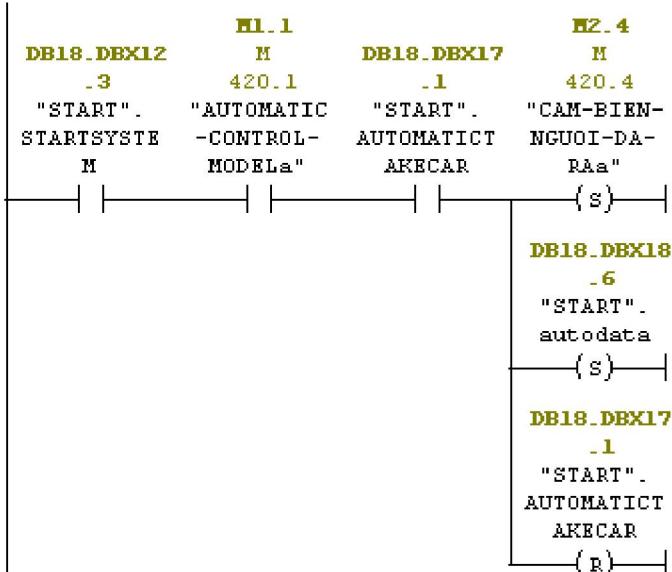
**Network 10 : Title:**

Comment:



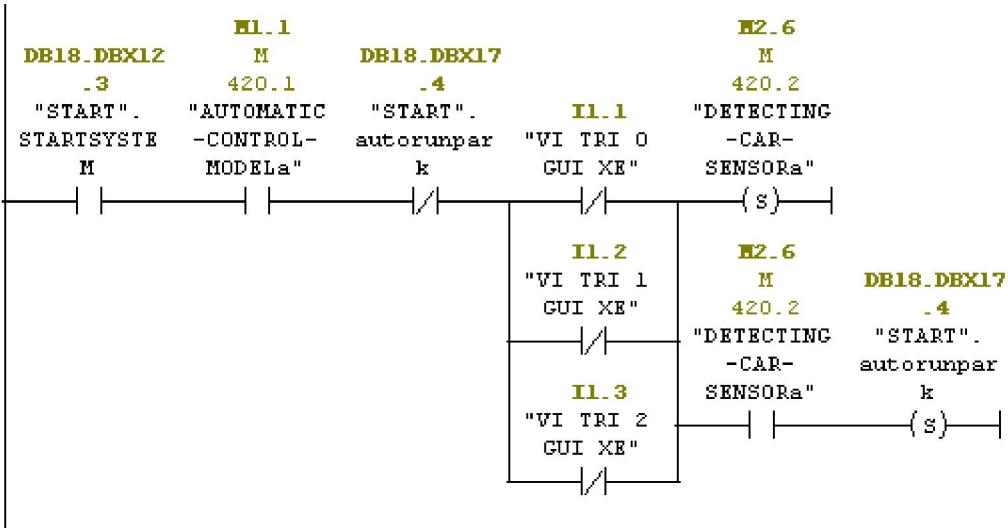
**Network 11:** M 420.4

Comment:



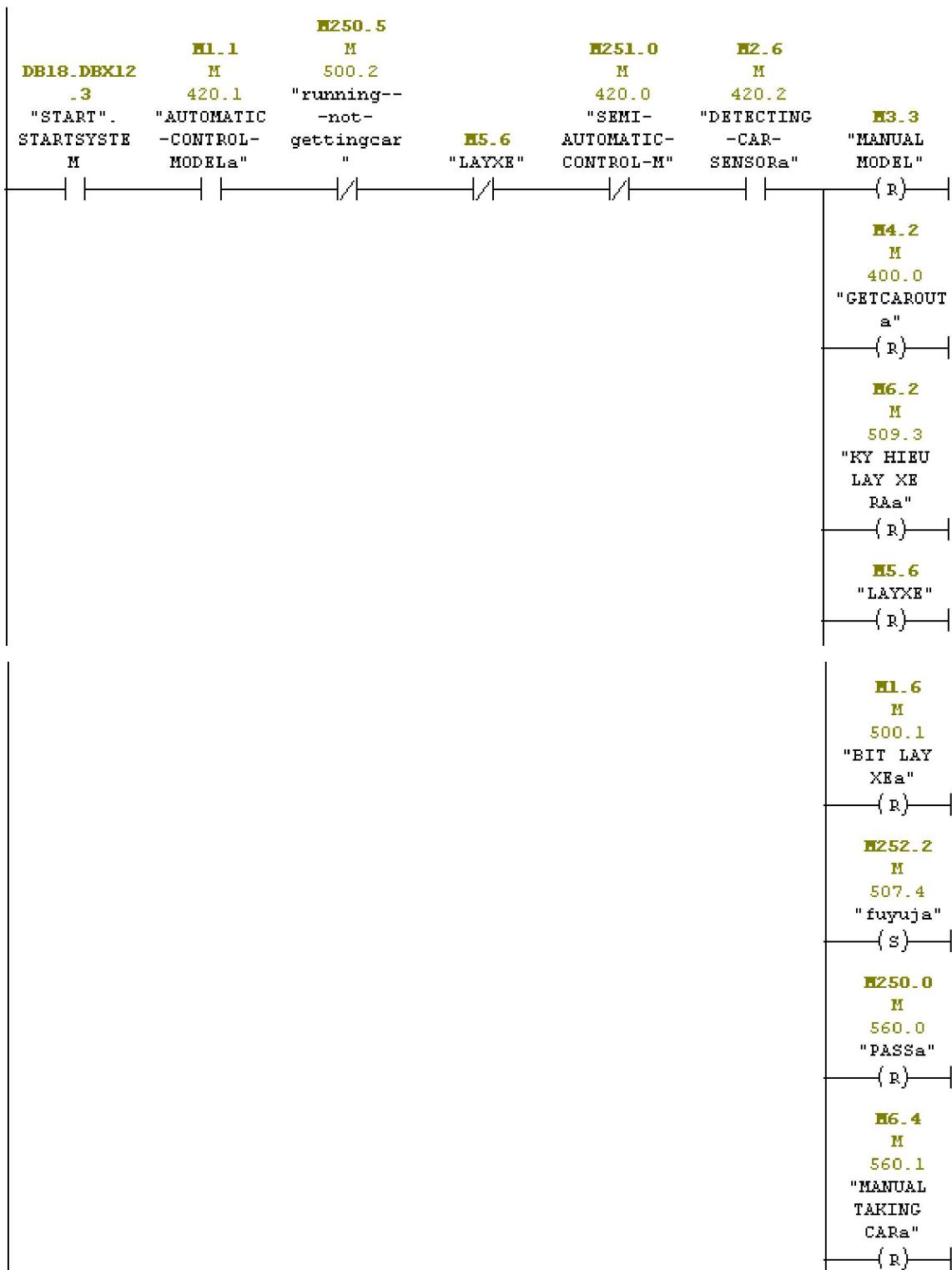
**Network 12 :** M 420.2

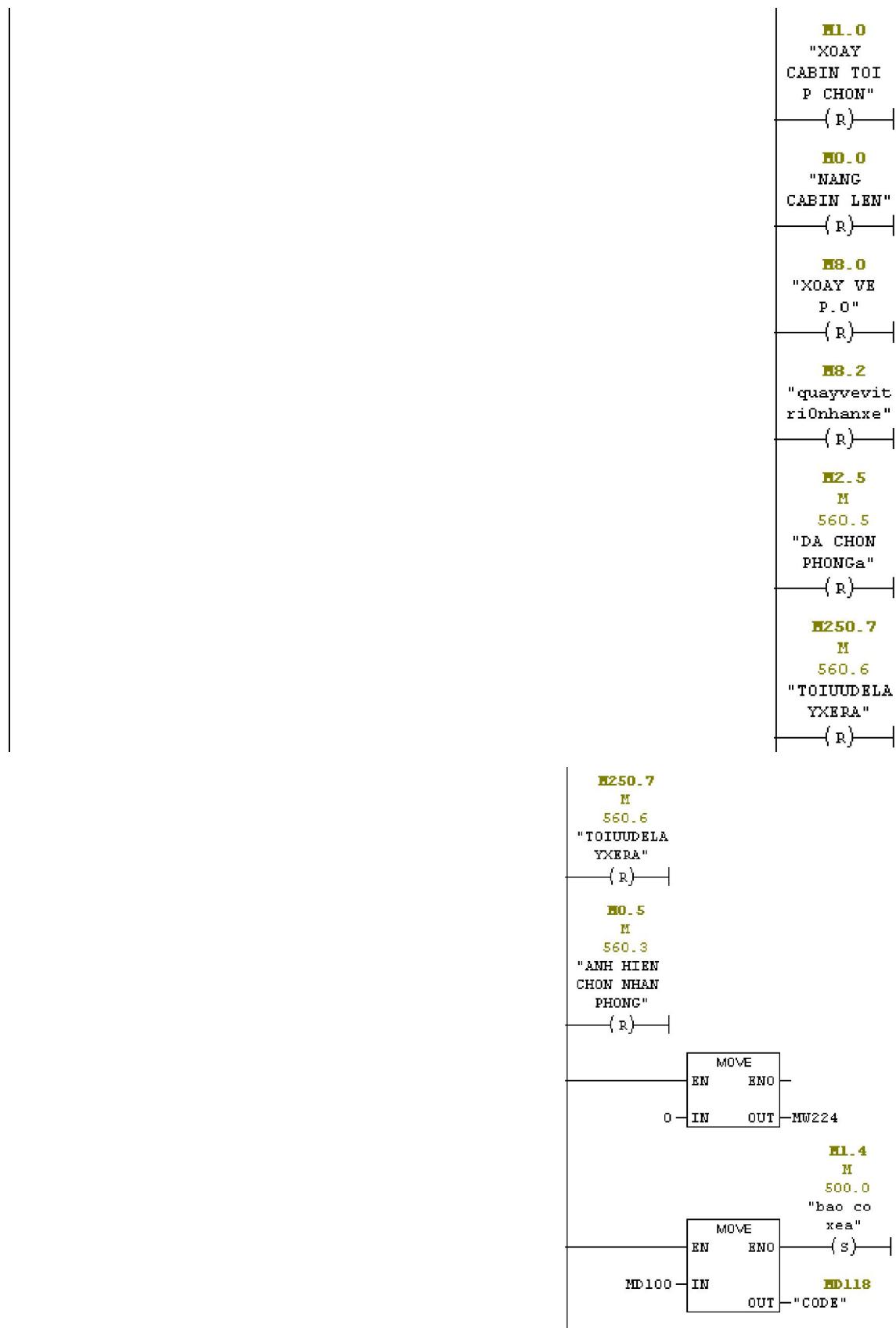
Comment:

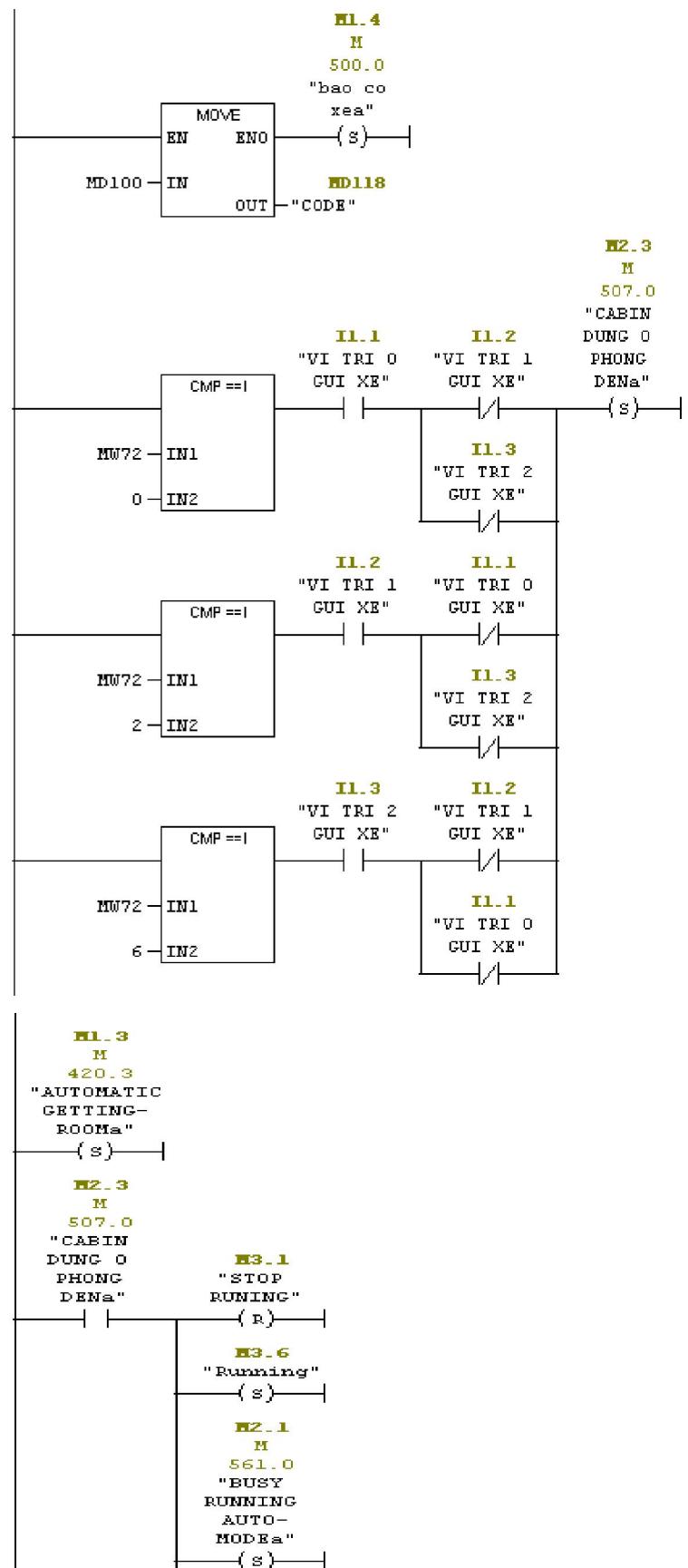


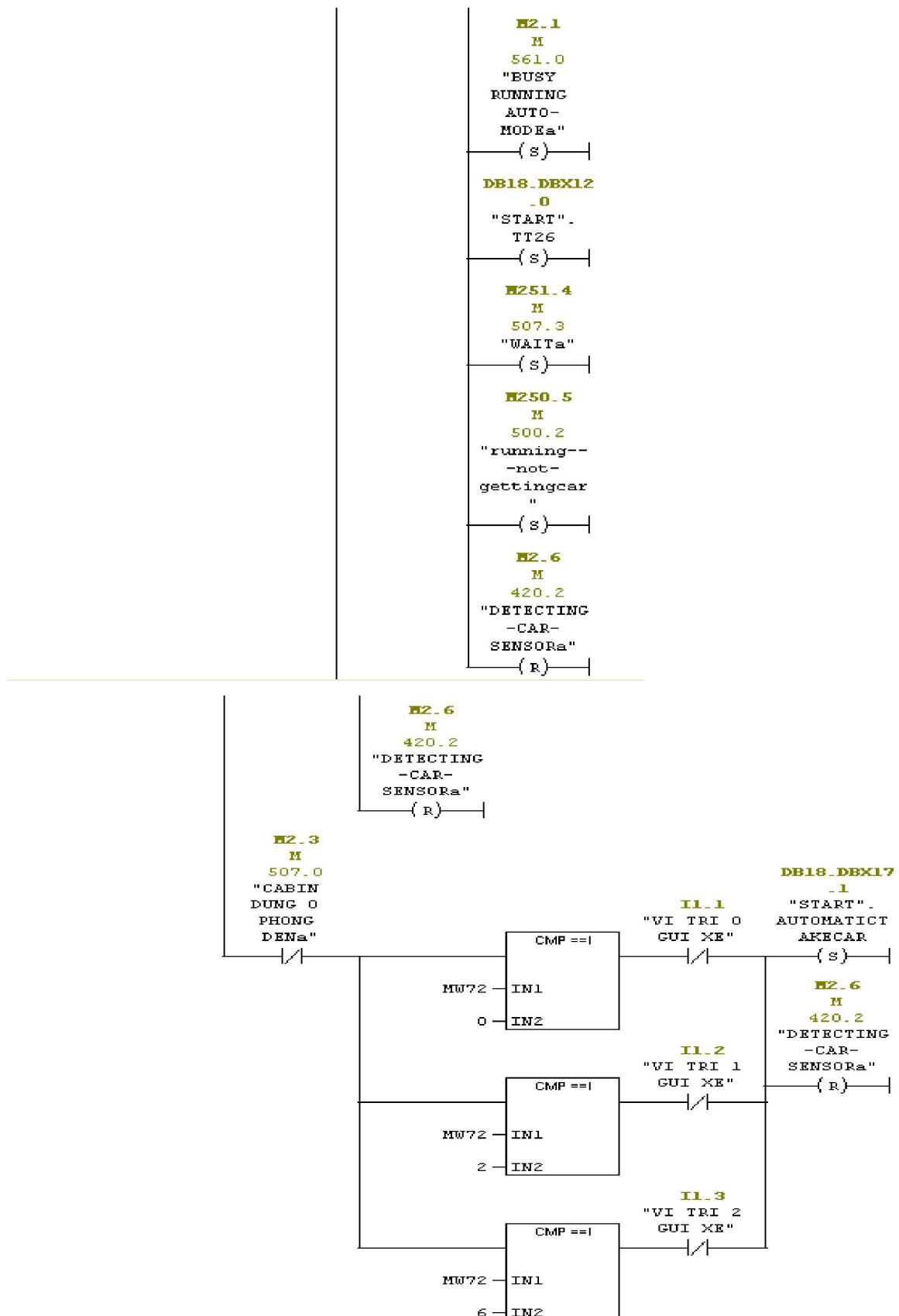
Network 13 : M 420.3

CHON CHE DO GUI XE TU DONG, KIEM TRA CABIN CO DANG O TAI ZERO (PHONG GIAO TRA XE) KHONG? NEU KHONG THI RUN --> DE THUC HIEN DUA CABIN VE VI TRI ZORO TRUOC KHI NHAN XE VAO.



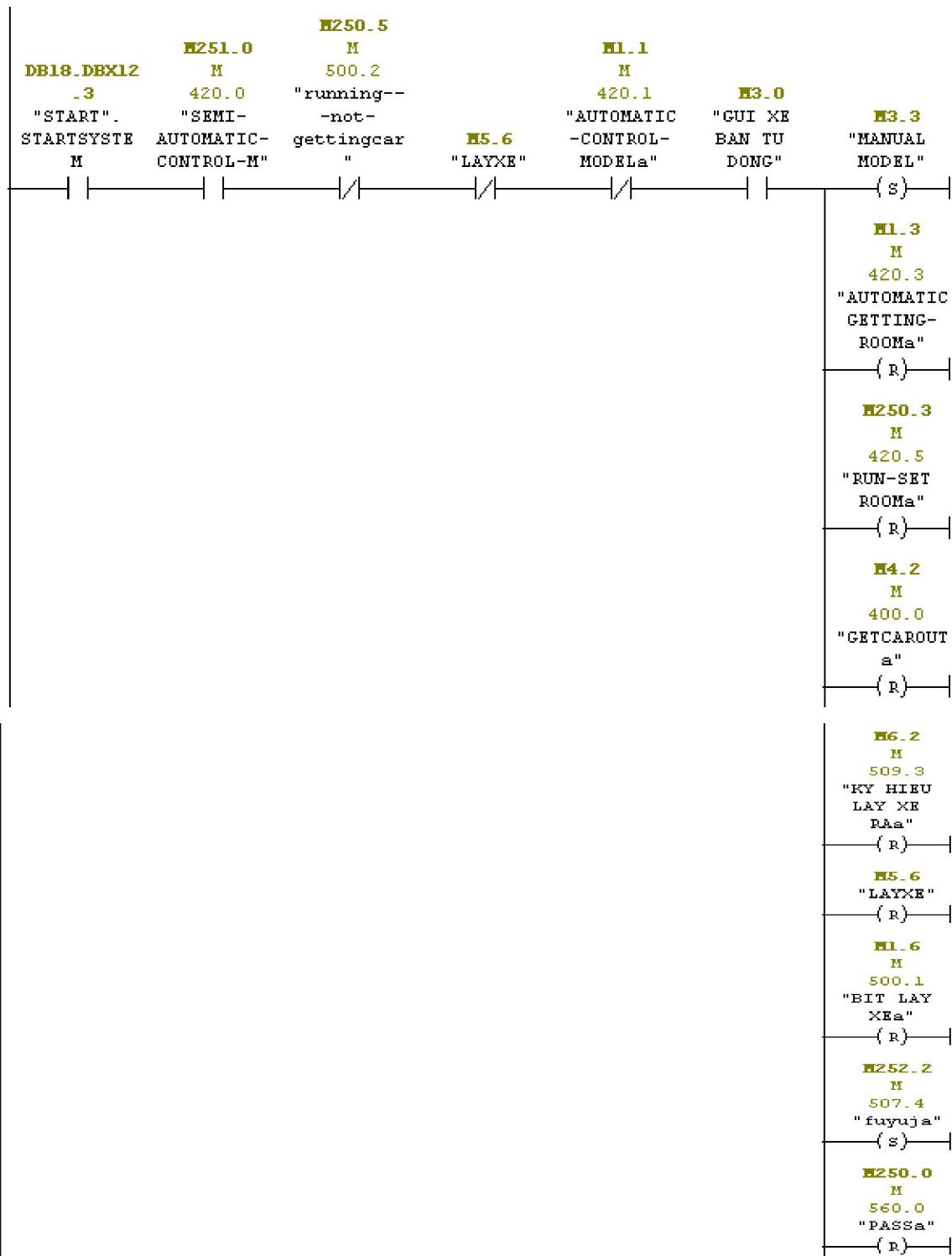


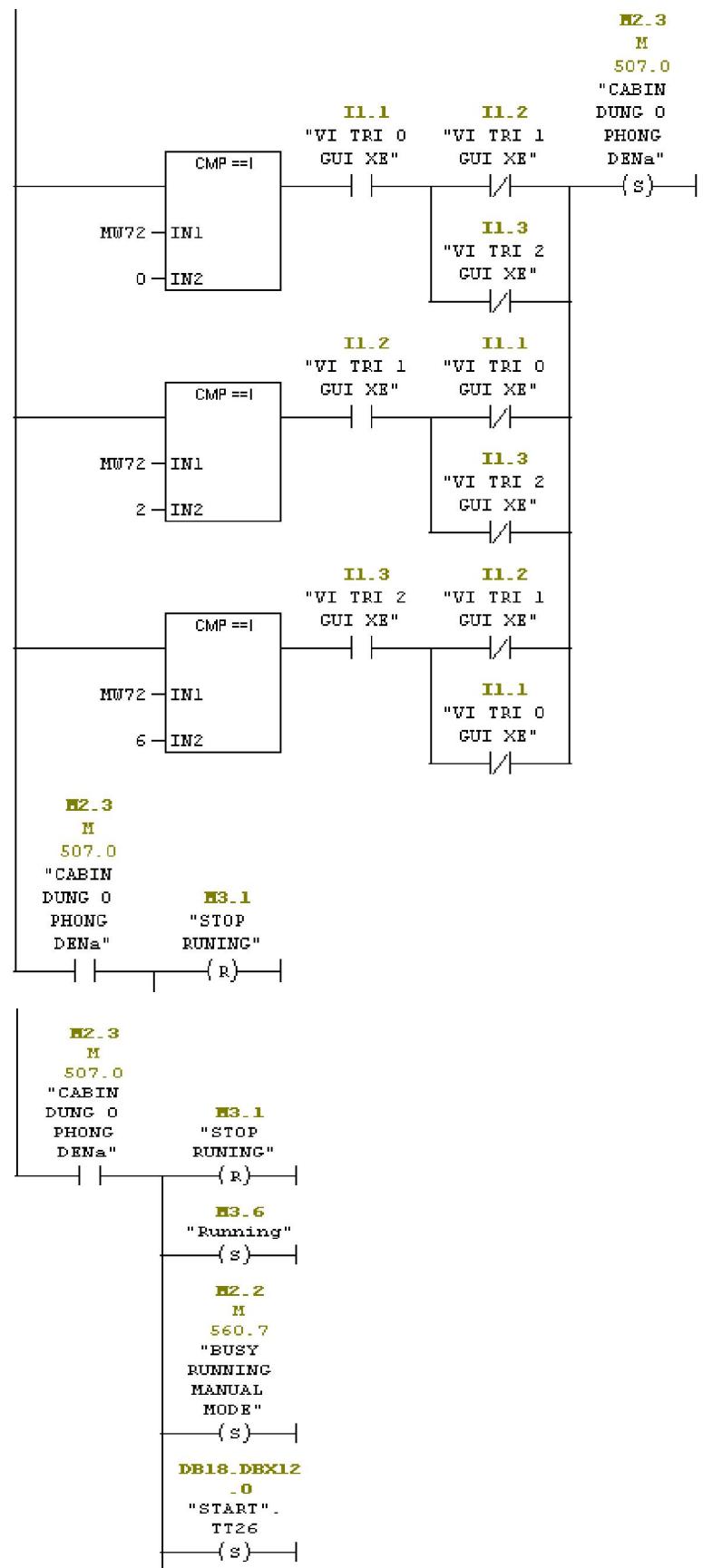




**Network 14 : Title:**

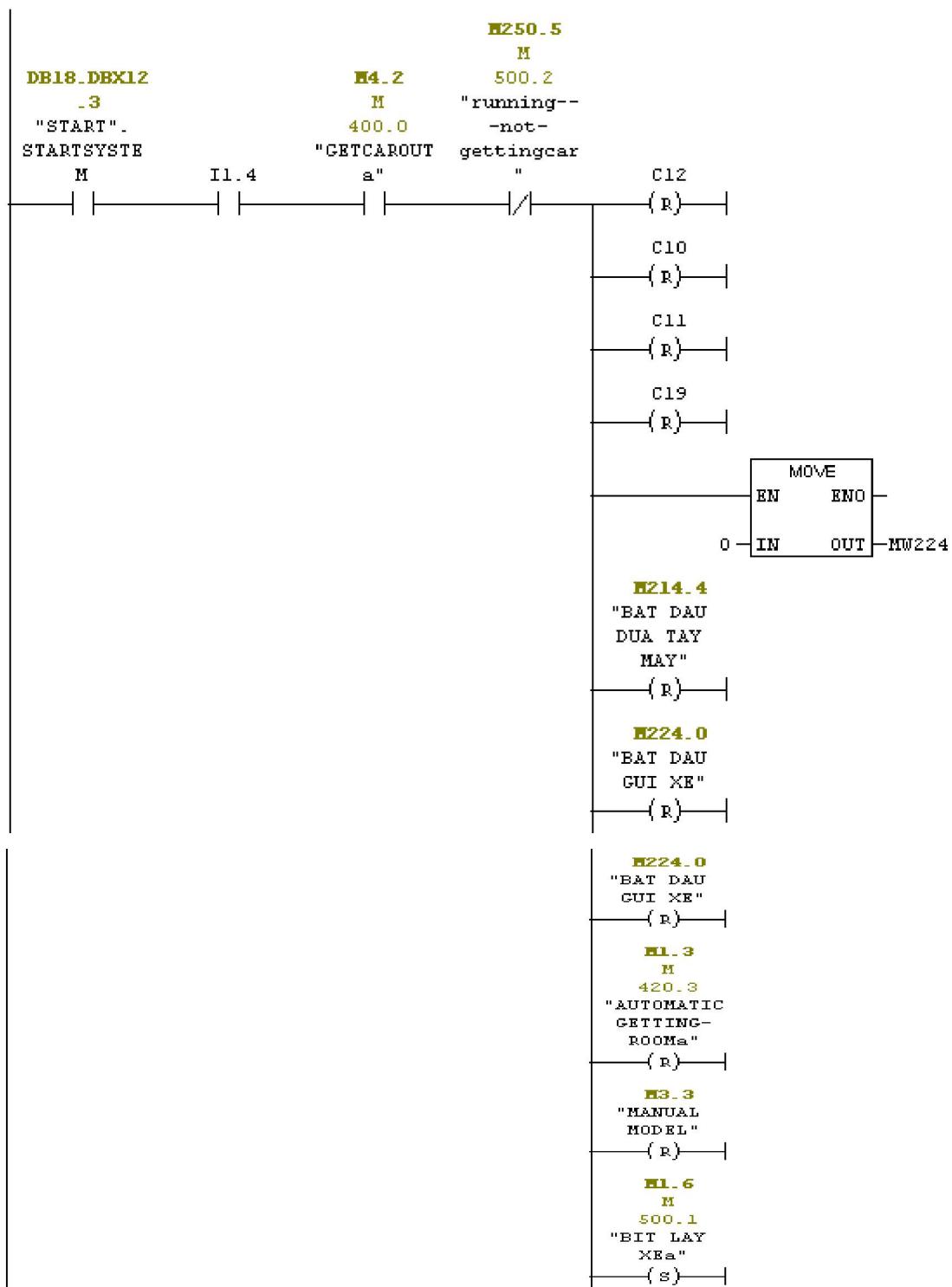
CHON CHE DO GUI XE BAN TU DONG, KIEM TRA CABIN CO DANG O TAI ZERO (PHONG GIAO TRA XE) KHONG? NEU KHONG THI RUN --> DE THUC HIEN DUA CABIN VE VI TRI ZORO TRUOC KHI NHAN XE VAO.





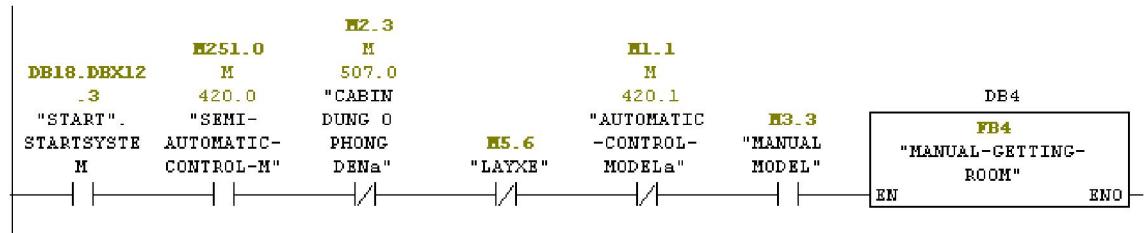
**Network 15 : Title:**

CHON CHE DO LAY XE RA, CHE DO NAY CHI DUOC THUC THI KHI HE THONG DA KET THUC QUY  
TRINH NHAN HAY TRA XE; CO THE HUY CHE DO LAY XE RA NAY (DUOC VIET TRONG CHUONG TRINH CON).



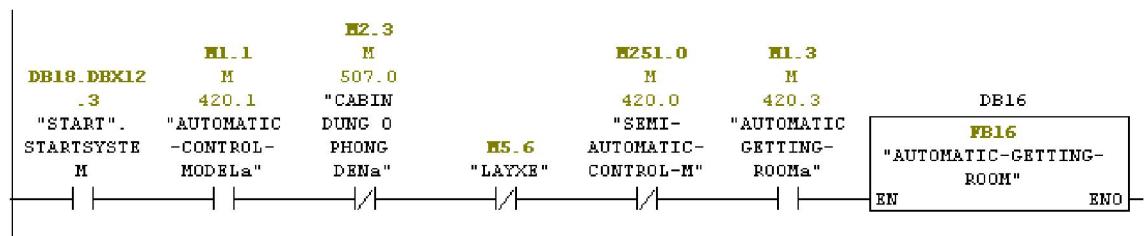
**Network 16 : Title:**

CHUONG TRINH CON CHON PHONG VA XU LY SO LIEU TRONG CHE DO BAN TU DONG



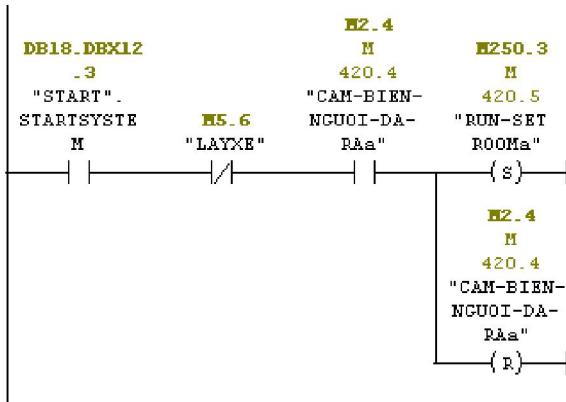
### Network 17 : Title:

CHUONG TRINH CON TIM VA CAC DINH PHONG CON TRONG TRONG CHE DO TU DONG



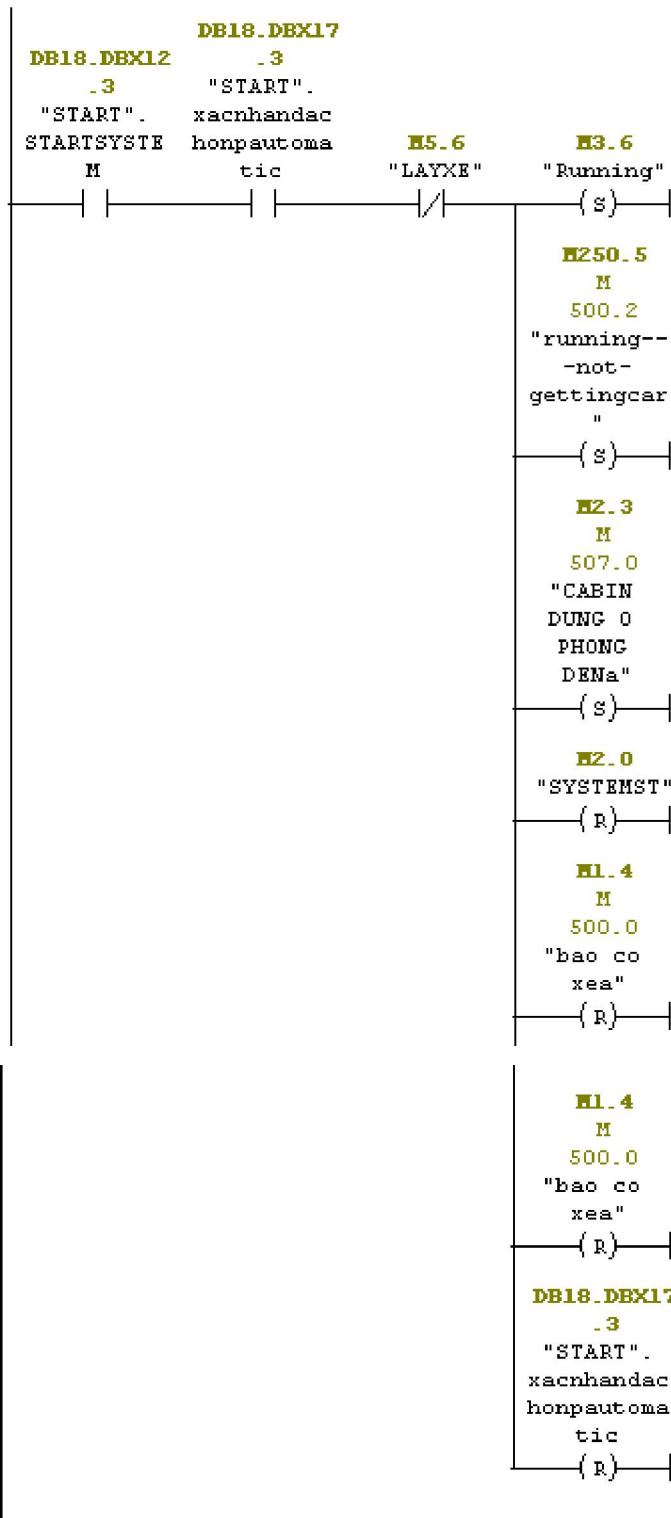
**Network 18 : M** 420.5

XAC NHAN XE DA VAO VI TRI VA NGOAI DA RA NGOAI VAO VI TRI AN TOAN, BAT DAU THUC HIEN QUY TRINH GUI XE TRONG CHE DO TU DONG



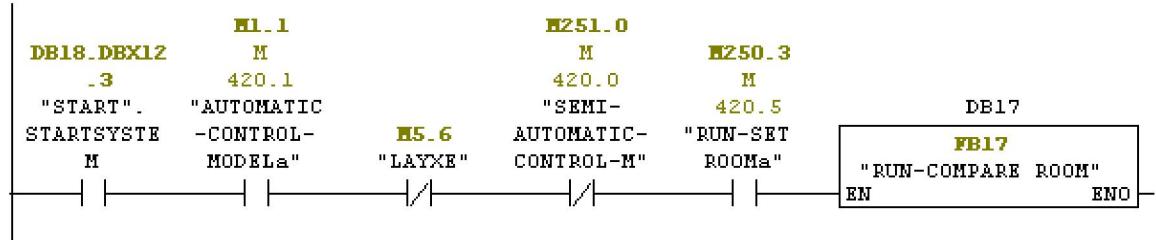
**Network 19 : Title:**

XAC NHAN XE DA VAO VI TRI VA NGOAI DA RA NGOAI VAO VI TRI AN TOAN, BAT DAU THUC HIEN QUY TRINH GUI XE TRONG CHE DO TU DONG

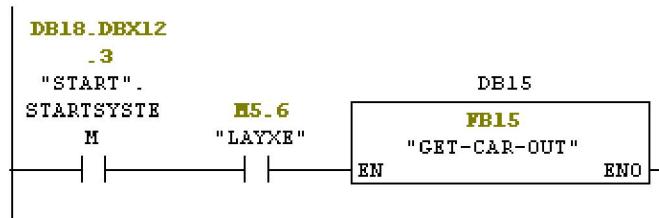


**Network 20 : Title:**

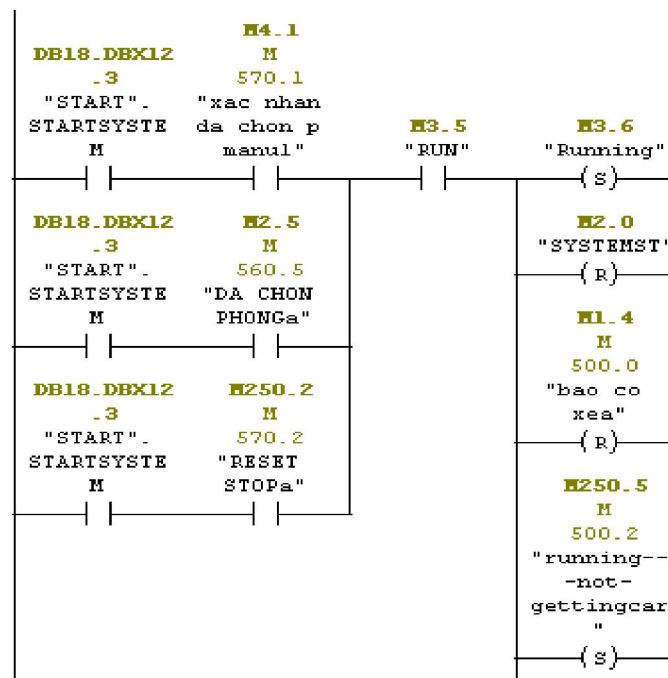
SO SANH VA XU LY SO LIEU CHO PHONG DUOC CHON TRONG CHE DO TU DONG

**Network 21 : Title:**

CHUONG TRINH CON XU LY VA TINH TOAN SO LIEU CHO QUY TRINH LAY XE RA

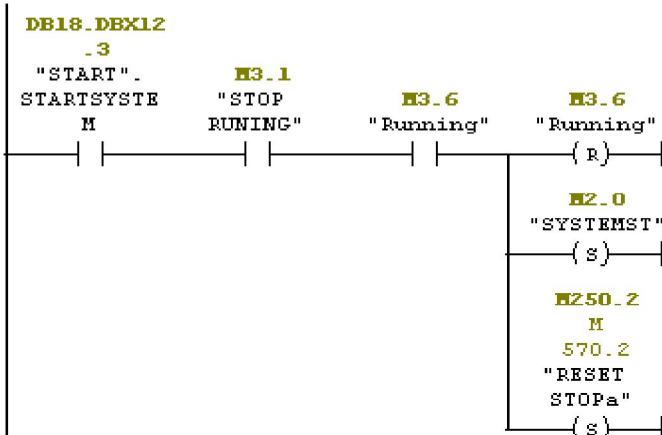
**Network 22 : Title:**

NHAN RUN DE THUC THI QUY TINH GUI XE, RUN CHI DUOC CHO PHEP NHAN KHI PHONG DA DUOC XAC DINH, HOAC DUNG DE RESET STOP KHI STOP TRONG QUY TRINH DICH CHUYEN CABIN



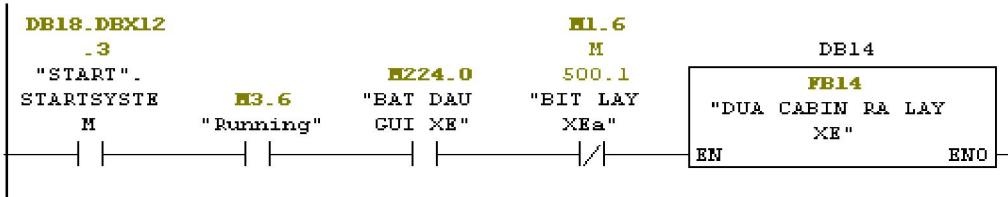
Network 23 : Title:

STOP TRONG QUY TRÌNH DỊCH CHUYỀN CABIN, CHỈ DÙNG ĐƯỢC KHI ĐANG TRONG QUY TRÌNH  
DỊCH CHUYỀN CABIN TỐI VI TRÍ MONG MUỐN



Network 24 : Title:

Comment:



Network 25 : Title:

NANG CABIN TOI LAU MONG MUON



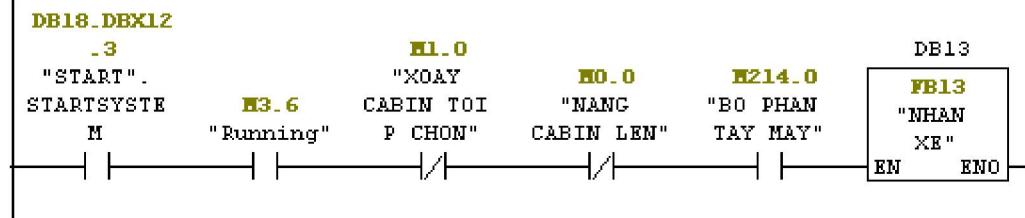
Network 26 : Title:

XOAY CABIN TOI PHONG MONG MUON



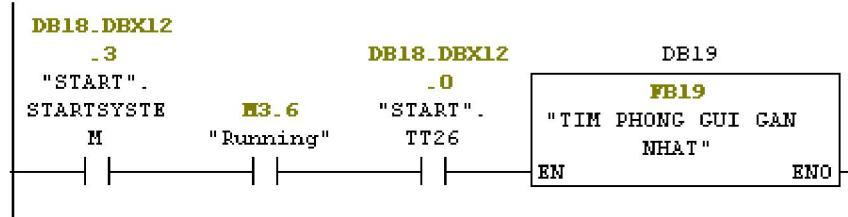
Network 27 : Title:

DUA CABIN VAO PHONG



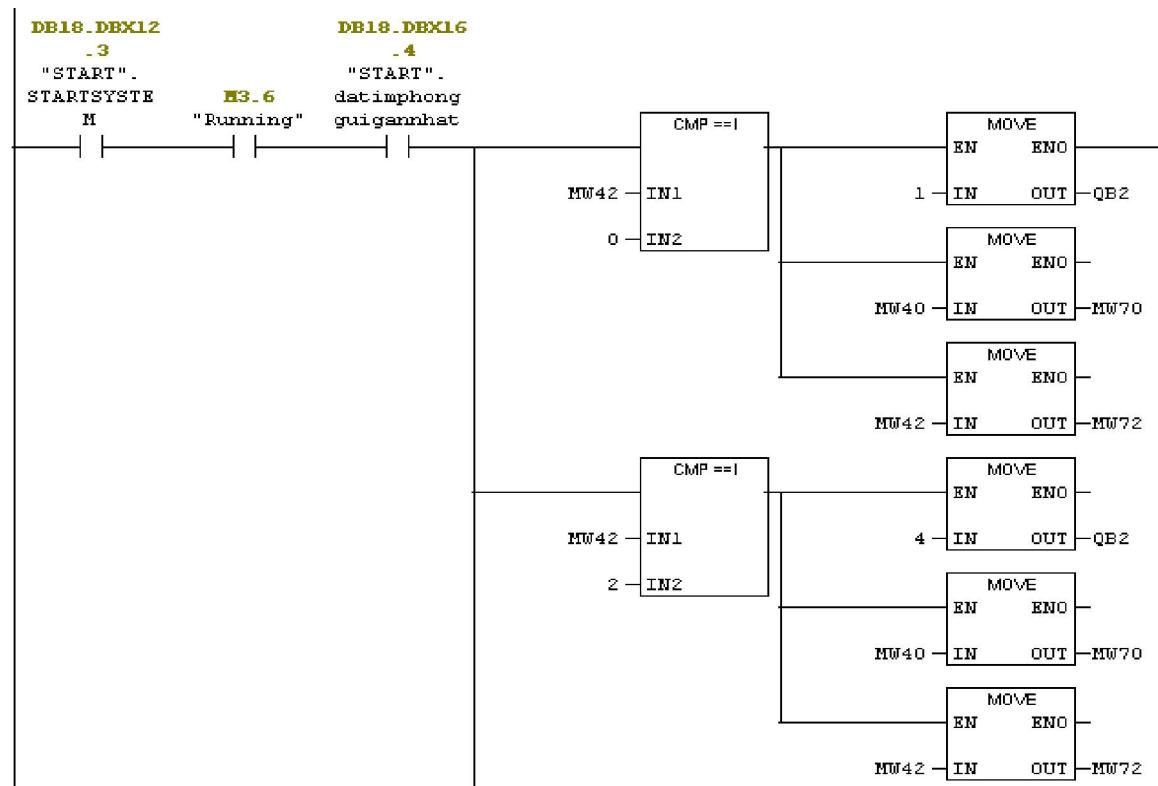
Network 28 : Title:

Comment:



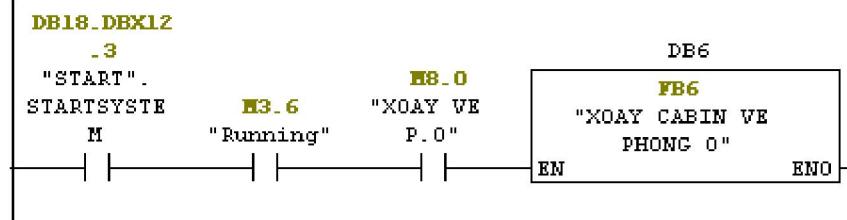
Network 29 : Title:

Comment:



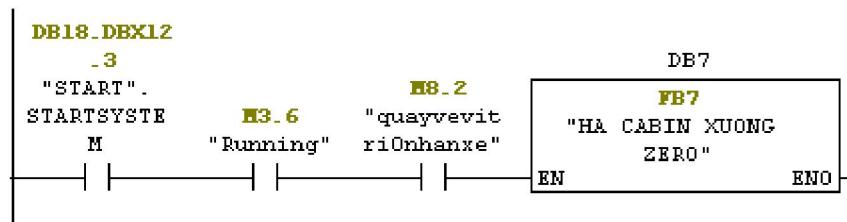
**Network 30 : Title:**

DUA CABIN VE VI TRI PHONG SO O TRUOC KHI DUA NO VE VI TRI ZERO (VI TRI GIAO TRA HAY NHAN GUI XE)



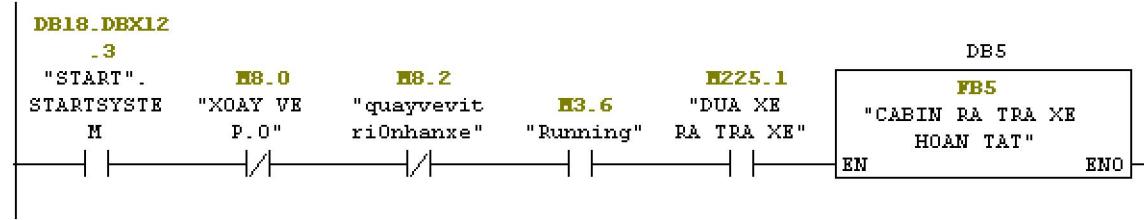
**Network 31 : Title:**

HA CABIN XUONG TOI VI TRI ZERO DE GIAO TRA HAY NHAN GUI XE



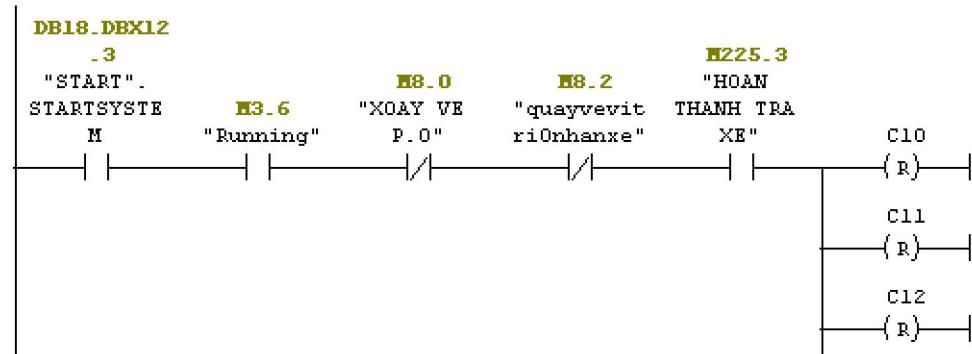
**Network 32 : Title:**

Comment:



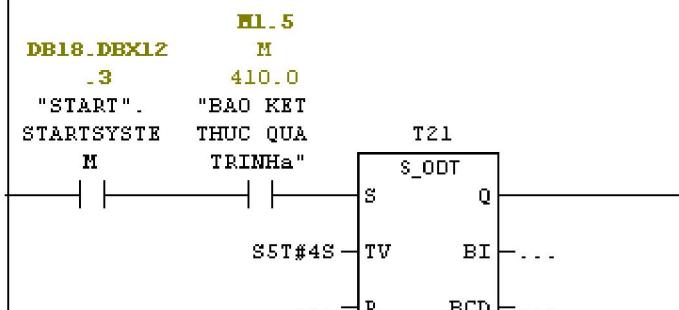
**Network 33 : Title:**

Comment:



**Network 34 : Title:**

BAO QUY TRINH GUI XE HAY LAY XE DA KET THUC THANH CONG

**Network 35 : M 410.0**

NHAP NHAY DONG CHU THONG BAO TRONG VONG 4S ROI TAT

