

UNIVERSIDAD SAN FRANCISCO DE QUITO

COLEGIO DE CIENCIAS E INGENIERÍAS

**PARQUEADERO MODULAR INTELIGENTE: PROGRAMACIÓN, DISEÑO
ELECTRÓNICO E IMPLEMENTACIÓN A ESCALA DE UN PARQUEADERO
AUTOMATIZADO CON MONITOREO SCADA**

Juan Carlos Tite Carrera

Patricio Xavier Verdesoto Vega

Omar Aguirre, M.Sc., Director de Tesis

Tesis de grado presentada como requisito para la obtención del título de Ingeniero Electrónico

Quito, Mayo 2013

Universidad San Francisco de Quito
Colegio de Ciencias e Ingenierías

HOJA DE APROBACIÓN DE TESIS

**PARQUEADERO MODULAR INTELIGENTE: PROGRAMACIÓN, DISEÑO
ELECTRÓNICO E IMPLEMENTACIÓN A ESCALA DE UN PARQUEADERO
AUTOMATIZADO CON MONITOREO SCADA**

Juan Carlos Tite Carrera

Patricio Xavier Verdesoto Vega

Omar Aguirre, M.Sc.
Director de Tesis
Director de Ingeniería Eléctrica y Electrónica

René Játiva, DEA.
Miembro del Comité de Tesis

Luis Caiza, M.Sc.
Miembro del Comité de Tesis

Ximena Córdova, Ph.D.
DECANA
Escuela de Ingenierías
Colegio de Ciencias e Ingenierías – El Politécnico

Quito, Mayo 2013

© DERECHOS DE AUTOR

Por medio del presente documento certifico que hemos leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estamos de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizamos a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art.144 de la Ley Orgánica de Educación Superior.

Firma:

Nombre: Patricio Xavier Verdesoto Vega
C. I.: 171933516-6

Firma:

Nombre: Juan Carlos Tite Carrera
C. I.: 172264163-4

Fecha: Quito, mayo 2013

AGRADECIMIENTO

A nuestras familias por su amor, comprensión, paciencia y apoyo que nos brindaron a lo largo de todo el camino recorrido durante nuestra carrera como universitarios.

A Nelson Herrera, Laurent Sass, Alberto Sánchez y todos los profesores de la carrera de Electrónica y Mecánica que nos brindaron sus conocimientos a través de los años universitarios y para hacer de esta Tesis una realidad.

A los amigos de la carrera que con su apoyo y amistad, nos brindaron la perseverancia que se necesita para culminar una ingeniería como ésta.

A la Universidad en general por abrirnos sus puertas y darnos una excelente educación.

DEDICATORIA

A mis padres Alicia y Germán, a mis abuelos Sara y Juan por su gran amor, comprensión y apoyo durante toda mi carrera universitaria.

A Dios, por sus enseñanzas y mostrarme el camino correcto y por hacerme saber que con esfuerzo y dedicación se puede alcanzar cualquier meta.

Patricio Xavier Verdesoto Vega.

A Dios por su fortaleza en este largo camino lleno de triunfos y derrotas pero siempre la bendición me hizo ganador.

A mis padres que confiaron en mí y me apoyaron regalándome una educación de excelencia y un hogar lleno de amor y por supuesto a mis hermanos y única y gran inspiración.

A Katty, mi compañera perfecta, mujer sabia y luchadora; su paciencia, su amor, su sabiduría me ayudó a coronar esta cumbre. Eres mi amor.

Juan Carlos Tite Carrera.

Resumen

El presente proyecto se centra en el diseño, programación y funcionamiento de un parqueadero automático con la finalidad de optimizar el espacio, y así, de esta manera, llegar a ser capaces de estacionar 5 vehículos en un lugar donde normalmente se estacionan 3. El parqueadero en sí tiene un sistema de control o sistema SCADA que es necesario para el correcto funcionamiento de las plataformas, y llevar un debido control de las variables importantes dentro del parqueadero como son: motores, sensores, reportes, tiempo de uso, etc.

Ésta es una idea que resuelve el problema del progresivo aumento sin límite del número de automóviles que circulan en la ciudad, los cuales requieren eventualmente un espacio para ser estacionados, y por lo tanto, se necesita optimizar dichos espacios o lugares dedicados para parqueaderos de vehículos, así construyendo parqueaderos de más de un solo nivel y automatizándolos para mayor facilidad de los usuarios. Es un proyecto ambicioso en donde se busca crear un parqueadero del primer mundo en nuestro país; actualmente estos parqueaderos funcionan perfectamente en países como Alemania, Japón, China, EEUU y otros más, en donde se ha visto el crecimiento exponencial de vehículos y por lo tanto la reducción de espacios en ciertas metrópolis del mundo.

Abstract

This project focuses on the design, programming and operation of an automatic parking in order to optimize space and thus become able to park 5 cars in a place where usually only fit three. The parking lot, itself has a control system or SCADA system that is necessary for the proper work of the platforms, and keeps a proper control of important variables of the parking lot such as: motors, sensors, reports, time of use, etc.

This is an idea that solves the problem of progressive increase without limit of the number of cars in the city, which eventually require a space to be parked, and therefore, it is necessary to optimize the spaces or places dedicated for car parking and start to build parking lots more than one level and automating them for convenience of users. It is an ambitious project which seeks to create a parking lot of the first world, in our country, these parking currently work perfectly in countries like Germany, Japan, China, USA and others, where it has seen exponential growth of vehicles and so the reduction of space or areas in certain metropolises around the world.

INDICE

1.	INTRODUCCIÓN	14
2.	MARCO TEÓRICO	16
2.1.	Conceptos Básicos.....	16
2.2.	Dispositivos de Control	16
2.2.1	PLCs (Programmable Logic Controllers)	16
2.2.2	Lógica Neumática e Hidráulica	17
2.3.	Dispositivos Eléctricos y Electrónicos	17
2.3.1.	Sensores	17
2.3.2.	Pulsadores	17
2.3.3.	LEDs.....	18
2.3.4.	Circuitos Integrados.....	18
2.3.5	Motores DC	18
2.4.	Automatización Industrial	18
2.4.1.	Automatización Fija	20
2.4.2.	Automatización Programable.....	20
2.4.3.	Automatización Flexible.....	20
3.	DISEÑO E IMPLEMENTACIÓN DEL MODELO A ESCALA	20
3.1.	Estructura Mecánica	20
3.1.1.	Estructura Mecánica Principal y Diseño de Plataformas.....	20
3.1.2.	Diseño y descripción del primer y segundo nivel	23
3.2.	Estructura Electrónica	26
3.2.1.	Descripción de las Variables a Controlar.....	26
3.2.2.	Detección de Vehículos	26
3.2.3.	Circuito Controlador de Motores	28
3.2.4.	Circuito Detector de Peso	31
3.2.5.	Ubicación de los Sensores de Tope.....	35
3.2.6.	Semaforización	36
3.2.7.	Descripción Pulsadores.....	36
4.	AUTOMATIZACIÓN, PROGRAMACIÓN Y FUNCIONAMIENTO DE LA PLANTA. 37	
4.1.	Introducción al PLC Siemens	37
4.2.	Introducción al Software Step 7/ MicroWin.....	38

4.3. Comunicación entre PLC y PC	40
4.3.1 Conceptos de Comunicación.....	40
4.4. Descripción del Programa.....	41
4.5. Variables a conectar como entradas	52
4.6 Variable a conectar como salidas	53
4.7 Designación de direcciones	54
4.8 Diagramas de flujo rutina principal.....	56
4.9 Diagramas de flujo subrutinas diferentes plataformas	57
4.9.1 Descripción y diagrama de flujo plataforma 1	58
4.9.2 Descripción y diagrama de flujo plataforma 2	59
4.9.3 Descripción y diagrama de flujo plataforma 3	61
4.9.4 Descripción y diagrama de flujo plataforma 4	63
4.9.5 Descripción y diagrama de flujo plataforma 5	65
5. SISTEMA SCADA	67
5.1 Introducción SCADA	67
5.2 Conceptos Generales.	70
5.2.1 Misión de un sistema SCADA.	70
5.2.2. Flujo de información en los sistemas SCADA.....	71
5.2.3 Adquisición de datos.....	71
5.2.4 Control.	71
5.2.5 Supervisión.....	72
5.2.6 Reportes.	72
5.3 Requisitos de un Sistema SCADA	72
5.3.1 Hardware.....	72
5.3.2 Software	74
5.4 Introducción Indusoft	75
5.5 Protocolo Modbus	76
5.6 Designación de Tags	77
5.7 Programación PLC	78
5.8 Programación Software	79
5.8.1 Descripción de las variables.....	79
5.8.2 Métodos de programación y comandos utilizados	80
5.8.3 Efectos de Movimiento.....	81
5.8.4 Descripción del programa	83

5.8.6 Aparición y Desaparición de Objetos	85
5.8.7 Alarmas	86
5.8.8 Reportes	87
6. Conclusiones y Recomendaciones	88
6.1 Conclusiones	88
6.2 Recomendaciones	91
REFERENCIAS	92

LISTADO DE FIGURAS

Figura 1. Partes de un sensor	17
Figura 2. Proceso de eventos discretos	19
Figura 3. Diseño de estructura mecánica de las plataformas en Solid Works.....	22
Figura 4. Modelo a escala construido.....	23
Figura 5. Vista frontal del modelo construido.....	24
Figura 6. Motor y acople	25
Figura 7. Fotorresistencia en la plataforma.....	25
Figura 8. Fines de Carrera.....	26
Figura 9. Diseño del circuito del sensor de presencia.....	28
Figura 10. Control de Motor DC bidireccional.....	29
Figura 11. Conexión de pines y dispositivo real.....	30
Figura 12. PCB de los circuitos para los motores y fotorresistencias.....	31
Figura 13. Diagrama de conexión del AD620.....	32
Figura 14. Diagrama de conexión vista superior.....	33
Figura 15. Diseño y simulación del circuito en Multisim.....	34
Figura 16. PLC Siemens S7-200 digital.....	38
Figura 17. Representación de contacto normalmente abierto y cerrado.....	39
Figura 18. Representación de una bobina.	39
Figura 19. Inicialización del sistema automatizado.....	43
Figura 20. Llegada de un nuevo usuario.....	43
Figura 21. Pesar Carro Nuevo.....	44
Figura 22. Verificando puesto 1.....	44
Figura 23. Verificación de puesto 2.....	45
Figura 24. No hay puesto para carros pesados.....	45
Figura 25. Verificación puesto 3 carro liviano.	46
Figura 26. No hay puesto disponible.....	46
Figura 27. Subrutina para saber dónde se encuentra plataforma 1 y encender semáforo.....	47
Figura 28. Creando espacio para que baje la plataforma 3.....	48
Figura 29. Encendido de semáforo.....	49
Figura 30. Regreso al puesto de la plataforma 3.....	50
Figura 31. Fin de ciclo plataforma 3.....	51
Figura 33. Diagrama de flujo rutina principal.....	57
Figura 34. Diagrama de flujo plataforma 1.	59
Figura 35. Diagrama de flujo plataforma 2.....	61
Figura 36. Diagrama de flujo plataforma 3.....	63
Figura 37. Diagrama de flujo plataforma 4.....	65
Figura 38. Diagrama de flujo plataforma 5.....	67
Figura 39. Niveles de Automatización Industrial.....	69
Figura 40. Modo del RS232/PPI cable multimaestro.....	76
Figura 41. Tags.....	77
Figura 42. Modelo de configuración Tags.....	78
Figura 43. Tiempo total estacionado en la plataforma 1.....	79
Figura 44. Propiedades de Objeto.....	81

Figura 45. Etiquetado de las Variables.....	82
Figura 46. Formato de Programación.....	83
Figura 47. Configuración para cambiar de color.....	84
Figura 48. Propiedades de animaciones.....	85
Figura 49. Aparición de objetos.....	86
Figura 50. Configuración alarmas.....	86

LISTADO DE TABLAS

Tabla 1. Bits de Memoria.....	42
Tabla 2. Designación de Entradas.....	54
Tabla 3. Designación de Salidas.....	56
Tabla 4. Estado de las variables a controlar.....	84

1. INTRODUCCIÓN

Desde el periodo de la revolución industrial, el hombre ha tenido la necesidad de buscar innovaciones tecnológicas, las cuales optimizan energía y simplifican tareas complejas en muchas de las operaciones simples que en la antigüedad demandaban mucho esfuerzo tanto físico como energético. Desde ese entonces se planteó la idea de automatizar máquinas para una mayor producción continua, es decir sin descanso y que funcionan durante todo un año, durante todo el día y por lo tanto bajan los costos de los productos, y como consecuencia facilitar al obrero con su trabajo. Es así entonces como la automatización de las industrias comienza a jugar un rol importante para la evolución y desarrollo del hombre, debido a que trabajadores comienzan a ser reemplazados en tareas que eran peligrosas y atentaban contra la salud del mismo. Entonces si vemos a la automatización desde el punto de vista económico, nos produce una mayor producción, reducción de costos y mayor confiabilidad.

A finales del siglo XIX se empieza a crear una necesidad de tener un control de los sistemas o máquinas automatizadas y así es como nacen los sistemas SCADA que son sus siglas en inglés y significa "Supervisory Control And Data Acquisition", estos sistemas utilizan computadoras y un sistema de comunicación para automatizar el monitoreo y control de procesos industriales. Estos sistemas conforman la parte principal de la mayoría de las empresas industriales, en las cuales diferentes plantas y maquinarias se encuentran geográficamente dispersos, ya que pueden recoger toda la información necesaria de varias fuentes y procesarla rápidamente para llevar un adecuado control de la producción y el estado de las diferentes plantas de trabajo. Por lo tanto estos sistemas son eficaces en el monitoreo y control dentro de las industrias, y así mejora la toma de decisiones operacionales. [5]

SCADA (control supervisor y adquisición de datos) ha estado presente siempre que han existido los sistemas de control. Los primeros sistemas "SCADA" utilizaban la adquisición de datos a través de largos paneles, las luces y los registradores de banda. El operador, operaba manualmente varias perillas de control ejerciendo el control de supervisión. Estos dispositivos han sido y todavía son utilizados para hacer el control de supervisión y adquisición de datos en las plantas, fábricas e instalaciones generadoras de energía. [3]

Actualmente se sabe que el crecimiento exponencial de vehículos de las principales ciudades del Ecuador, ha generado un problema en la circulación y tráfico vehicular de todas las vías dentro de estas ciudades. Esto sumado a que no existen suficientes espacios donde los choferes puedan parquear sus autos, crea un caos en las horas pico y por lo tanto la ciudadanía se ha visto en la necesidad de inventar nuevas formas de crear u optimizar espacio.

En este proyecto se plantea una solución al anterior problema, es decir diseñar y construir un modelo a escala de un parqueadero automático de dos niveles, en donde se planifica estacionar 5 vehículos en un lugar donde solo entraría 3. Con esto se ahorraría un espacio del 60% y por lo tanto se daría una solución al problema con algo innovador y tecnológico. Este modelo a escala consta de la automatización completa de cada una de las 5 plataformas que van a servir para estacionar los vehículos de manera independiente, y por supuesto también constará de un sistema SCADA en el cual se llevará un total control del parqueadero, en otras palabras se tendrá un control de la supervisión y tiempo de encendido de los motores que van adheridos a las 5 plataformas, el tiempo total que los carros permanecen estacionados, un conteo total diario de carros que utilizan el parqueadero y asimismo reportes diarios para el correcto control y funcionamiento de dicho parqueadero. Se busca obtener un sistema totalmente automático con capacidad de decisión, y memoria, estas dos características permitirán que el

modelo a escala pueda ser fácilmente manejado por el operador o usuario mediante una interfaz gráfica dentro del sistema SCADA.

Para la automatización del parqueadero se utilizará un PLC (Controlador Lógico Programable) Siemens que más adelante en los siguientes capítulos se dará una breve introducción del mismo; el PLC básicamente es el cerebro del modelo ya que mediante este dispositivo electrónico digital se ejecutarán las órdenes debidas para el adecuado funcionamiento del parqueadero. Los lenguajes que serán utilizados en este proyecto serán: Basic y Ladder para programar el PLC Siemens que consta de 24 entradas y 16 salidas, las cuales se tiene previsto utilizarlas todas en este proyecto. De hecho se utilizará una extensión del PLC para tener más capacidad de entradas y salidas.

2. MARCO TEÓRICO

2.1. Conceptos Básicos

Jon Stenerson (2003) dice que: “Un sistema automático se le define como un conjunto de equipos trabajando juntos para realizar tareas o producir un producto o una familia de productos”. Desde el siglo XVIII se inició con esta ola de desarrollo tecnológico con el fin de ayudar en la vida del hombre.

2.2. Dispositivos de Control

2.2.1 PLCs (Programmable Logic Controllers)

Este dispositivo es uno de los más utilizados en la actualidad, es un controlador de celda que es capaz de coordinar y comunicarse con los demás dispositivos electrónicos [2]. Como se mencionó anteriormente el PLC actúa como la parte master o cerebro el cual es programable de acuerdo a los requerimientos del usuario; el lenguaje de programación

utilizado se llama ladder logic, esto quiere decir que es una programación en escalera y más gráfica.

2.2.2 Lógica Neumática e Hidráulica

Son sistemas sencillos que tienen una flexibilidad limitada y son diseñados a través de válvulas hidráulicas o neumáticas.

2.3. Dispositivos Eléctricos y Electrónicos

2.3.1. Sensores

Los sensores son dispositivos que en la mayoría de aplicaciones tecnológicas se encuentran interactuando dentro de un sistema ya sea para medir, sentir y monitorear algún parámetro que se requiera. Estos dispositivos son los encargados de enviar una señal para que el sistema de control lo interprete de manera que haga funcionar los actuadores de una manera lógica. Los sensores se clasifican en dos tipos que son analógicos y digitales. Las principales partes de un sensor se muestran en la figura 1.

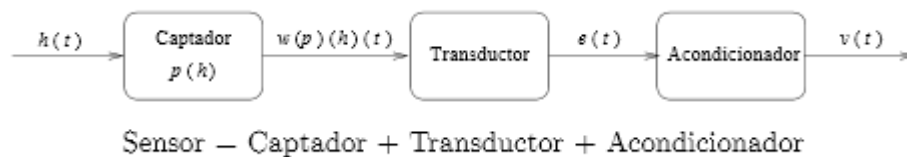


Figura 1. Partes de un sensor [11]

2.3.2. Pulsadores

Es un dispositivo muy utilizado en todos los lugares y sirve para activar alguna función. En este caso vamos a utilizar 7 pulsadores para la construcción del parqueadero que

se utilizarán para manejar las entradas y salida de los vehículos y de sus respectivas plataformas.

2.3.3. LEDs

Son dispositivos emisores de luz los cuales se usan como indicadores en algún panel o determinada planta. En la actualidad existen LEDs de diferentes colores y de alta intensidad para diferenciar un aviso de otro.

2.3.4. Circuitos Integrados

Los primeros circuitos integrados fueron creados por la empresa Texas Instruments a inicios de los años 60 [6]. De esta época en adelante comienza la producción de CI, con lo que es posible integrar los transistores dentro de una sola cajita a la cual denominaron con este nombre. Primeramente empezaron con la lógica DTL (Lógica diodo transistor) para luego hacerla más compleja con la lógica TTL a partir de la anterior [7].

2.3.5 Motores DC

Los motores DC o de corriente continua (cc) en la actualidad son utilizados en muchas aplicaciones, como accionamientos muy precisos, controlador de posiciones, accionamiento de velocidades variables para ajustar los caudales en las bombas y también se los está utilizando mucho en el área de robótica donde se requiere de mucha precisión y control de velocidad y torque [9]. Estos motores son capaces de funcionar desde unos cuantos watts hasta muchos miles de kilowatts dependiendo en que área de trabajo se los necesite.

2.4. Automatización Industrial

Como se sabe en las últimas 3 décadas, la automatización ha estado progresando muy rápidamente ya que esta, está ligada al desarrollo tecnológico y económico a nivel mundial.

Esto quiere decir que tanto la tecnología como el desarrollo de las industrias han progresado significativamente en estos últimos años.

Se sabe que la automatización y la robótica son complementarias, ya que los robots industriales están concentrados en operaciones muy simples y repetitivas y por lo tanto no necesitan de mucha precisión. Industrialmente hablando, a la automatización se la define como una tecnología que está relacionada con el uso de sistemas mecánicos-eléctricos mediante la utilización de computadoras para la operación y control de la producción. Por lo tanto la robótica es una forma de automatización industrial [12].

Básicamente en lo que se basa una automatización es que: La parte operativa recibe órdenes de la parte de mando, opera sobre el proceso productivo, los procesa y los devuelve hacia la parte de mando nuevamente para estar en comunicación con las máquinas y acondicionar las señales [11]. Ver Figura 2.

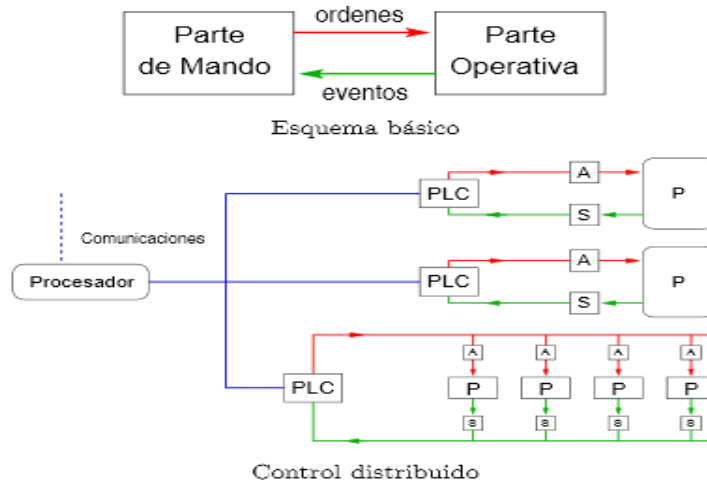


Figura 2. Proceso de eventos discretos [11]

Existen tres tipos de automatización industrial:

2.4.1. Automatización Fija

Es el tipo de automatización en la cual el objetivo principal, es obtener una producción muy alta para tener una reducción en los costos, como por ejemplo una ensambladora de automóviles. Este tipo de automatización también está relacionada con sistemas lógicos como son: compuertas lógicas y sistemas relevadores; pero con la creación de elementos de programación como es el caso de los PLC's (Controladores Lógicos Programables), se han ido flexibilizando estos sistemas [12].

2.4.2. Automatización Programable

Este tipo de automatización está más relacionada con la robótica ya que como su nombre lo dice, se necesita de una programación para crear una diversidad de productos y por lo tanto su producción no va a ser tan alta como la automatización fija.

2.4.3. Automatización Flexible

Esta automatización es más adecuada para un rango de producción medio. Se puede decir que este tipo de automatización tiene características de la automatización fija y de la automatización programada.

3. DISEÑO E IMPLEMENTACIÓN DEL MODELO A ESCALA

3.1. Estructura Mecánica

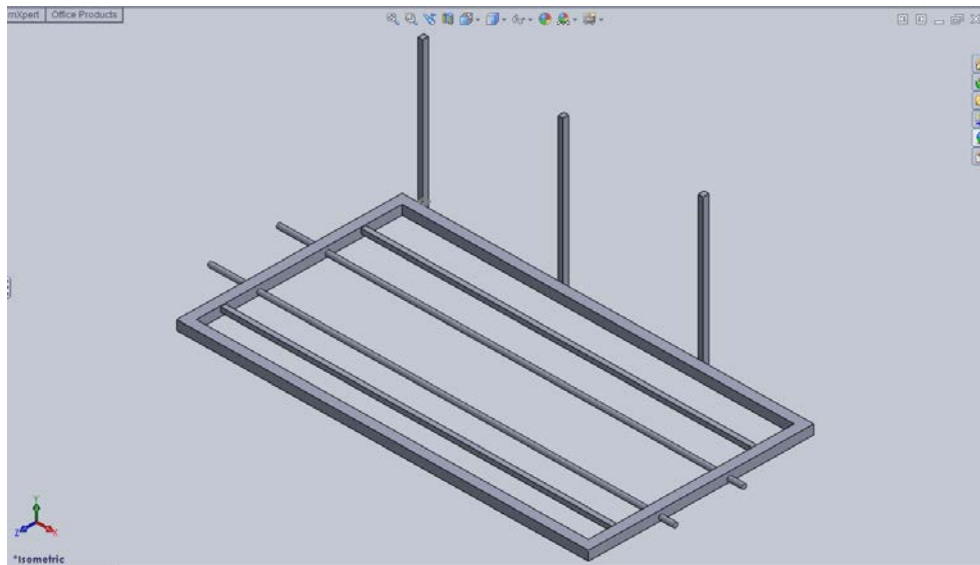
3.1.1. Estructura Mecánica Principal y Diseño de Plataformas

Para construir un modelo a escala de un parqueadero automático se procedió a realizar el respectivo diseño mecánico en el programa SolidWorks, en el cual se establecieron las

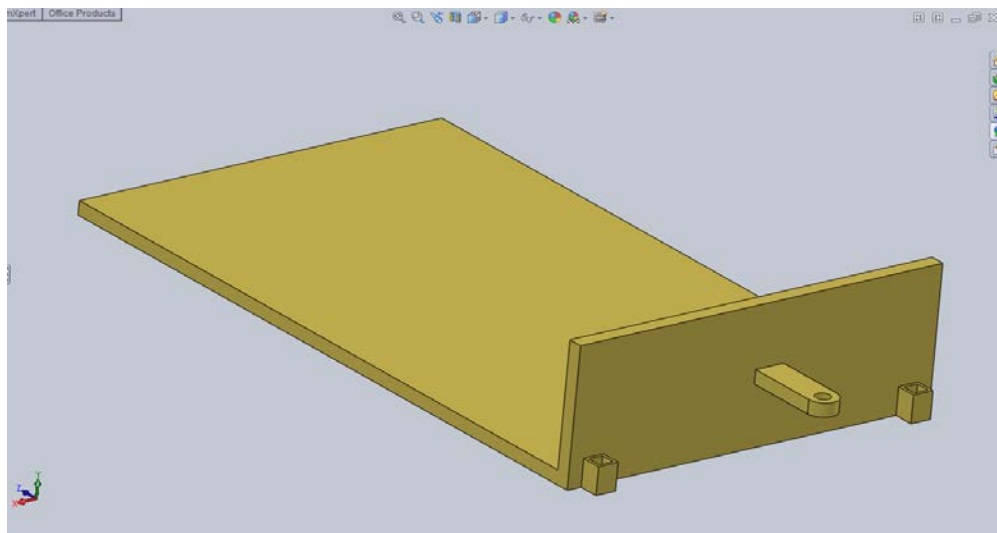
medidas y los lugares donde iban a ser ubicados los componentes necesarios para el prototipo.

La figura 3 muestra el diseño y los planos realizados en SolidWorks.

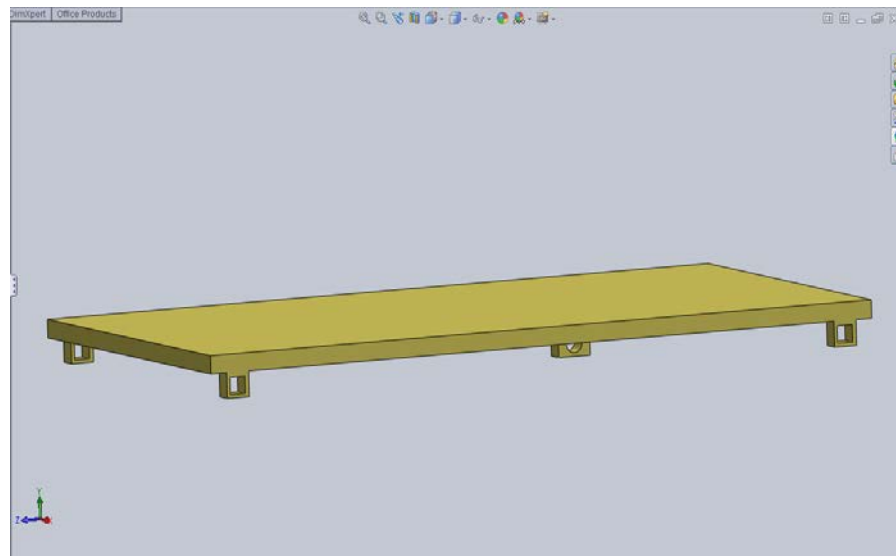
La construcción de la estructura mecánica se llevó a cabo en el Taller Mecánico de la Universidad, y se utilizaron materiales tanto metálicos como de madera para obtener una mejor estabilidad en el movimiento en los diferentes ejes de las plataformas.



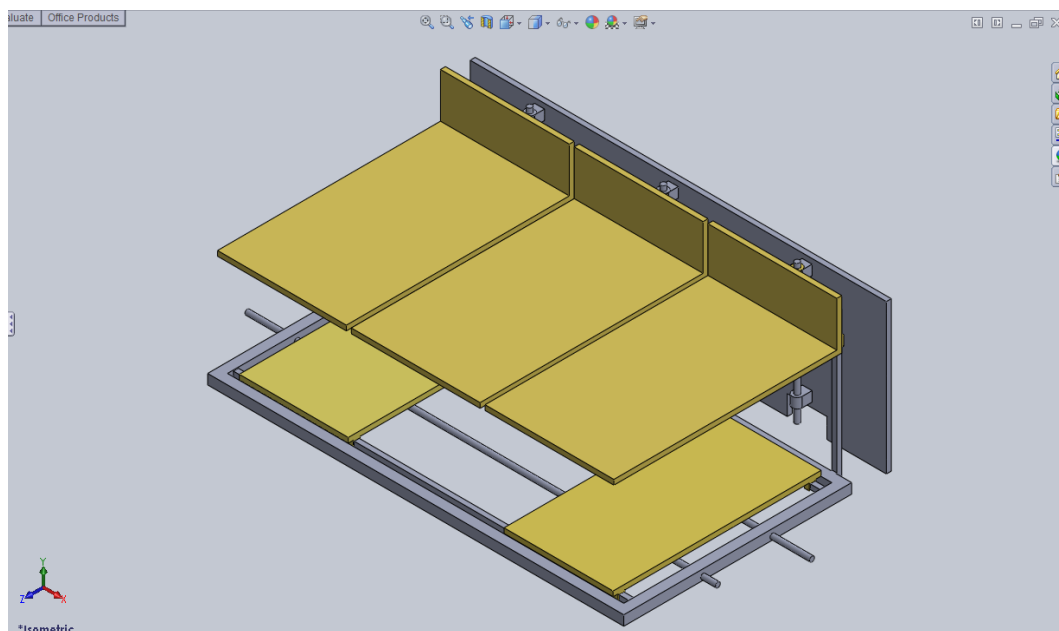
a) Estructura Metálica Principal, Ejes y Tornillos inferiores



b) Modelo para Plataformas Superiores



c) *Modelo para Plataformas Inferiores*



d) *Diseño final del parqueadero*

Figura 3. Diseño de estructura mecánica de las plataformas en Solid Works.

Para la estructura principal donde va a sostenerse en sí todas las plataformas y el parqueadero en general, se usaron vigas de madera cuadradas de dos pulgadas, y sus dimensiones tienen las siguientes medidas: 60 cm de largo por 40 cm de ancho y 40 cm de alto.

3.1.2. Diseño y descripción del primer y segundo nivel

La transmisión del movimiento hacia las plataformas se da mediante tornillos sin fin de media pulgada. Cada tornillo va unido a una plataforma, es decir existen 5 tornillos instalados en el parqueadero. Además cada uno es independiente y realiza su movimiento de acuerdo a la necesidad del usuario. Para el primer nivel, existen 2 tornillos los cuales mueven los dos espacios inferiores a lo largo del eje X dependiendo de qué carro se desee sacar del segundo nivel. Para la parte superior, en cambio, están instalados 3 tornillos, los cuales se mueven a lo largo del eje Z, es decir que con este movimiento podemos subir o bajar los automóviles del segundo nivel.



Figura 4. Modelo a escala construido

Cada tornillo se encuentra apoyado en dos rulimanes, cada uno está ubicado en los extremos para así evitar la fricción causada por la rotación de estos y obtener un movimiento rápido y estable para no forzar los motores. La figura 4 muestra una vista total del prototipo construido.

Cada plataforma está construida de madera, cada una tiene las siguientes medidas: 15.5cm de ancho por 30 cm de largo. Asimismo, cada una de ellas esta acoplada a los tornillos sin fin con tuercas; para el nivel inferior, cada plataforma consta de dos tuercas acopladas al tornillo, y para el nivel superior cada plataforma es alineada y estabilizada por una tabla posterior, que sirve de guía para direccionar bien el movimiento y evitar que las plataformas choquen entre sí, dicha guía también divide el sistema de control del sistema mecánico para un mejor ordenamiento del modelo a escala; esto se puede observar desde la vista frontal como se muestra en la figura 5.



Figura 5. Vista frontal del modelo construido

Para mover las plataformas se utilizaron 5 motores con caja reductora, de 150 rpm y 12 VDC, acoplados mediante matrimonios y prisioneros a los tornillos sin fin. Dichos motores fueron alineados con los tornillos y fijados a una base de madera mediante abrazaderas sujetas por pernos para así evitar la desestabilización de los mismos. La siguiente figura muestra un motor, su acople al tornillo y su abrazadera de sujeción.



Figura 6. Motor y acople

En cada plataforma se taladró un agujero de $\frac{1}{4}$ de pulgada con la finalidad de poder mantener fijas las fotorresistencias, que funcionan como detectores o sensores de presencia de los vehículos. También se les dio el suficiente espacio para su movilidad ya que como se sabe cada plataforma realiza un movimiento ya sea en el eje X o Z.



Figura 7. Fotorresistencia en la plataforma

Para la detección de las plataformas y apagado automático de los motores se utilizaron fines de carrera ubicados estratégicamente para detectar la ubicación de cada una de las plataformas, y transportarlas de un lugar a otro automáticamente. Estos sensores digitales son los que envían la información hacia el PLC, para así seguidamente trasladarlo al sistema SCADA y visualizar gráficamente donde se encuentran cada una de las plataformas. Se utilizaron dos modelos de fines de carrera pero que igual cumplen con la misma función que es la de enviar la información al PLC para ser procesada.



Figura 8. Fines de Carrera

Los circuitos para el funcionamiento de los componentes electrónicos también fueron diseñados y probados en el software MultiSim. En la siguiente sección se hablará más a fondo de los diseños electrónicos.

3.2. Estructura Electrónica

3.2.1. Descripción de las Variables a Controlar

Esta sección del capítulo se concentrará en el diseño de los circuitos electrónicos, tal que se procederá a la implementación de los mismos para la detección e ingreso de vehículos, funcionamiento de los motores en sentido horario y anti-horario, la detección de peso de los vehículos, semaforización, los diferentes sensores y su semaforización para una mejor señalización visual del parqueadero.

3.2.2. Detección de Vehículos

En el parqueadero se controlan los espacios disponibles y ocupados mediante sensores de presencia, los cuales en este caso son fotorresistencias. Éstas son dependientes del nivel de luz que perciben, es decir es una resistencia variable. El funcionamiento acerca de la detección y llegada de un vehículo en el parqueadero se lo describe a continuación:

En el momento que arriba un vehículo al parqueadero, el usuario mediante un botón (pulsador de nuevo carro) da aviso al sistema automatizado que un nuevo carro está llegando y comienza a buscar un espacio disponible. Cuando un carro llega a un espacio disponible, genera una sombra en el piso, lugar donde se encuentra ubicada una fotorresistencia. Entonces al momento de que el carro ingresa a un lugar del parqueadero, disminuye el flujo de luz que está llegando al sensor, lo cual produce una variación de resistencia; y mediante el acoplamiento de un circuito, ésta variación nos permite activar un relé, lo cual es la señal que se envía al PLC.

El circuito de acoplamiento básicamente consiste en trabajar con un BJT en modo de corte y saturación, o en otras palabras como interruptor, la corriente de base es controlada con un potenciómetro y la fotorresistencia. El potenciómetro controla la sensibilidad y la fotorresistencia controla la activación del sensor. Se utiliza una configuración “Darlington” para la ganancia de corriente de colector, la cual activa el relé.

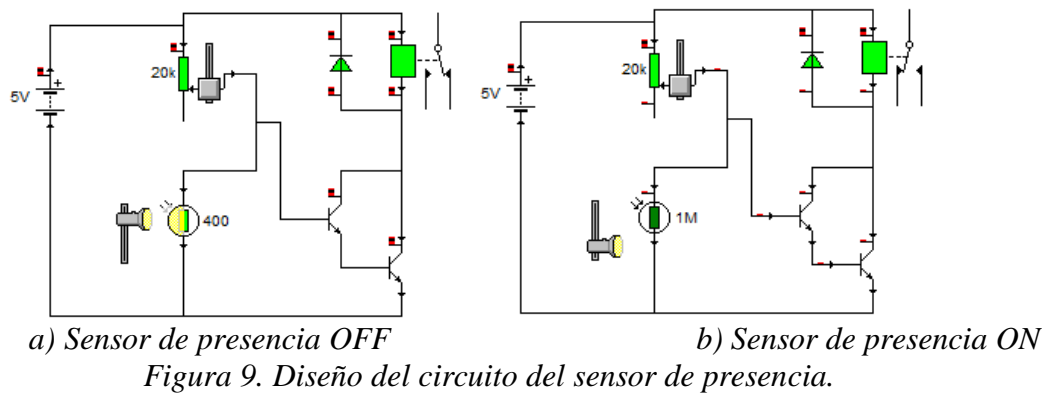
Para enviar la señal desde relé al PLC, se utiliza los contactos normalmente abierto y el común, es decir cuando un vehículo provoca una disminución de luz que llega al sensor, el relé cierra los contactos antes mencionados y esto lógicamente envía señal de activación, o “1” hablando en lógica digital.

Al momento en que el carro se retira del lugar, la bobina del relé se desactiva y vuelve a su posición original, se abre el circuito, se deja de enviar señal o se envía un “0”.

Se utiliza un programa de simulación electrónica (crocclip), para el diseño del circuito mostrado en la figura 9. La representación de que el parqueadero está libre se lo hace mediante la lámpara que muestra la intensidad que llega a la fotorresistencia y como se puede observar,

cuando hay luz el relé está desactivado en la posición normalmente abierto la cual es la que se utiliza (figura 9a).

En la figura 9b, se muestra el circuito cuando el sensor detecta una disminución de luz que es provocada a la llegada de un vehículo y como se observa en la figura, esto es simulado apagando la luz lo cual detecta la fotorresistencia y esto hace que el relé se active en normalmente abierto enviando la señal al PLC.



Pues bien, como se explicó anteriormente, estos sensores de presencia son necesarios para la programación y automatización del parqueadero ya que aquellos, nos envían la información necesaria para el correcto funcionamiento de las plataformas.

3.2.3. Circuito Controlador de Motores

Los motores constituyen una parte importante dentro del desarrollo del proyecto, ya que éstos son lo que movilizan todo el diseño mecánico de cada una de las plataformas. Primeramente se dará una pequeña introducción al circuito integrado utilizado para el control de giro de cada uno de los motores para luego ser diseñados e implementados en el PC. Para controlar el cambio de giros de los motores, se utiliza un puente H, para el cual por facilidad se dispone de un integrado. Este integrado controla el giro dependiendo de la

configuración lógica que se le envíe. Entonces para que el puente H envié las señales al motor y éste gire para un lado o al otro, necesita de un configurador lógico el cual es el PLC.

3.2.3.1. Funcionamiento Circuito Integrado L298N

El circuito integrado L298N, o mejor conocido como puente H, es un circuito constituido por 15 pines que es capaz de soportar altas corrientes, que van desde 0 hasta 4 amperios; y también altos voltajes hasta los 46 voltios. Es un puente controlador doble diseñado para aceptar niveles lógicos TTL [13].

Se escogió este circuito integrado debido a que los motores DC que se van a manipular pueden llegar a corrientes de arranque de hasta 1.5 A y por lo tanto se necesita una protección para los demás dispositivos electrónicos, que de hecho brinda este puente H. Las 2 entradas habilitadas que se tienen en este CI son para permitir o deshabilitar el dispositivo independientemente de la señal de entrada. A continuación se da un breve esquema para la conexión de un motor DC de acuerdo con el datasheet de este dispositivo en la siguiente figura.

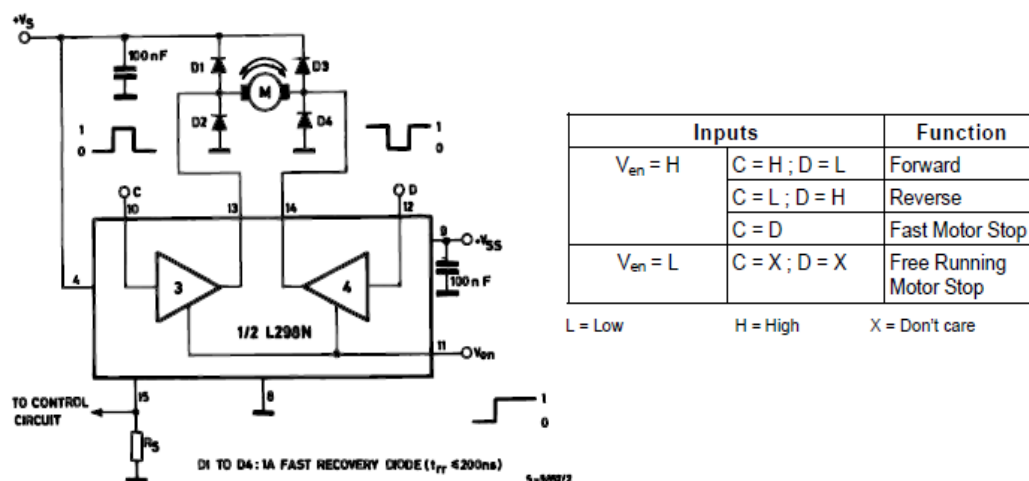


Figura 10. Control de Motor DC bidireccional.

De la misma manera, como el circuito integrado consta de 4 entradas, y 2 habilitadores o enables, es posible ahorrar dispositivos de estos y conectar 2 motores a un mismo CI; esto quiere decir que se pueden controlar dos motores en los dos sentidos de dirección con un solo dispositivo, porque el primer motor va conectado a las 2 primeras entradas y al primer enable, y el segundo motor va conectado a las dos últimas entradas y el segundo enable. De esta manera las 4 salidas que se disponen se dividen en dos para cada motor. Por lo tanto para el control de los 5 motores que demanda el proyecto, solo se utilizaron 3 dispositivos L298N. Ahora bien, para tener un estado alto o high, se conectó a una fuente de 7 V debido a que se optimizaron los dispositivos y las fuentes, porque esta misma fuente también alimenta y da paso al funcionamiento de los LEDs que están ubicados como semáforos en cada puesto. Para una mayor comprensión y conocimiento de los pines se muestra lo siguiente:

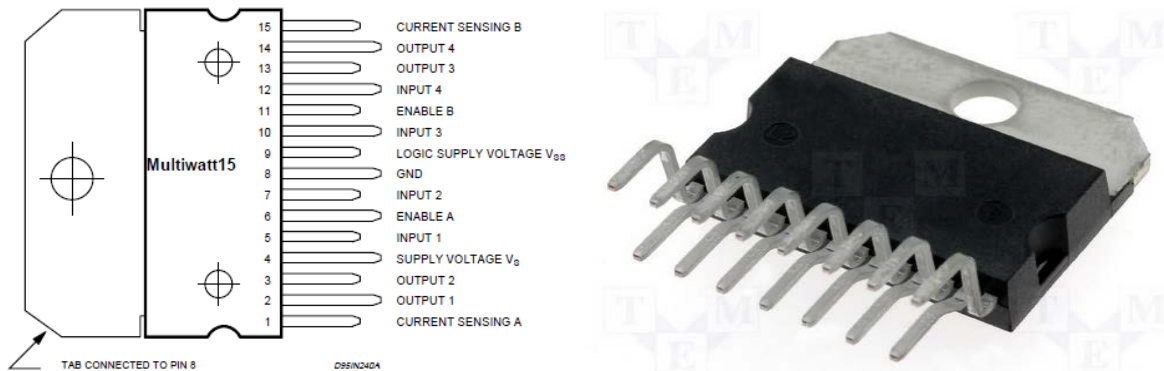


Figura 11. Conexión de pines y dispositivo real.

3.2.3.2. PCB para los 5 Motores

Una vez analizados y realizados los respectivos diseños para el funcionamiento de los motores y control de giro con su respectivo puente H, se procedió a implementarlo en un PCB de manera que sea estable y no exista ningún tipo de inconveniente al momento de su funcionamiento en el proyecto. De esta manera se obtuvo el siguiente PCB en una placa para su respectiva conexión con el PLC y la fuente de alimentación.

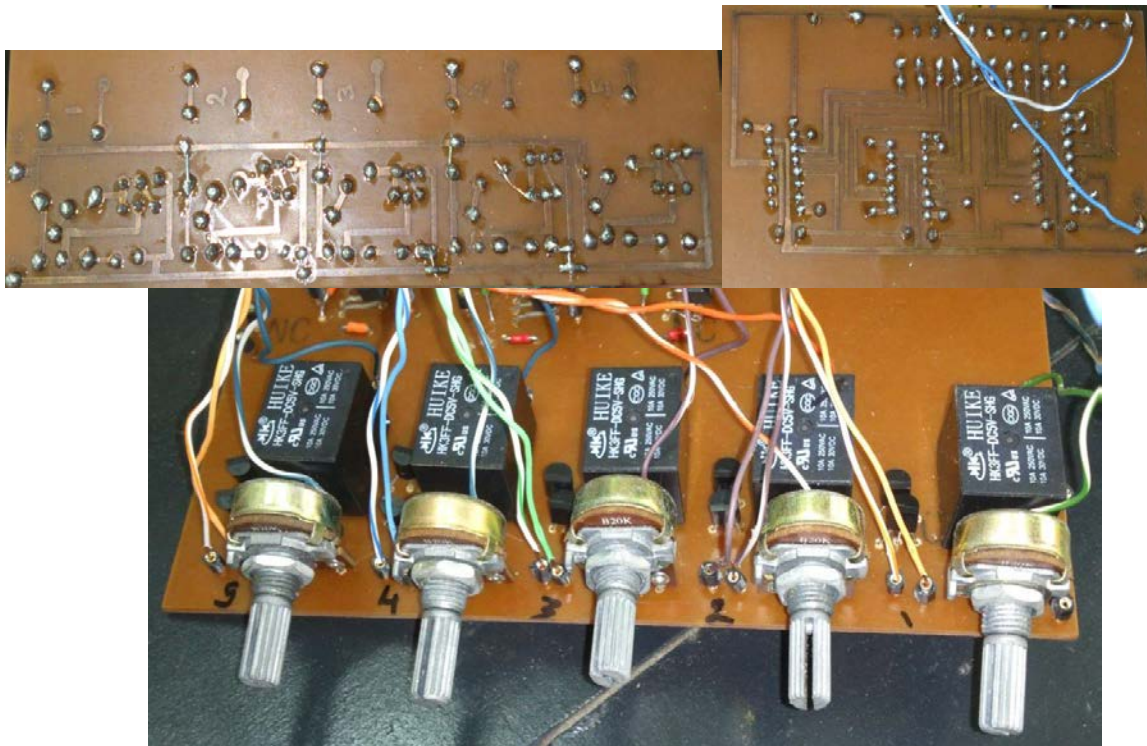


Figura 12. PCB de los circuitos para los motores y fotorresistencias.

3.2.4. Circuito Detector de Peso

En esta sección se analizará y hablará acerca de los CI utilizados para el diseño de un circuito que de acuerdo al peso vaya distribuyendo los puestos del parqueadero, ya que se tiene como condición que los carros livianos se estacionen en el segundo nivel y los pesados en el primero, las señales enviadas al PLC son procesadas para luego enviar una orden a los

actuadores e indiquen donde se puede estacionar. Para el control de peso se utiliza los siguientes materiales:

- Una balanza
- Un amplificador de instrumentación AD620
- Un AOP TL084
- Resistencias, transistor 2n3904, relé.

3.2.4.1. Circuito Integrado AD620

Este dispositivo se lo llama amplificador de instrumentación de alta precisión que requiere de una resistencia externa para ajustar las ganancias de 1 a 10000. Este dispositivo es muy sensible a factores externos como la estática, por lo que hay que manejarlo cuidadosamente. En cuanto al consumo de energía, se puede decir que este dispositivo tiene una corriente de alimentación máxima de 1.3 mA, lo que quiere decir que es óptimo para aplicaciones portátiles. Es ideal para su uso en sistemas de adquisición de datos de precisión, tales como básculas y las interfaces de transductores. Además, el bajo nivel de ruido, el sesgo de baja corriente de entrada, y la baja potencia del AD620 hacen muy adecuado para aplicaciones médicas [14]. Para la polarización de este circuito integrado se lo hace con 15V+ y 15V- .

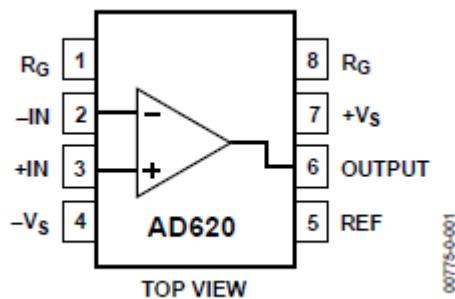


Figura 13. Diagrama de conexión del AD620 [14]

3.2.4.2. Circuito Integrado TL084

El amplificador operacional TL084 en esta aplicación se lo va a utilizar como un comparador, ya que es aquí en esta parte del circuito donde se compara la señal de entrada con una señal de referencia, la cual está regulada por un potenciómetro de precisión, y esta señal antes de llegar a este OPAM es amplificada por medio del AD620. Este dispositivo es polarizado con +15V de manera que funcione correctamente. Los dispositivos cuentan con altas velocidades de rotación, sesgo de entrada bajo y las corrientes de compensación.

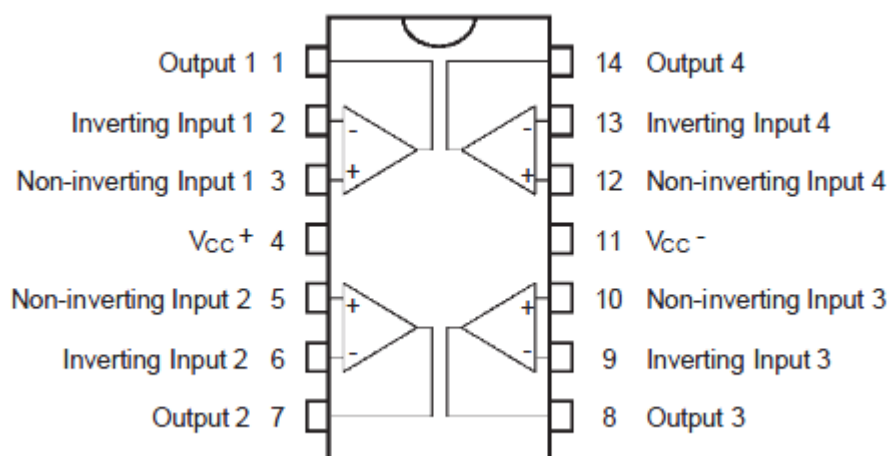


Figura 14. Diagrama de conexión vista superior [15].

Hay que tener en cuenta que este circuito integrado consta de 4 amplificadores operacionales que se los utiliza como comparadores o según el requerimiento del caso mejor dicho. En este proyecto sólo se utilizó uno de los OPAM ya que se necesita comparar una sola señal que sale del amplificador de instrumentación.

3.2.4.3. Descripción del Circuito

Para el diseño del sensor de peso se tienen dos opciones, la primera es utilizar una balanza, encontrar los terminales del voltaje, amplificar, y esta señal analógica enviar al PLC, mediante un módulo de entradas análogas. La otra opción es igualmente utilizar una balanza, tomar el voltaje amplificar, y comparar con un voltaje referencia, y mediante esto poder enviar señales digitales al PLC.

Como no se dispone de un PLC con entradas análogas se procedió a realizar la segunda opción. Básicamente el sensor de peso, es la balanza, que por medio de una galga, transforma la fuerza en voltaje, este voltaje está en el orden de los mV, es por esto que se utiliza un amplificador de instrumentación (AD620), para amplificar esta señal y tener un voltaje en el orden de los Volts.

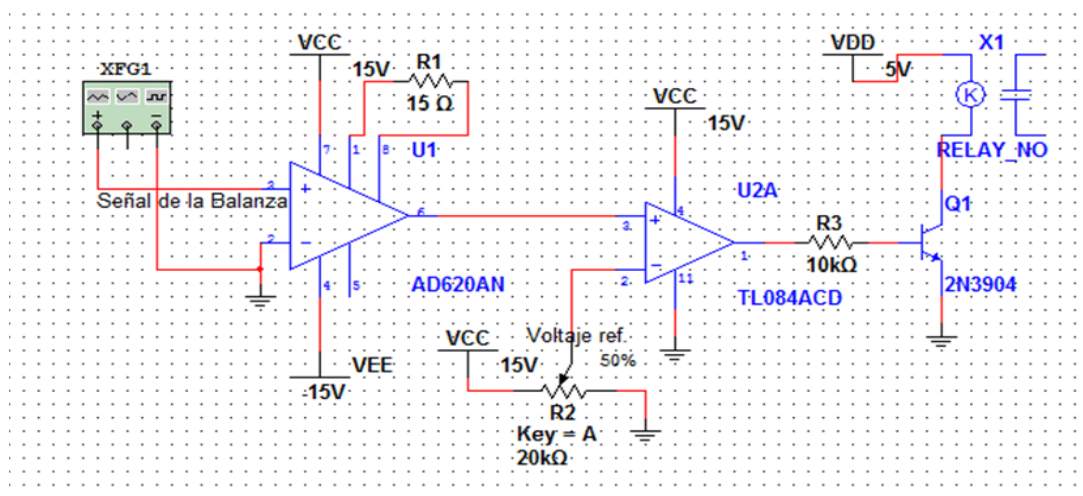


Figura 15. Diseño y simulación del circuito en Multisim.

Siguiendo con el diagrama del circuito de la figura 15, se observa que la señal de la salida del amplificador de instrumentación se la compara con una señal de referencia, esto se lo hace con un TL084 lo cual es un OPAM. A la salida de este amplificador se obtiene una diferencia de voltaje, el cual cuando éste sea lo suficiente para polarizar la base del transistor,

el transistor se polariza y como el BJT está configurado para que funcione como interruptor entonces, activa el relé y envía la señal al PLC.

Esta señal es enviada cuando la señal que entra al terminal positivo (+) del amplificador operacional es mayor que la señal que entra al terminal negativo (-). Esto sucede cuando la señal de la balanza es alta, o en otras palabras cuando la balanza detecta un carro pesado. Para designar si es carro pesado o liviano se lo hace controlando el potenciómetro el cual regula el voltaje de referencia. Entonces mediante esto el potenciómetro regula la sensibilidad del sensor de peso.

3.2.5. Ubicación de los Sensores de Tope.

Los sensores de tope o comúnmente conocidos como "fines de carrera", tienen como finalidad detectar cada una de las plataformas en movimiento y hacer que se detenga un determinado motor en el momento que entra en contacto con este sensor. Es decir cuando el primer puesto de parqueo está en movimiento hacia la derecha, existe un fin de carrera ubicado estratégicamente entre las dos plataformas del primer nivel, tal que cuando entra en contacto el primer puesto que está moviéndose hacia la derecha con el segundo puesto, éste hace que el motor de la primera plataforma se detenga y deje de continuar con el movimiento antes mencionado.

De la misma manera existen sensores de tope ubicados en la parte superior e inferior del parqueadero para tener un control absoluto de cada uno de los puestos independientes (2 nivel inferior y 3 nivel superior); esto ayuda a visualizar la ubicación de cada uno de los puestos en la interfaz gráfica del SCADA. En total los sensores de tope utilizados en la construcción del modelo a escala son 9. En la figura 8 se pueden observar dichos sensores.

3.2.6. Semaforización

En la figura 4 y 5 se pueden apreciar los semáforos ubicados en la parte superior del parqueadero.

Esto permite al usuario una mayor facilidad en cuanto a la ubicación de su vehículo, y que al momento de ingresar hacia el parqueadero exista una mayor rapidez de donde irá ubicado el vehículo ya que los semáforos se pondrán en verde dependiendo de la plataforma que se le haya asignado a dicho carro.

3.2.7. Descripción Pulsadores

El parqueadero cuenta con 7 pulsadores independientes para realizar una función específica designada. Existen 5 pulsadores que están ubicados al lado izquierdo del parqueadero, los cuales están programados para manejar cada una de las 5 plataformas respectivamente, y así el momento que el usuario quiera sacar su vehículo simplemente debe presionar el pulsador correcto que se encuentra numerado del 1 al 5.

Dos pulsadores adicionales se encuentran dentro del diseño del modelo a escala, ubicados de acuerdo a la función que cumplen y son: el pulsador OK y el pulsador de Nuevo Carro; estos dos pulsadores funcionan para asegurar al usuario que el vehículo está estacionado y para hacer saber al PLC que un nuevo carro está llegando al parqueadero respectivamente. Son de mucha ayuda estos botones, ya que así se envía información hacia el PLC para el control del parqueadero y por ende también esta información es recibida en el sistema SCADA para el debido despliegue en la interfaz gráfica y adquirir datos de cuantos carros están entrando diariamente.

4. AUTOMATIZACIÓN, PROGRAMACIÓN Y FUNCIONAMIENTO DE LA PLANTA.

En este capítulo se hablará ampliamente acerca de todos los pasos a seguir para la automatización completa del parqueadero y sus diferentes conexiones, comunicaciones e interfaces que se tiene entre la parte mecánica del proyecto con la parte electrónica. Es decir la forma de comunicación entre el PLC con los circuitos detallados en los anteriores capítulos y de esta manera que se pongan en acción los actuadores (motores, LEDs) para el correcto funcionamiento de la parte mecánica.

4.1. Introducción al PLC Siemens

Existen muchos modelos de PLC's, y entre estos está la gama de los S7-200, el cual se utiliza en muchos sistemas de automatización, algunos en pequeñas automatizaciones (Micro-PLCs) [1]. Estos dispositivos, que comprenden y se basan en una programación lógica para su funcionamiento, se los utiliza en numerosas tareas debido a su diseño compacto, su bajo costo y su facilidad al momento de programarlo. Una de sus ventajas principales es que tiene una amplia gama de operaciones que se pueden introducir para que las automatizaciones sean más precisas y eficaces.

En el desarrollo de esta tesis, se escogió este tipo de PLC porque es fácil de manipular debido a su tamaño compacto y como se mencionó anteriormente, encaja perfectamente en la automatización de un parqueadero debido a la gran cantidad de entradas y salidas que se necesitan para que cada uno de los dispositivos funcione correctamente en conjunto con el sistema SCADA que más adelante se detallará con mayor precisión. En este caso se utilizó un PLC Siemens S7-200 de 24 entradas y 16 salidas para la implementación del parqueadero.



Figura 16. PLC Siemens S7-200 digital.

Este dispositivo tiene incorporado una unidad central de procesamiento (CPU), en donde se carga y ejecuta el programa para la realización de la automatización o el proceso. Los diodos luminosos que se encuentran en la parte izquierda superior de la figura 15, indican el modo de operación de la CPU (RUN o STOP), el estado de las salidas y entradas digitales, y así se puede tener una mayor visualización y control de algún fallo en el sistema.

4.2. Introducción al Software Step 7/ MicroWin

Primeramente hay que mencionar que este software es un medio de comunicación entre el PLC y la computadora, eso quiere decir que dentro de este programa se empieza a escribir el código de programación para luego ser cargado en el PLC. Este software utiliza un lenguaje de programación ladder logic, este tipo de lenguaje se lo conoce también como programación escalera, que básicamente es una representación simbólica de los dispositivos de un circuito eléctrico, y esto facilita el entendimiento y el manejo para el usuario en caso de que nunca haya visto un PLC.

Los PLCs son programados principalmente en este tipo de programación y la principal función que tiene este tipo de programación es de controlar las salidas, que se las llamará bobinas, basadas en las condiciones de las entradas que se los llamarán contactos [2].

En su mayoría, las entradas de los PLC son dispositivos sencillos que están en un estado alto (1) o bajo (0), o se puede también decir en un estado de encendido o apagado. Estas entradas son sensores e interruptores que detectan cierta presencia, mandando un estado que cierran o abren los contactos.

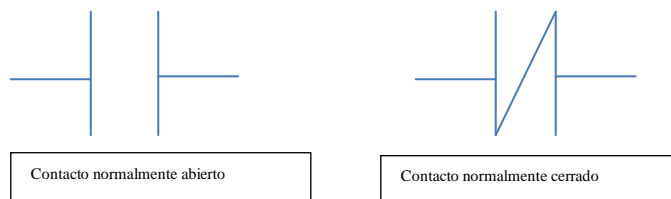


Figura 17. Representación de contacto normalmente abierto y cerrado

Como se mencionó anteriormente las entradas son los contactos, y las salidas son las bobinas. En general y en la realidad existen muchos dispositivos de salida como son: luces, relojes, motores, contadores entre otros. En esta programación de escalera se toma a la bobina como dispositivo de salida para la programación, en la cual podemos encenderla o apagarla, de acuerdo al requerimiento del programa o de la aplicación, seteando en 1 o 0. En la siguiente figura se dará su representación simbólica dentro de la programación. Se llama programación escalera porque se ve similar a una escalera.



Figura 18. Representación de una bobina.

4.3. Comunicación entre PLC y PC

Para una óptima comunicación entre la computadora y el PLC, se utiliza un cable RS232 con protocolo PPI, el cual es un protocolo muy conocido llamado maestro/esclavo, es decir los maestros envían peticiones a los esclavos. Es una interfaz punto a punto.

La comunicación serial RS232, utiliza un puerto serial la cual es una interfaz física de comunicación en serie. Hoy en día se utiliza mucho la comunicación en serie ya que la comunicación en paralelo requiere de varios cables y esto implica mayor gasto aunque también tiene algunas ventajas. La comunicación en serie se envía la información enviando y recibiendo un bit, a través de ambos pines de transmisión y recepción. Se utilizó los conectores DB9 a través de un cable UTP y también los conectores USB para la comunicación serial entre computadora y PLC.

Es necesario establecer las condiciones en ambos dispositivos conectados, mediante el DB9 hembra la cual hace la conexión RS232 para tener una óptima comunicación entre ellos.

4.3.1 Conceptos de Comunicación

4.3.1.1. Fuente o nombre de recurso: es un parámetro que designa el puerto que será utilizado para la comunicación de algún programa.

4.3.1.2. Velocidad de transmisión: determina los bits por segundo a los cuales las computadoras se encuentran transmitiendo o en comunicación, es decir es la velocidad con la que se están comunicando a través del DB9.

4.3.1.3. Bits de Datos: Este parámetro determina el tamaño de cada paquete de datos que está siendo enviado y recibido en la comunicación. Este parámetro debe ser configurado dentro del MicroWin.

4.3.1.4. Bit de paridad: Este bit es para la detección de errores y va al último de cada mensaje, esta forma de detección de errores es básica y muy elemental y sirve para mensajes cortos, es decir un frame o tren de bits pequeño.

4.3.1.5. Bit de Parada: Es el bit que indica que un mensaje ha terminado. Se especifican el número de bits de parada utilizada en la comunicación.

4.4. Descripción del Programa

Para los finales de carrera, se usaron y conectaron los contactos normalmente abiertos, de tal manera que cuando una plataforma llega al destino necesitado, presiona el interruptor del final de carrera, y éste cierra el contacto y envía señal a la entrada conectada del PLC.

De la misma manera, las salidas de los relés de las fotorresistencias fueron conectadas en el contacto normalmente abierto, para que cuando se ocupa un lugar, el sensor activa el relé, y los contactos se cierran enviando una señal al PLC.

Igualmente, hablando de programación, fue necesario el uso bits de memoria, de flanco positivo para un mejor funcionamiento del programa y de inicialización. Los bits de memoria están tabulados en la tabla 3, y cada uno realiza una función importante que ayuda al desarrollo del programa dentro del PLC y almacenan por un cierto periodo de tiempo una sentencia lógica para luego usarlas de manera de contactos dentro de la programación del parqueadero en MicroWin. Hay que recalcar que estos bits no están tan ordenados como los de entradas y salidas, ya que estos se iban creando o fueron necesarios durante el transcurso de la escritura del programa.

Bit de Memoria	Descripción
M0.0	Memoria salida 3
M0.1	Esperar usuario
M0.2	Memoria salida 4
M0.3	Fin de ciclo
M0.4	Memoria salida 5
M0.5	Memoria salida 1
M0.6	Memoria salida 2
M0.7	Carro pesado
M1.0	Carro liviano
M1.1	Pesar
M1.2	No hay puesto

Tabla 1. Bits de Memoria

Los bits cuyo nombre tiene las palabras “Memoria salida #” son los que inician las subrutinas de cada plataforma. El bit llamado “Fin de Ciclo” es usado solamente para las plataformas 3, 4 y 5, y se activa después de que el usuario ha presionado el botón “OK”, y se inicia el ascenso de las plataformas; y se desactiva cuando la plataforma ha llegado a su posición superior. Los demás bits de memoria tienen su descripción en la tabla 1, que explica de manera obvia su uso.

El bit SM0.1, es el de inicialización, y solo se ejecuta una vez en el programa. Este es usado al inicio, cuando se enciende el sistema, para prender las luces rojas de los semáforos desde el principio.

Primeramente se inicializa el programa (bit SM0.1), encendiendo todos los semáforos en rojo, de manera que se indique que la automatización y ejecución del parqueadero ha empezado. Las plataformas del segundo nivel están en su posición máxima superior y las del primer nivel deben estar haciendo contacto en algún final de carrera. A partir de este momento, se esperan órdenes del usuario.

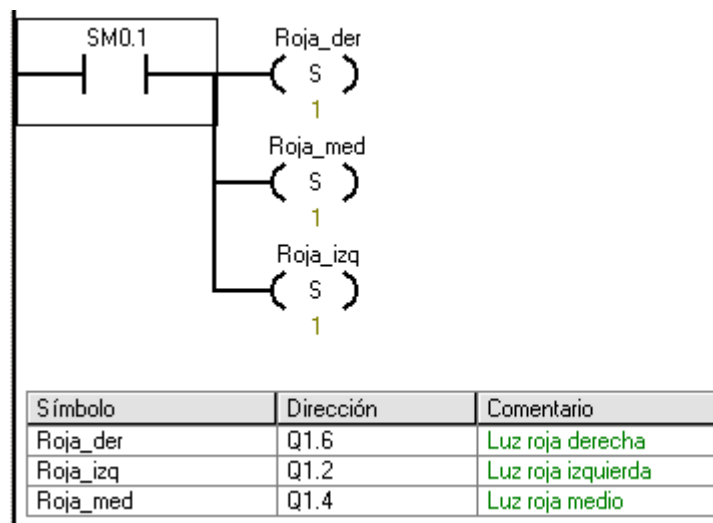


Figura 19. Inicialización del sistema automatizado.

Un usuario puede, en este momento, llegar con su auto para estacionar, o salir del parqueadero presionando el pulsador correspondiente.

Para el caso primero, el usuario, al llegar a la balanza del parqueadero, mediante un pulsador (I1.1) registra en el programa que se encuentra un nuevo auto a la espera de ser parqueado.

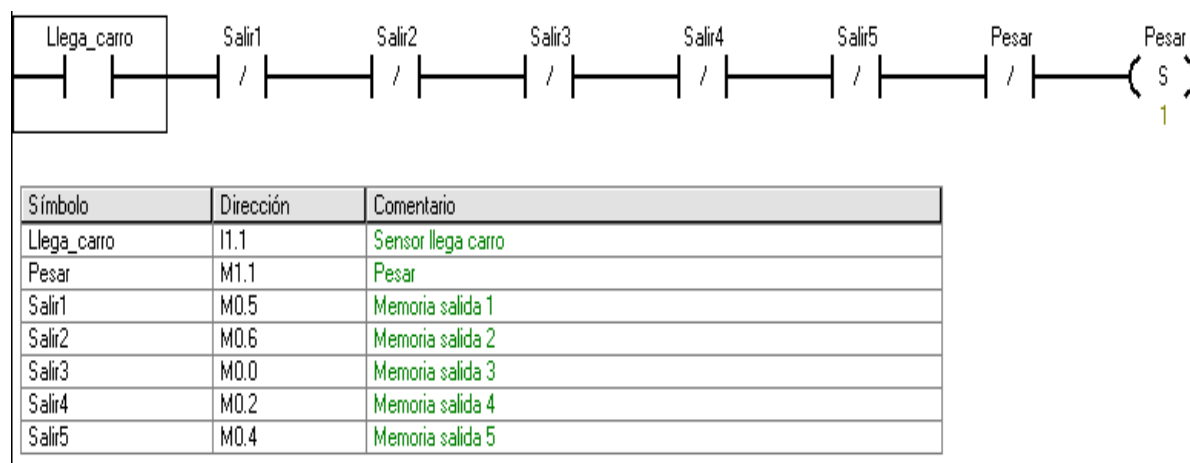


Figura 20. Llegada de un nuevo usuario

El programa entonces verifica que no exista ningún movimiento en el parqueadero, mediante los bits de memoria “Memoria Salir #”, y se inicia la función de pesar, mediante el bit (M1.1) llamado “Pesar”. Cuando este bit de memoria se activa, se compara en la balanza, que si el vehículo pasa de un peso establecido mediante el circuito comparador en el bit (I2.5),

es un vehículo pesado y se asigna el bit de memoria (M0.7) “Pesado”, caso contrario es un vehículo liviano y se asigna el bit (M1.0) “Liviano”.

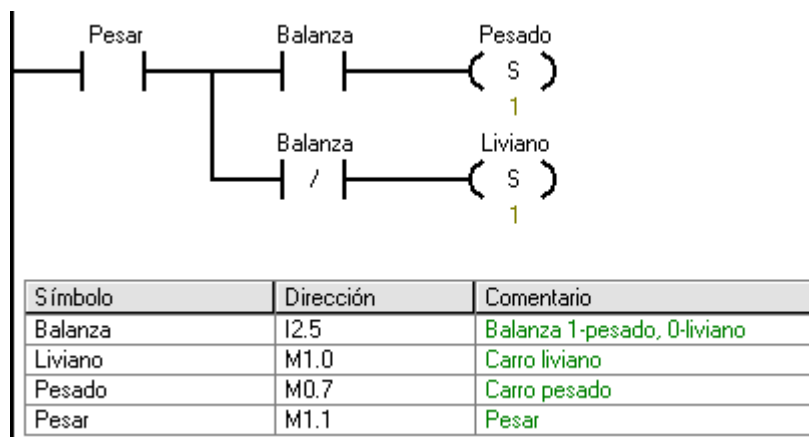


Figura 21. Pesar Carro Nuevo

Para el caso de que el vehículo es pesado, el programa empieza a verificar puestos libres en el parqueadero. La verificación se da en orden ascendente de acuerdo a la numeración de los puestos. Por lo tanto, primero verifica el puesto 1. Si la fotorresistencia del puesto 1 (I1.2) no detecta la presencia de un vehículo, el programa asigna al vehículo en espera el puesto 1, iniciando la subrutina del bit (M0.5) llamado “Memoria Salir 1”, además de eliminar los bits de memoria ya no necesarios, “Pesado” y “Pesar”.

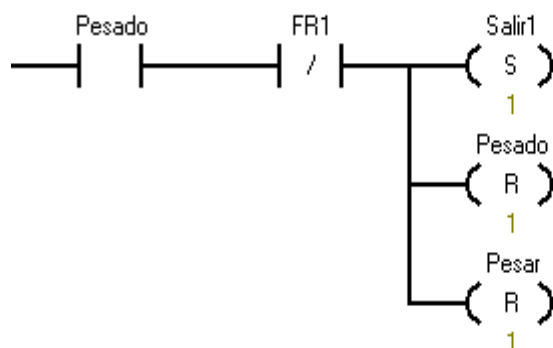


Figura 22. Verificando puesto 1.

En cambio, si se detecta un carro en el puesto 1, se procede a verificar el puesto 2 mediante la fotorresistencia (I1.3). Asimismo, si esta fotorresistencia no detecta presencia de vehículo, el programa asigna al vehículo al puesto 2, crea el bit (M0.6) “Memoria Salir 2” y elimina los otros bits explicados anteriormente.

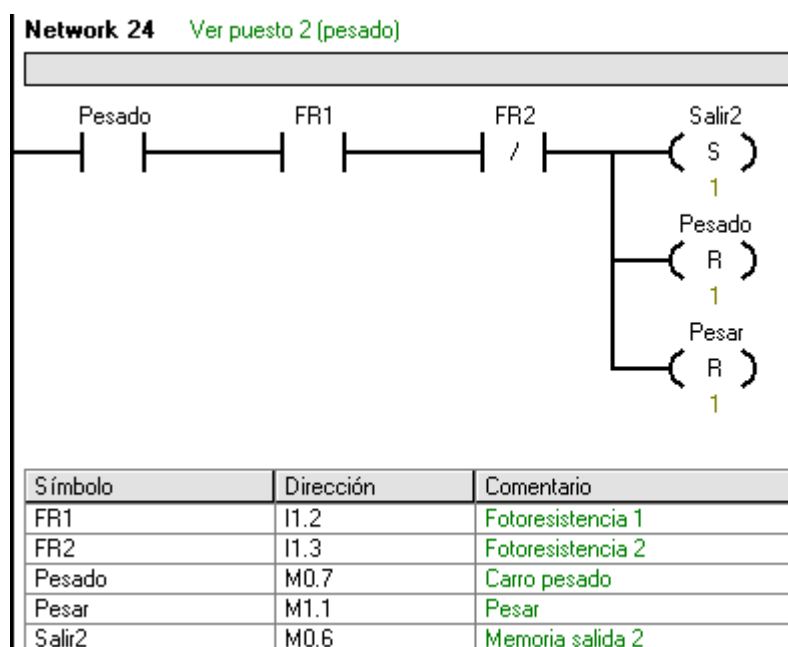


Figura 23. Verificación de puesto 2.

Finalmente, si está ocupado el puesto 2, el programa no asigna a ninguno, creando el bit (M1.2) “No hay puesto”, eliminando simplemente los bits de memoria creados para el vehículo, se reinicia el ciclo, y se espera una nueva tarea.

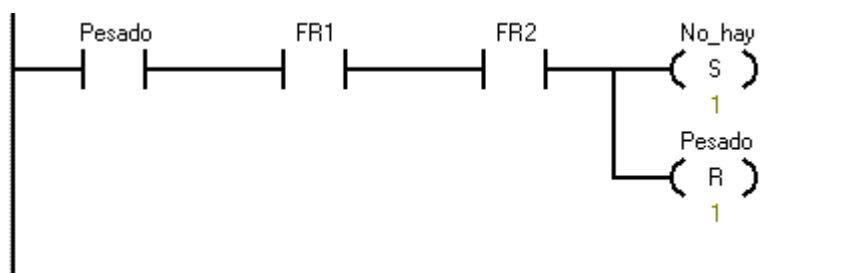
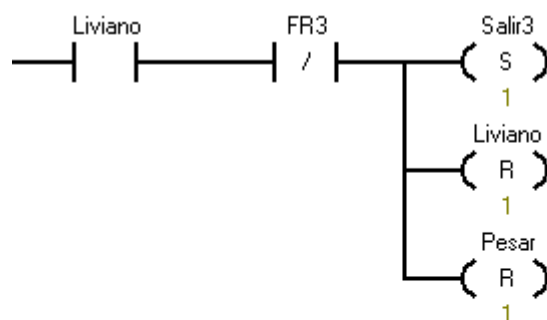


Figura 24. No hay puesto para carros pesados.

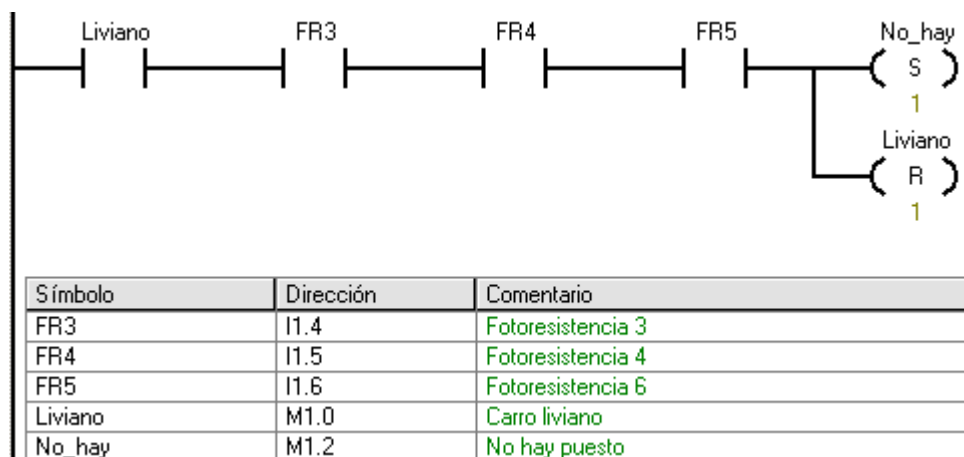
De la misma manera ocurre cuando el vehículo en espera es detectado como liviano. Se empieza por el puesto 3, y dependiendo de su fotorresistencia (I1.4), asigna o no este puesto, prendiendo y apagando los bits necesarios. Si no se asigna, continúa la verificación al puesto 4, y al puesto 5 si se diera el caso.



Símbolo	Dirección	Comentario
FR3	I1.4	Fotorresistencia 3
Liviano	M1.0	Carro liviano
Pesar	M1.1	Pesar
Salir3	M0.0	Memoria salida 3

Figura 25. Verificación puesto 3 carro liviano.

Si los todos los puestos superiores están ocupados, el programa no asigna a ningún puesto, e igualmente se crea el bit (M1.2) “No hay puesto”, que elimina los bits de memoria, y se reinicia el ciclo en espera de una nueva tarea.



Símbolo	Dirección	Comentario
FR3	I1.4	Fotorresistencia 3
FR4	I1.5	Fotorresistencia 4
FR5	I1.6	Fotorresistencia 6
Liviano	M1.0	Carro liviano
No_hay	M1.2	No hay puesto

Figura 26. No hay puesto disponible

Cuando se ha asignado un puesto a cualquier vehículo, sea liviano o pesado, se inician los procesos de preparación de la plataforma asignada para que quede lista para el ingreso de un vehículo. Para las plataformas inferiores, no es necesario moverlas del lugar en el que están en el momento necesitadas, ya que su acceso siempre es factible. Lo que se realiza es la determinación de su posición. La plataforma 1, solo puede estar en la posición izquierda o mitad. Por lo tanto, si el final de carrera de la izquierda (I0.0) detecta, significa que la plataforma se encuentra en la izquierda, sin importar que el fin de carrera de la mitad este o no activado. Por otro lado, si el fin de carrera derecho (I0.2) y el de la mitad (I0.1) están prendidos, significa que la plataforma 1 se encuentra en la posición de la mitad.

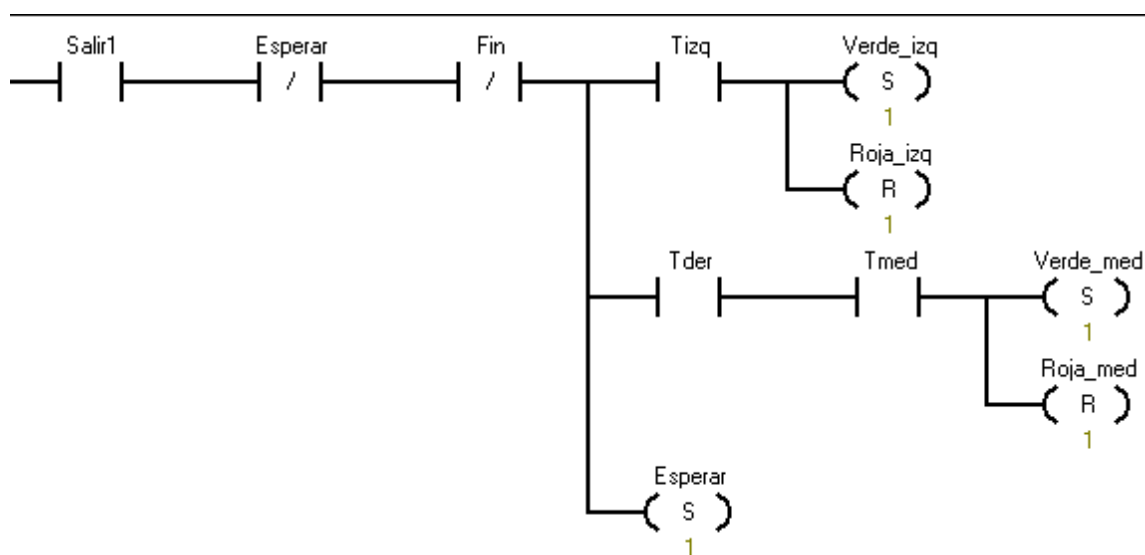


Figura 27. Subrutina para saber dónde se encuentra plataforma 1 y encender semáforo.

Este mismo análisis se realiza para determinar la posición de la plataforma 2, cuando es necesitada. Esta plataforma solo puede estar en la posición de la mitad o de la derecha. Por lo tanto, si el final de carrera de la derecha (I0.2) detecta, significa que la plataforma se encuentra en la derecha, sin importar que el fin de carrera de la mitad este o no activado. Por otro lado, si el fin de carrera de la izquierda (I0.0) y el de la mitad (I0.1) están prendidos, significa que la plataforma 2 se encuentra en la posición de la mitad.

Para las plataformas superiores, se necesita primeramente crear el espacio para el descenso de estas, consecuentemente, el proceso es secuencial y diferente para cada una. Por lo tanto, para la plataforma 3, primero se necesita que la plataforma 1 confirme su posición en la izquierda con el final de carrera izquierdo (I0.0). Después, se necesita que la plataforma 2 confirme su posición en la mitad con el final de carrera de la mitad (I0.1). Así entonces, con el espacio libre, puede iniciar el descenso de la plataforma 3 hasta que su final de carrera inferior (I0.7) detecte.

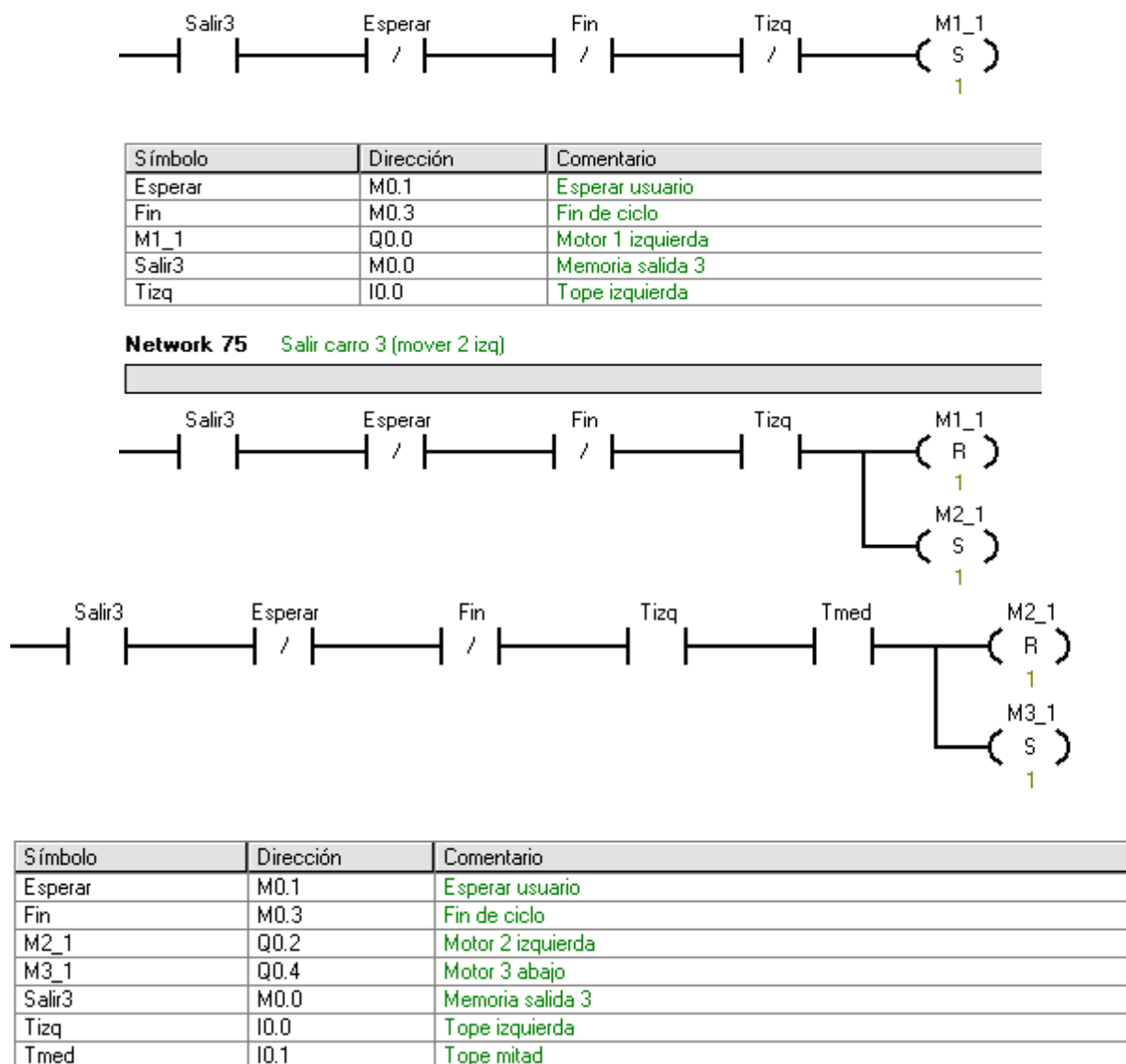
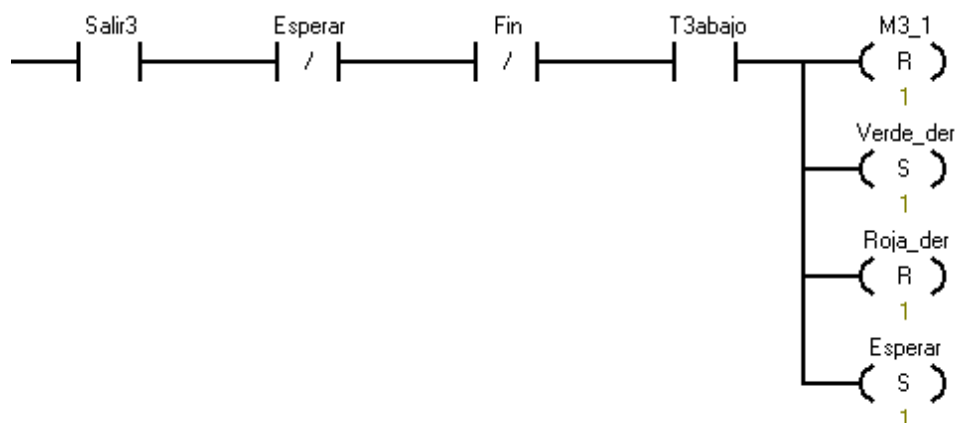


Figura 28. Creando espacio para que baje la plataforma 3

Ahora bien la misma lógica se lleva para el funcionamiento de las 5 plataformas y el encendido de los semáforos a la luz verde, anunciando que es permitido parquear o arrancar el carro para salir del estacionamiento.



Símbolo	Dirección	Comentario
Esperar	M0.1	Esperar usuario
Fin	M0.3	Fin de ciclo
M3_1	Q0.4	Motor 3 abajo
Roja_der	Q1.6	Luz roja derecha
Salir3	M0.0	Memoria salida 3
T3abajo	I0.7	Tope abajo puesto 3
Verde_der	Q1.7	Luz verde derecha

Figura 29 Encendido de semáforo.

El procedimiento para la plataforma 4 es análogo a lo descrito para la 3. Se necesita confirmar la plataforma 1 en la posición izquierda, y la plataforma 2 en la posición derecha, haciendo así posible que la plataforma 4 descienda hasta que su final de carrera inferior (I0.5) se active.

De la misma manera, para la plataforma 5, es necesario que primeramente la plataforma 2 confirme la posición izquierda y, a continuación, la plataforma 1 se mueva a la posición de la mitad. Solo así va a ser posible iniciar el descenso de la plataforma 5, hasta que su final de carrera inferior (I0.3) detecte.

Después de esto, cuando la plataforma necesaria esta lista para que el vehículo ingrese, se enciende la luz verde del semáforo de la posición respectiva, para que el conductor del vehículo proceda y parquee (figura 29). En este momento se crea el bit de memoria (M0.1) “Esperar usuario”, cuya función es que el programa no realice ninguna acción hasta que el conductor confirme que ha estacionado el vehículo.

Una vez parqueado, el conductor debe confirmar que el vehículo esta parqueado y que ninguna persona se ha quedado en él, mediante un pulsador “OK” (I1.7), el cual se encuentra afuera del parqueadero por medidas de seguridad.

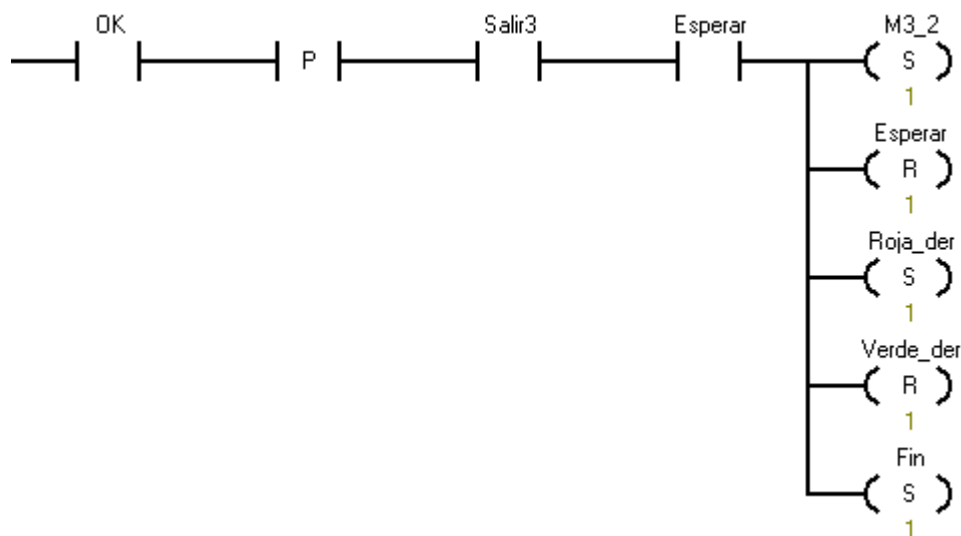
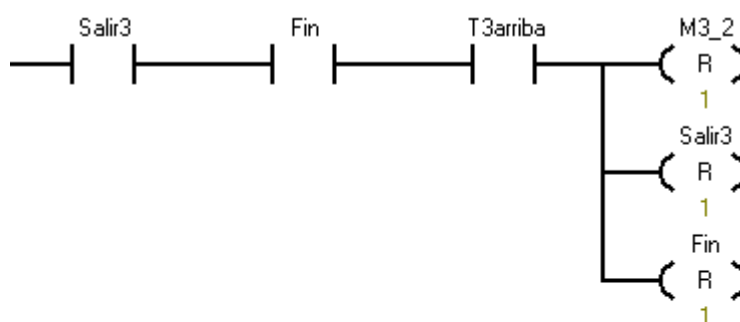


Figura 30. Regreso al puesto de la plataforma 3.

Cuando se ha dado la confirmación del conductor y el vehículo fue estacionado en el piso inferior (puestos 1 o 2), se enciende la luz roja del semáforo de la posición respectiva, el ciclo termina, apaga los bits de memoria usados (“Esperar usuario” y “Memoria Salida #”) e inicia uno nuevo en espera de una tarea.

Sin embargo, si el vehículo es liviano y se estaciono en las plataformas 3, 4 o 5, después de la confirmación existe un paso más, ya que estas plataformas deben regresar a su posición original, o sea, en la posición superior. Por lo tanto, después de la confirmación del

usuario, se apaga la luz verde, enciende la luz roja del semáforo, se apaga el bit “Esperar usuario” y se crea el bit de memoria (M0.3) “Fin de Ciclo”, el cual realiza la tarea señalada. Enciende el motor respectivo para que la plataforma usada llegue hasta su respectivo final de carrera superior. Cuando se ha confirmado esta posición, el ciclo termina, apaga los bits de memoria usados (“Fin de Ciclo” y “Memoria Salida #”) e inicia uno nuevo en espera de una tarea.



Símbolo	Dirección	Comentario
Fin	M0.3	Fin de ciclo
M3_2	Q0.5	Motor 3 arriba
Salir3	M0.0	Memoria salida 3
T3arriba	I1.0	Tope arriba puesto 3

Figura 31. Fin de ciclo plataforma 3.

Así entonces, el sistema puede recibir o despachar un vehículo.

Cuando un usuario necesita retirar su vehículo, existe un panel con pulsadores numerados de acuerdo al espacio asignado de su vehículo en el parqueadero (1 – 5). Por lo tanto, el usuario presiona el pulsador correspondiente para decir al programa que se quiere retirar el vehículo del puesto determinado.

Para iniciar la tarea de despacho de cualquier puesto, primero se verifica que no se esté moviendo ninguna plataforma del parqueadero, y a continuación, se crean las subrutinas de preparación de plataformas por medio de los bits de memoria llamados “Memoria Salir #”, ya

que se necesita cumplir los mismos procedimientos que cuando se recibe un vehículo, para dejar las plataformas en posiciones listas para poder retirar el vehículo.

Dependiendo el puesto necesitado, se activa el bit de memoria “Memoria Salir #”, se mueven las diferentes plataformas para que el carro quede listo para salir, como se explicó los diferentes procedimientos anteriormente.

Igualmente, cuando todas las plataformas están en la posición deseada, se prende la luz verde del espacio respectivo, para avisar al usuario que puede retirar el vehículo, y se crea el bit de memoria “Esperar Usuario”.

Finalmente, el conductor debe confirmar que el vehículo ha sido retirado mediante el pulsador “OK”, así terminar el ciclo de trabajo actual, iniciar un nuevo ciclo de trabajo, y continuar con el funcionamiento del parqueadero.

4.5. Variables a conectar como entradas

Como se puede observar el PLC consta de varias entradas las mismas que son utilizadas para el accionamiento de las salidas del PLC, es decir las entradas serán el punto de partido del funcionamiento del parqueadero. Hay que tomar en cuenta que el PLC consta de 24 entradas digitales en las cuales se conecta las diferentes variables que son necesarias para el correcto y lógico funcionamiento del parqueadero. Para el diseño de la maqueta se considera que se necesitan:

- 9 fines de carrera (topes)
- 7 pulsadores (1 ok , 1 nuevo carro, 5 para sacar el vehículo)
- 5 sensores de presencia (fotorresistencias acopladas al circuito con relé para hacer la señal digital.)
- 1 sensor de peso (balanza acoplada con un circuito con relé para hacer la señal digital)

Estas variables todas son digitales las mismas que para su conexión en el PLC se dispone de puntos comunes y de las entradas como tal. Todos los puntos común están unidos entre sí en la cual se deberá conectar un extremo del sensor (entrada) y el otro extremo se conectara a las entradas designadas con la letra I. tomar en cuenta que todas las entradas solo manejan con el termino cerrar el circuito es decir todas las señales digitales provenientes de las entradas son aisladas de voltaje. En otras palabras con los sensores estamos haciendo lo que manualmente seria colocar un cable desde el punto común del PLC hasta cualquier dirección de entrada todo automáticamente. Como se puede ver en ningún momento se trabaja con señales de voltaje ni de corriente.

4.6 Variable a conectar como salidas

Refiriéndose a la parte de salidas tenemos todos los dispositivos que actuaran dependiendo de las entradas activadas. El modulo del PLC consta con 16 salidas todas digitales tipo relé, las mismas que al seguir una secuencia dada son activadas dependiendo del estado de las entradas. Existe también 4 puntos comunes los mismos que están divididos de la siguiente manera: dos puntos comunes se encargan de 9 salidas y los otros dos puntos se encargan de las otras siete salidas. A diferencia de las entradas en las salidas ya se introduce señales de voltaje a los puntos comunes para que estos repartan la señal de voltaje a los distintos puntos designados como salidas las cuales se les reconoce fácilmente porque están marcadas con la letra Q. En el diseño del parqueadero se considera y se designa salidas a:

- 10 salidas de 7V (para controlar los “enables” de los motores, 2 “enables” por motor)
- 6 salidas 7V (Para el encendido de 3 luces verdes y 3 luces rojas).

La conexión de las salidas se realiza enviando la señal de 7V a los comunes de las salidas del PLC para que esta señal al activarse una salida se dirija a y sea enviada por las cualesquiera de las 16 salidas disponibles, esto dependerá de la activación de cada una de las salidas, como se puede ver no es necesario que cada salida se active por separado ya que esto dependerá de la programación y de la lógica que se maneja en el parqueadero.

4.7 Designación de direcciones

El cableado y conexión de los componentes electrónicos del parqueadero hacia el PLC fue estructurado de una manera ordenada para no confundirse ni perderse en ninguna conexión.

Las entradas y salidas usadas en el PLC se muestran a continuación:

Numero de Entrada	Descripción
I0.0	Tope izquierda
I0.1	Tope mitad
I0.2	Tope derecha
I0.3	Tope abajo puesto 5
I0.4	Tope arriba puesto 5
I0.5	Tope abajo puesto 4
I0.6	Tope arriba puesto 4
I0.7	Tope abajo puesto 3
I1.0	Tope arriba puesto 3
I1.1	Pulsador nuevo carro
I1.2	Fotorresistencia 1
I1.3	Fotorresistencia 2
I1.4	Fotorresistencia 3

I1.5	Fotorresistencia 4
I1.6	Fotorresistencia 5
I1.7	Pulsador Listo
I2.0	Pulsador 1
I2.1	Pulsador 2
I2.2	Pulsador 3
I2.3	Pulsador 4
I2.4	Pulsador 5
I2.5	Relé Balanza

Tabla 2. Designación de Entradas

Numero de Salida	Descripción
Q0.0	Motor 1 izquierda
Q0.1	Motor 1 derecha
Q0.2	Motor 2 izquierda
Q0.3	Motor 2 derecha
Q0.4	Motor 3 abajo
Q0.5	Motor 3 arriba
Q0.6	Motor 4 abajo
Q0.7	Motor 4 arriba
Q1.0	Motor 5 abajo
Q1.1	Motor 5 arriba
Q1.2	Luz roja izquierda
Q1.3	Luz verde izquierda

Q1.4	Luz roja medio
Q1.5	Luz verde medio
Q1.6	Luz roja derecha
Q1.7	Luz verde derecha

Tabla 3. Designación de Salidas

4.8 Diagramas de flujo rutina principal.

Una vez que se tiene la conexión de entradas y salidas se procede a realizar un diagrama de flujo el mismo que será programado en el PLC, para el diseño de este diagrama se realiza las secuencias que automáticamente el parqueadero debe realizar para su correcto funcionamiento. El parqueadero tiene dos etapas muy importantes y muy diferentes la primera es al momento de recibir un vehículo y la otra es al momento que va a salir el vehículo. Son las dos partes en las que se divide la programación.

Al momento de recibir el vehículo el usuario primero se encuentra con la balanza, la cual determina si el carro es pesado o liviano una vez establecido eso con un pulsador de nuevo carro el usuario da la señal de partida para que las plataformas reciban las distintas órdenes y comiencen a desplazarse. Primero para esto se realiza un historial para saber que puestos están ocupados como los puestos 3,4 y 5 son de preferencia para los vehículos livianos si es que el puesto 3 está ocupado se crea una sentencia para que vaya al puesto 4 y así sucesivamente, lo mismo sucede con los carros pesados los cuales los puestos son el 1 y 2. Una vez que se realiza la sentencia para saber que puesto está libre para el carro que ingreso proseguimos con otra sentencia la cual es la más crítica que son los movimientos de las plataformas. Para que una plataforma se mueva primero tiene que estar segura en qué posición esa encuentra y en qué posición están las demás solo así la plataforma deberá moverse. Este es

el principio el cual se utilizará para la programación básica del parqueadero considerando las varias condiciones para que tal o cual sentencia sean ejecutados.

El otro punto de la programación es al momento de sacar el vehículo el inicio de este proceso comienza con el pulsador para la salida los cuales son 5, ya que cada plataforma tiene su pulsador para su salida respectiva. De igual manera en este proceso para sacar el vehículo se necesita saber las posiciones de las plataformas para su respectivo desplazamiento. A continuación se muestra en diagrama de bloques el diagrama de flujo de las rutinas principales:

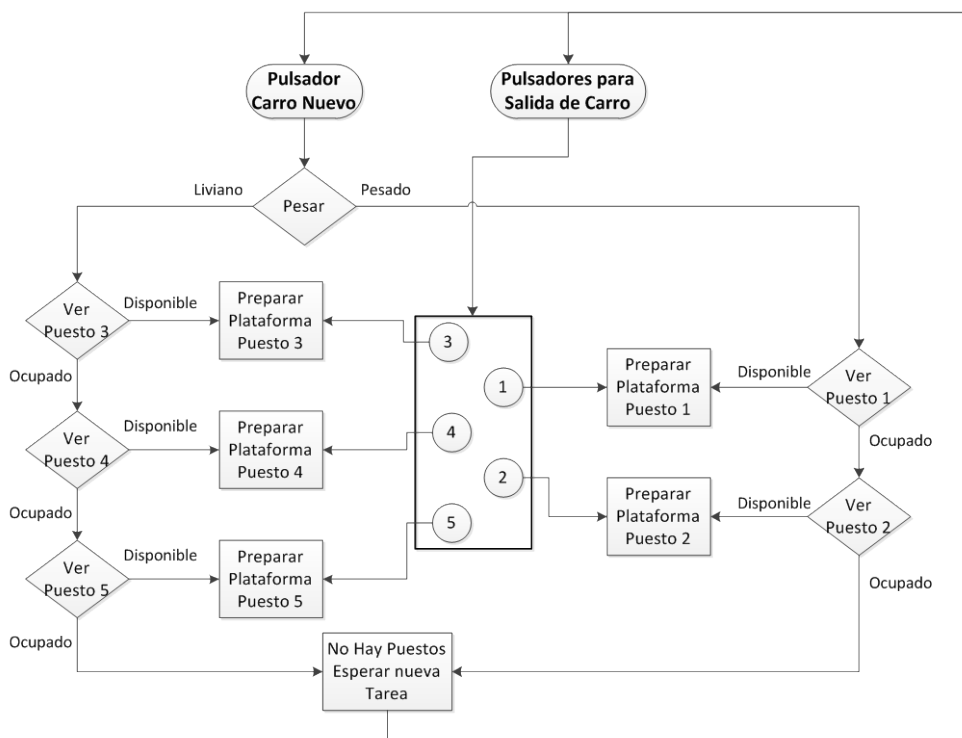


Figura 33. Diagrama de flujo rutina principal

4.9 Diagramas de flujo subrutinas diferentes plataformas

Una vez establecido una rutina principal en la programación se realizan unas subrutinas, estas serán las que dependiendo de las secuencias iniciales procederán a activarse.

Estas subrutinas en el flujo principales son las que dicen preparar plataforma 1, 2, 3, 4 y 5 como se puede ver hay un total de 5 subrutinas esto se debe a que a cada plataforma se le designó una subrutina por separado. También es importante recordar que las subrutinas son diferentes para los carros pesados y para los carros livianos además de esto, también se toma en cuenta las distintas posibilidades, que como secuencias se pueden dar, esto se explicará a continuación:

4.9.1 Descripción y diagrama de flujo plataforma 1

La plataforma 1 está destinada para la colocación de vehículos pesados. Esta plataforma tiene la opción de estar en dos posiciones las cuales se manejan de la siguiente manera: posición derecha, posición mitad, posición izquierda. Entonces continuando, la plataforma 1 sólo puede estar en la posición izquierda o en la posición mitad es decir se manejarán dos opciones.

En estas posiciones si es que están disponibles es decir sin vehículos, el parqueo del mismo es inmediato. En otras palabras no es necesario mover las otras plataformas ya que como se encuentran en el primer nivel del parqueadero el parqueo es instantáneo. Para facilidad del usuario se colocan semáforos los cuales dependiendo de la actividad a realizarse se enciende una luz verde o roja. Luz verde significando que el vehículo puede ser retirado o el vehículo puede ingresar y luz roja que principalmente indica que en esta plataforma no hay actividad. Así para esta plataforma se manejan dos semáforos los cuales dependiendo de la posición se encenderá la luz verde la cual indicará dónde se encuentra la plataforma ya sea en posición mitad o en posición izquierda. Esta luz verde va a representar realizar la acción de retirar el vehículo o puesto disponible para estacionar, establecida esta sentencia hay que esperar cual la acción del usuario la misma que será la activación del pulsador OK. Con este

accionar se dice al flujo de programa que el ciclo terminó y ahora tiene que esperar una nueva tarea.

Es importante aclarar que esta sentencia está destinada para el retiro de vehículos, ya que cuando se ingresa un vehículo la actividad acaba cuando la plataforma detecta la presencia de mismo, haciendo que se active el sensor y posteriormente el semáforo ubicado en esa posición de la plataforma cambiará de luz verde a luz roja, lo cual es el fin del ciclo porque de ahí el programa se queda en stand-by a la espera de otra tarea.

A continuación se muestra el diagrama de flujo de la plataforma 1 el cual sigue la secuencia de importancia y programación.

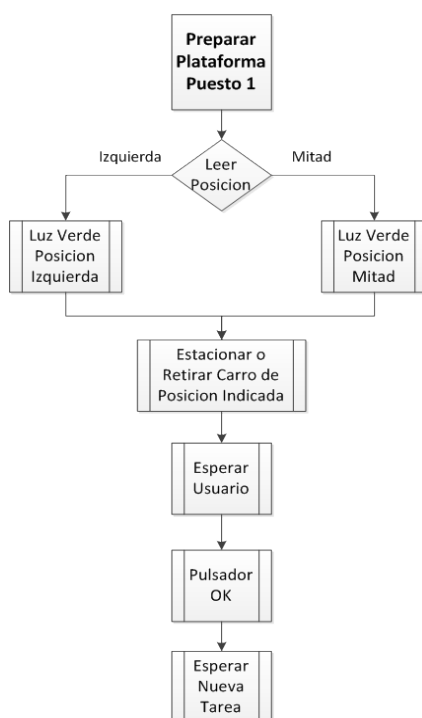


Figura 34. Diagrama de flujo plataforma 1.

4.9.2 Descripción y diagrama de flujo plataforma 2

La plataforma 2 está destinada para la colocación de vehículos pesados. Esta plataforma tiene la opción de estar en dos posiciones: en la posición derecha o en la posición

mitad es decir se manejarán dos opciones. En estas posiciones si es que están disponibles es decir sin vehículos, el parqueo del mismo es inmediato. En otras palabras no es necesario mover las otras plataformas ya que como se encuentran en el primer nivel del parqueadero el parqueo es instantáneo.

Para facilidad del usuario se colocan semáforos los cuales dependiendo de la actividad a realizarse se enciende una luz verde o roja. Luz verde significando que el vehículo puede ser retirado o el vehículo puede ingresar y luz roja que principalmente indica que en esta plataforma no hay actividad. Así, para esta plataforma se manejarán dos semáforos los cuales dependiendo de la posición se encenderá la luz verde la misma que indicará dónde se encuentra la plataforma ya sea en posición mitad o en posición derecha. Esta luz verde va a representar realizar la acción de retirar el vehículo o puesto disponible para estacionar. Establecida esta sentencia hay que esperar cual la acción del usuario la misma que será la activación del pulsador OK, con este accionar se dice al flujo de programa que el ciclo terminó y ahora tiene que esperar una nueva tarea.

Es importante aclarar que esta sentencia está destinada para el retiro de vehículos, ya que cuando se ingresa un vehículo la actividad acaba cuando la plataforma detecta la presencia de mismo, haciendo que se active el sensor y posteriormente el semáforo ubicado en esa posición de la plataforma cambiara de luz verde a luz roja, lo cual es el fin del ciclo porque de ahí el programa se queda en stand-by a la espera de otra tarea. A continuación se muestra el diagrama de flujo de la plataforma 2 el cual sigue la secuencia de importancia y programación.

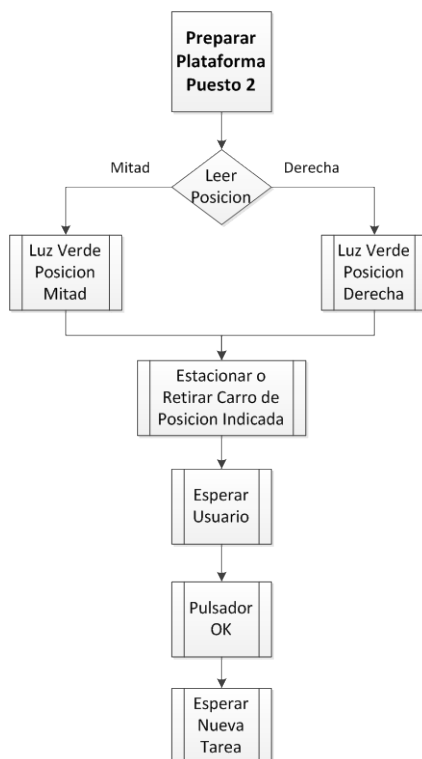


Figura 35. Diagrama de flujo plataforma 2

4.9.3 Descripción y diagrama de flujo plataforma 3

La plataforma 3 tiene un flujo de programación distintos a las plataformas 1 y 2 es que esta plataforma está designada a recibir vehículos livianos y al estar en el nivel 2 del parqueadero sus desplazamientos deben ser muy bien sincronizados ya que en la parte de abajo se encuentran vehículos. Entonces, para esta secuencia se toma en consideración primeramente las plataformas del nivel 1, obligatoriamente debemos saber las posiciones de las mismas ya que, así podremos tener un control de la plataforma 3 para realizar su desplazamiento hacia abajo. La plataforma 3 se encuentra en el nivel 2 posición derecha, una vez establecida la posición inicial se comprueba las posiciones de las plataformas 1 y 2. La plataforma 1 se la debe enviar obligatoriamente a la posición izquierda nivel 1, si es que ésta ya se encuentra ahí, se comprueba la posición de la plataforma 2, ésta se la ubicará en el nivel

1, posición mitad; una vez que estas dos plataformas estén en las posiciones correctas se envía la orden para que la plataforma 3 se desplace hacia abajo.

Luego que la plataforma está en el nivel 1 posición derecha el semáforo derecho de luz roja pasa a luz verde el cual nos indica que la plataforma está lista en su posición. Aquí se manejan dos opciones, ya que la rutina anterior descrita puede pasar al momento de sacar un vehículo o al momento de que llegue un nuevo vehículo, la única diferencia es su accionamiento puesto que con el pulsador nuevo carro en ON se dirá que se ejecute la rutina nuevo carro puesto 3 y con el pulsador 3, ubicado en el panel para retirar el vehículo, se accionará la rutina salida vehículo puesto 3.

Una vez que la plataforma está abajo hay que esperar al usuario para que dé la orden con el pulsador OK de que todo está listo para que la plataforma pueda regresar a su posición inicial. Esta orden del pulsador OK es la misma ya sea para nuevo vehículo puesto 3 o para salir vehículo puesto 3; en ambos casos hay que pulsar el botón OK, con esto hacemos que primeramente, el semáforo de la posición derecha cambie de luz verde a luz roja y la plataforma realice su asenso hacia su posición inicial. Luego de esto el ciclo de la plataforma 3 se acaba y el programa está listo para realizar una nueva tarea.

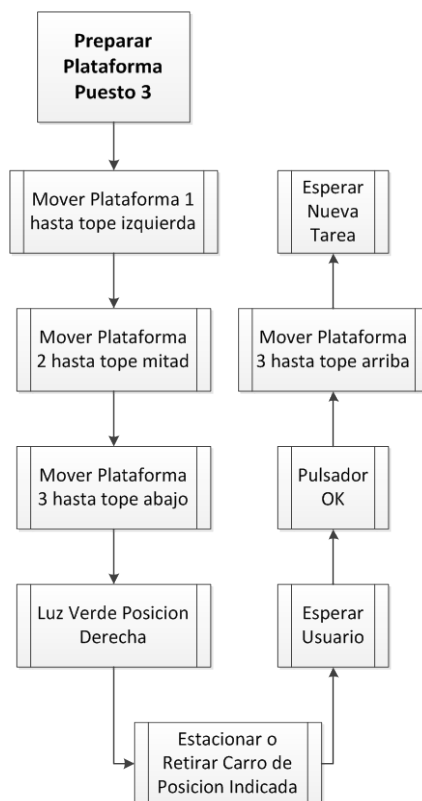


Figura 36. Diagrama de flujo plataforma 3

4.9.4 Descripción y diagrama de flujo plataforma 4

La plataforma 4 de igual manera que la plataforma 3 tienen un flujo de programación distintos a las plataformas 1 y 2, esto se debe a que esta plataforma está designada a recibir vehículos livianos y al estar en el nivel 2 del parqueadero la lógica se complica y los desplazamientos deben ser muy sincronizados, ya que en la parte de abajo se encuentran vehículos y errores causarían grandes daños. Entonces para esta secuencia se toma en consideración primeramente las plataformas del nivel 1, obligatoriamente debemos saber las posiciones de las mismas ya que, así podemos tener un control de la plataforma 4 para realizar su desplazamiento hacia abajo.

La plataforma 4 se encuentra en el nivel 2, posición mitad, una vez establecida la posición inicial se comprueba las posiciones de las plataformas 1 y 2. La plataforma 1 se la

debe enviar obligatoriamente a la posición izquierda nivel 1, si es que ésta ya se encuentra ahí, se comprueba la posición de la plataforma 2 , ésta se la ubicará en el nivel 1 posición derecha, una vez que estas dos plataformas estén en las posiciones correctas se envía la orden para que la plataforma 4 se desplace hacia abajo, una vez que la plataforma está en el nivel 1 posición mitad el semáforo mitad de luz roja pasa a luz verde el cual nos indica que la plataforma está lista en la posición requerida. Aquí hay dos opciones ya que la anterior rutina puede pasar al momento de sacar un vehículo o al momento de que llegue un nuevo vehículo, la única diferencia a las rutinas salida vehículo puesto 4 o entrada vehículo puesto 4 es el accionamiento.

Con el pulsador nuevo carro en ON se dirá que es la rutina nuevo carro puesto 4 y con el pulsador 4 ubicado en el panel para retirar los vehículos se accionará la rutina salida vehículo puesto 4. Una vez que la plataforma está abajo hay que esperar al usuario para que dé la orden con el pulsador OK y así todo está listo para que la plataforma pueda subir a su posición inicial. Esta orden del pulsador OK es la misma ya sea para nuevo vehículo puesto 4 o para salir vehículo puesto 4 en ambos casos hay que pulsar el botón OK, con esto hacemos que primeramente el semáforo de la posición mitad cambie de luz verde a luz roja y la plataforma realice su asenso hacia su posición inicial. Luego de esto el ciclo de la plataforma 4 se acaba y el programa está listo para realizar una nueva tarea.

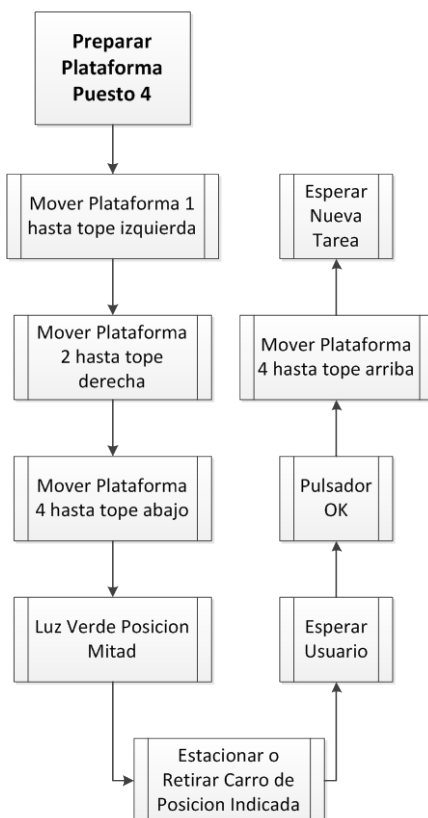


Figura 37. Diagrama de flujo plataforma 4

4.9.5 Descripción y diagrama de flujo plataforma 5

La plataforma 5 al igual que la plataforma 3 y 4 tienen un flujo de programación distintos a las plataformas 1 y 2, esto se debe a que esta plataforma está designada a recibir vehículos livianos y al estar en el nivel 2 del parqueadero la lógica se complica y los desplazamientos deben ser muy sincronizados, ya que en la parte de abajo se encuentran vehículos y errores causarían grandes daños. Entonces para esta secuencia se toma en consideración primeramente las plataformas del nivel 1, obligatoriamente debemos saber las posiciones de las mismas ya que, así podemos tener un control de la plataforma 5 para realizar su desplazamiento hacia abajo. La plataforma 5 se encuentra en el nivel 2 posición izquierda, una vez establecida la posición inicial se comprueba las posiciones de las plataformas 1 y 2.

La plataforma 1 se la debe enviar obligatoriamente a la posición mitad nivel 1, si es que ésta ya se encuentra ahí, se comprueba la posición de la plataforma 2 , ésta se la ubicará en el nivel 1 posición derecha, una vez que estas dos plataformas estén en las posiciones correctas se envía la orden para que la plataforma 54 se desplace hacia abajo, luego que la plataforma está en el nivel 1 posición mitad el semáforo izquierdo de luz roja pasa a luz verde el cual nos indica que la plataforma esta lista en la posición requerida. Aquí hay dos opciones, la anterior rutina puede pasar al momento de sacar un vehículo o al momento de que llegue un nuevo vehículo la única diferencia a las rutinas salida vehículo puesto 5 o entrada vehículo puesto 5 es el accionamiento.

Con el pulsador nuevo carro en ON se dirá que es la rutina nuevo carro puesto 5 y con el pulsador 5 ubicado en el panel para retirar los vehículos se accionará la rutina salida vehículo puesto 5. Una vez que la plataforma está abajo hay que esperar al usuario para que dé la orden con el pulsador OK y así todo está listo para que la plataforma pueda subir a su posición inicial. Esta orden del pulsador OK es la misma ya sea para nuevo vehículo puesto 5 o para salir vehículo puesto 5 en ambos casos hay que pulsar el botón OK, con esto hacemos que primeramente el semáforo de la posición izquierda cambie de luz verde a luz roja y la plataforma realice su asenso hacia su posición inicial. Luego de esto el ciclo de la plataforma 5 se acaba y el programa está listo para realizar una nueva tarea.

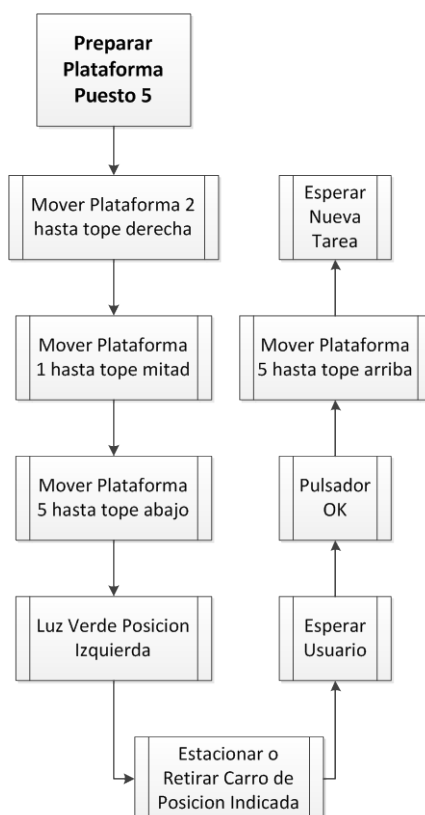


Figura 38. Diagrama de flujo plataforma 5

5. SISTEMA SCADA

5.1 Introducción SCADA

Una automatización eficiente y de alto nivel debe tener un sistema SCADA, ya que sin esto no habría un ente el cual pueda supervisar, controlar y administrar todo el proceso de la planta automáticamente. Esto comúnmente es realizado por el personal de una empresa, es decir, el proceso es manual, y lo que se busca en una automatización es reducir la intervención humana, sin quitarle la importancia al mismo, ganando con esto todas las ventajas de la automatización. Para esto hay que insertar un factor muy importante en el SCADA como son las redes de campo industriales las mismas que permiten desde un mismo sitio supervisar, controlar y adquirir las variables de una planta o de un proceso industrial. Estas redes se

diferencian de las redes administrativas de datos por el tipo de comunicación, por el protocolo y por los datos con los cuales operan estas redes.

En este proceso intervienen dos factores importantes: el primero es la parte de las redes física y el otro es la parte de visualización de interface. Para que estas dos herramientas fundamentales puedan tener un vínculo entre sí se requiere un protocolo de comunicación y para esto se necesita tanto de software como de hardware y todo constituye el sistema SCADA.

Hablando un poco sobre los antecedentes del SCADA podemos decir que ha tenido un avance muy significativo en lo que tiene que ver con el cableado, sensores, información, flexibilidad, re-configuraciones, simulaciones, almacenamientos, alarmas, monitoreo; en sí ha evolucionado tanto que ahora nos permite obtener y procesar información de procesos industriales dispersos y de lugares inaccesibles sin la necesidad de la intervención continua y rutinaria de un operador.

Para lograr este importante desarrollo existe la intervención de varios sistemas como: comunicación, unidades remotas, software de aplicación, etc. mediante esto los operadores tienen desde una PC una amplia visualización al proceso completo de la planta.

Es importante la instalación de un sistema SCADA en un proceso porque nos permite tener un control del historial de las alarmas y demás variables además también nos permite conocer el estado de las instalaciones; así también, no hay que olvidar que nos ayuda a coordinar eficazmente las labores de producción y mantenimiento en el campo, controlando y supervisando las diferentes operaciones críticas y proporcionando los recursos necesarios para recibir e enviar información en forma dinámica y en tiempo real.

La palabra SCADA viene de las siglas de supervisión control y adquisición de datos. Supervisión se refiere a representar en un monitor la evolución de las variables de proceso, la adquisición está destinando a recoger, procesar, adaptar, y almacenar la información recibida de los RTU (Unidad Terminal Remota) y el control se refiere a comandar o modificar la evolución del proceso.

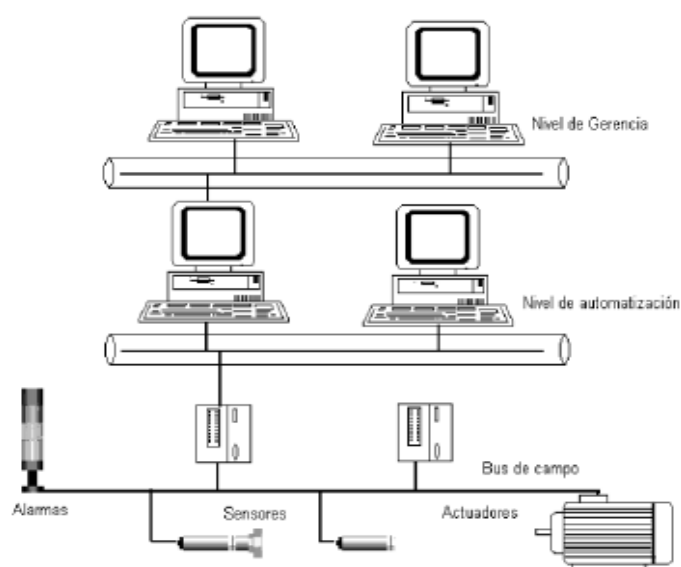


Figura 39. Niveles de Automatización Industrial.

Un proceso de automatización y SCADA es tan importante que puede significar sólo tener un nivel de gerencia para el control de toda la empresa, es decir toda la producción supervisión se confía en estas importantes herramientas que en la actualidad se están manejando. Es por esto que la importancia de un trabajo eficaz es indispensable, el correcto uso de sensores, el realizar una correcta comunicación, una correcta interface gráfica son detalles que hacen de una automatización el principal factor al desarrollo de una empresa.

El nivel que puede llegar un SCADA es tan complejo como queramos y ahí vienen los problemas asociados ya que existen un sinnúmero de caminos u opciones disponibles para ser

considerados como apropiados para tener un alto nivel de interface y lectura de datos. Uno de estos puntos son las comunicaciones industriales fiables, esto enrolla varios puntos importantes como el tipo de comunicación que se requiere entre el operador la PC y el RTU ya que en el medio existen no sola una opción sino varias como cables, fibra óptica, línea telefónica radio etc. Esto ya depende del uso del mismo. También hay que considerar el protocolo de comunicación entre los tantos que hay los más usados son Modbus, ASCII, la estructura de la comunicación como LAN y no hay que olvidar el software en los servidores como OPC server, como se puede ver tan sólo para la comunicación ya intervienen varios puntos a tomar en cuenta.

El software necesario para manejar el sistema SCADA es otro problema asociado ya que se tiene que manejar los datos e interpretarlos, a esto se llama la base de datos, además hay que crear una interface donde se muestre la información útil al usuario, para esto se puede usar (HMI,MMI). Estas son las consideraciones que hay que tomar en cuenta en el diseño de un sistema SCADA para que este sea interactivo, eficaz y la configuración dependerá de su uso y la importancia del mismo.

5.2 Conceptos Generales.

5.2.1 Misión de un sistema SCADA.

El sistema SCADA es realizado por medio de una estación central que puede ser un PC a lo cual se la denomina estación maestra o unidad terminal maestra (MTU); y una o varias unidades cercanas remotas (RTU) por medio de las cuales se hace el control, adquisición de datos hacia y desde el campo o lugar de donde está la unidad remota.

5.2.2. Flujo de información en los sistemas SCADA

Todo proceso el cual intervenga la automatización intervienen numerosas variables; éstas dependen del fenómeno físico que se observe como puede ser tope, luz, infrarrojo etc; al momento que estos fenómenos físicos son recibidos por un transductor, éste envía una señal eléctrica a un transmisor que normalmente son señales analógicas ya sea de voltaje o corriente normadas de 4 a 20 [mA], o 0 a 5 [V] DC, o desde 0 a 10 [V] DC. Normalmente estas señales eléctricas son convertidas en digitales para que luego sean enviadas a un cuarto de control donde se reúne la información de todo el proceso.

Al mismo tiempo la información es mostrada en una PC para que el personal a cargo pueda tomar decisiones; estos datos luego son digitalizados y al mismo tiempo almacenados para posteriormente realizar un análisis, logrando con esto generar históricos que luego servirán para tomar decisiones.

5.2.3 Adquisición de datos.

Se refiere a adquirir, procesar, y almacenar los datos para luego mostrar la información recibida en forma continua que se recibe de los equipos de campo.

5.2.4 Control.

Hay que ser bien claros en este punto ya que el SCADA no controla en sí el proceso, para eso está el PLC, se denomina control al hecho de controlar las variables que llegan del campo ya que con esto se puede realizar varios procesos subsiguientes como supervisiones, reportes, alertas etc.

5.2.5 Supervisión.

El encargado de la planta mediante el monitor tiene la opción de mirar la evolución de las variables de control, y los distintos cambios que se producen en la operación diaria de la planta o proceso.

5.2.6 Reportes.

Con los datos adquiridos es posible general todo la parte administrativa del proceso como: representaciones gráficas de los datos, predicciones, control estadístico, gestión de la producción, gestión administrativa y Financiera, etc.

5.3 Requisitos de un Sistema SCADA

Un sistema SCADA debe ser capaz de crecer o adaptarse según las necesidades cambiantes de la empresa o el proceso por ejemplo si se aumenta un sensor la interface debe ser tan eficaz que permita al usuario sin ninguna complejidad aumentar esa entrada a su visualización. La comunicación debe ser lo más fácil posible y transparente desde el usuario con el equipo de la planta y viceversa.

Los sistemas SCADA deben ser programas de software sencillos de instalar, sin muchos requerimientos de hardware, fáciles de utilizar y por supuesto con interfaces que sean muy ilustrativas al personal u operario encargado.

5.3.1 Hardware.

Para que un sistema SCADA opere con normalidad necesita ciertos componentes de hardware en su sistema como:

- Unidad terminal maestra (MTU).
- Unidad remota de telemetría (RTU).

- Red de comunicación.
- Instrumentación de campo.

5.3.1.1 Unidad terminal maestra (MTU).

La Unidad Terminal Maestra es el monitor o PC principal del sistema el cual supervisa y recoge la información del resto de las subestaciones, en éste se proyecta la información más importante del sistema. El sistema SCADA más sencillo está compuesto por un sólo computador ya que un sistema completo puede tener varios monitores controlando varios procesos.

5.3.1.2 Unidad Terminal Remota (RTU).

La Unidad Terminal Remota está destinada a recopilar los datos para luego ser transmitidos hacia la Unidad Terminal Maestra. En el caso de este proyecto es el PLC (Controlador Lógico Programable). Este sistema programable tiene entradas para la detección o medición de las variables de un destinado proceso y salidas para el control o activación de alarmas.

5.3.1.3 Red de comunicación.

La red de comunicación es una de las más importantes partes del sistema SCADA porque es la encargada de la transferencia de información entre la planta y el monitor que tiene a cargo la visualización de todo un proceso en tiempo real. El tipo de comunicación puede variar según las necesidades del sistema ya que la mayoría de equipos manejan distintos protocolos de comunicación lo cual hace que tanto el software como el equipo remoto sean compatibles manteniendo una misma comunicación ya que no todo los programas de software, así como los instrumentos de campo, pueden trabajar con el mismo medio de comunicación.

5.3.1.4 Instrumentación de campo.

Los instrumentos de campo constituyen todos los dispositivos que permiten realizar la automatización o control del sistema. Estos son el PLC, sensores, transductores, actuadores, etc.; ya que estos equipos son los destinados a la recepción de información del proceso.

5.3.2 Software

Los módulos o bloques de software que permiten las actividades de adquisición, supervisión y control son los siguientes:

- Configuración.
- Una Interface gráfica con el operador.
- Módulo de proceso.
- Gestión y archivo de datos.

5.3.2.1 Configuración.

Mediante esto el operador se encarga de definir el entorno de trabajo de su interface, ya que define las pantallas gráficas o de texto a utilizar.

5.3.2.2 Interfaz gráfico del operador.

Es la parte que permite al operador determinar el estado de los dispositivos de campo, ya sean estos en estados prendidos o apagados, de movimiento de activación etc. lo cual proporciona al operador las funciones de control y supervisión de la planta.

5.3.2.3 Módulo de proceso.

En este punto interviene programación la cual es necesaria para la ejecución distintos comandos o rutinas ya que estas están pre programadas a partir de valores actuales de las variables de campo obtenidas.

5.3.2.4 Gestión y archivo de datos.

Este punto se encarga del almacenamiento y proceso de los datos, según formatos inteligibles destinados a los periféricos tipo hardware como impresoras, registradores o de tipo software como reportes, bases de datos, hojas de cálculo, etc.

5.4 Introducción Indusoft

Indusoft es un Software HMI-SCADA basado en Web, tiene una amplia librería de objetos gráficos, alarmas, tendencias, reportes, acceso a base de datos Web Server, acceso remoto en tiempo real para monitoreo y acción.

La descripción principal del Indusoft es que es un conjunto de herramientas integradas, las cuales incluyen todo lo necesario para desarrollar modernas aplicaciones de Interface Hombre-Máquina (HMI), así como Sistemas de Adquisición de Datos (SCADA) que corren perfectamente en ambientes Windows NT, 2000, XP, Vista, CE, CE.NET, Internet e Intranet.

Entre las principales características tenemos:

- Tendencias en tiempo real e histórico.
- Alarmas y Eventos en tiempo real e histórico
- Administrador de recetas Envío de alarmas y eventos a e-mails y celulares
- Interface a bases de datos.

5.5 Protocolo Modbus

Esta designación corresponde a un protocolo de enlace (nivel OSI 2). Esto quiere decir entonces que se implementa una conexión física con diversos tipos, ya que cada fabricante comúnmente implementa un software de aplicación propia, de tal manera que estos productos entran dentro de un rango de parametrización [16]. La comunicación es serial punto a punto de manera que se pueda tener una mayor eficacia en la configuración y programación del sistema SCADA. Este protocolo Modbus permite mantener una comunicación entre los equipos de control, que en este caso serían el PLC y la computadora.

Este protocolo utiliza un bus de comunicación mediante el cable RS 232; hay que tener en cuenta que la comunicación se da a través del mismo cable con el que se programa el PLC (figura 39). Este tipo de comunicación asíncrona permite la comunicación entre el PLC y la interfaz gráfica del sistema SCADA, obviamente desarrollada mediante el software InduSoft. La estructura lógica es del tipo maestro-esclavo, el cual esta comandado por el maestro, es decir que todas las ordenes son enviadas por el maestro hacia el esclavo para que éste envíe la información solicitada hacia el maestro.

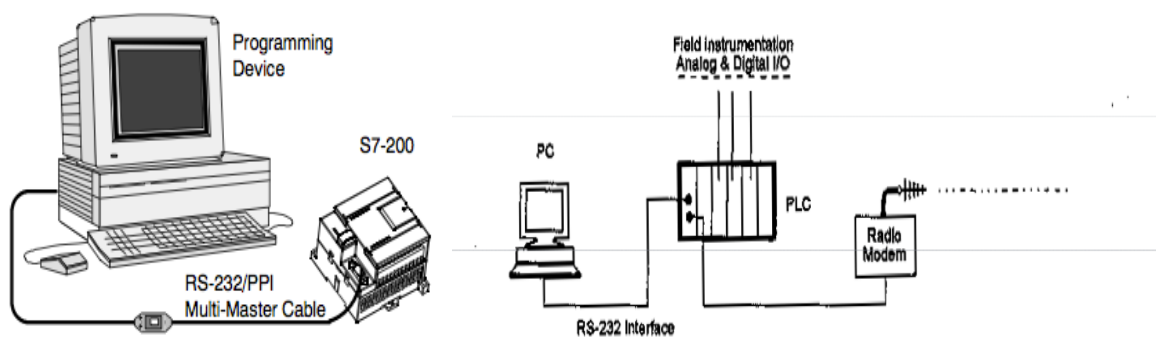


Figura 40. Modo del RS232/PPI cable multimaestro.

5.6 Designación de Tags

Los Tags son un componente esencial en cualquier proyecto IWS, porque son variables utilizadas por IWS para recibir y almacenar los datos obtenidos a partir de la comunicación con los dispositivos de la planta en este caso el PLC. A su vez, los tags se pueden utilizar para mostrar la información en las pantallas (y páginas web), para manipular objetos en la pantalla, y para controlar las tareas en tiempo real.

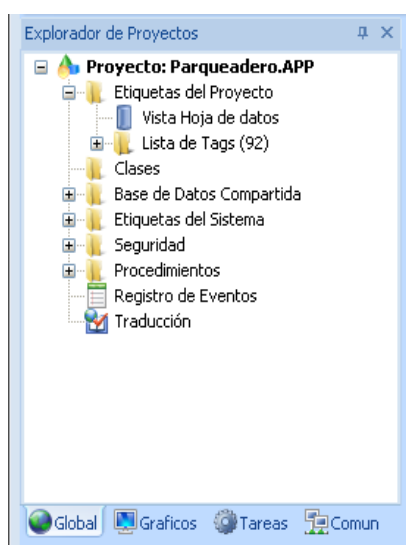


Figura 41. Tags

Para la configuración de los tags se busca la pantalla explorador de proyectos en la misma que encontramos el nombre del proyecto y una carpeta con el nombre de etiquetas de proyecto o (tags) como se puede observar en la figura 41:

Una vez abierta la carpeta Tags del proyecto nos aparece una pantalla mostrada en la figura 41. Como se puede observar existe varios puntos a tomar en cuenta para crear el tags entre estos están: nombre, el cual va a ser la identificación propia con el cual luego se podrá asignar alguna función específica, el arreglo en la cual se designa si tiene alguna subdivisión el tag, el tipo donde se indica de que tipo va a ser el tag, entre las opciones tenemos: booleano

(binario), entero, real y cadena; así mismo tenemos la opción descripción en la cual sólo se indica qué función va a cumplir ese tag o a que está asociado; y por último el alcance hay dos opciones: local y servidor, en nuestro proyecto como los datos a leer son del PLC, se lo modifica Servidor como se muestra en la figura.

	Nombre	Arreglo	Tipo	Descripcion	Alcance
15	☑ Tope_iz	0	Booleano		Servidor
16	☑ Tope_mitad	0	Booleano		Servidor
17	☑ Tope_der	0	Booleano		Servidor
18	☑ Tope_abajo_5	0	Booleano		Servidor
19	☑ Tope_arriba_5	0	Booleano		Servidor
20	☑ Tope_abajo_4	0	Booleano		Servidor
21	☑ Tope_arriba_4	0	Booleano		Servidor
22	☑ Tope_abajo_3	0	Booleano		Servidor
23	☑ Tope_arriba_3	0	Booleano		Servidor
24	☑ Pulsador_nuevo_carro	0	Booleano		Servidor
25		0	Booleano		Local
26	☑ Fotoresistencia2	0	Booleano		Servidor

Figura 42. Modelo de configuración Tags.

Cada dato a recibir estamos considerando como un tag es por esto que a cada variable que se va a leer del PLC se le asigna un tag, esto luego nos servirá para la interface gráfica del parqueadero. Se puede tener tantos tags como variables y los tags utilizados para el parqueadero se muestra en ANEXOS.

5.7 Programación PLC

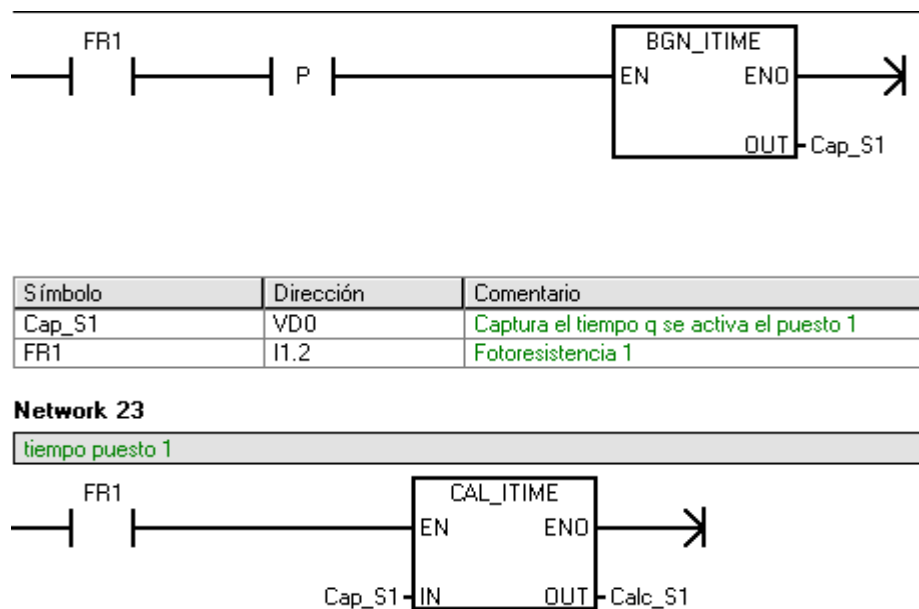


Figura 43. Tiempo total estacionado en la plataforma 1.

El momento de estacionar un vehículo en cualquier plataforma, digamos que en un caso se estaciona en la plataforma 1, entonces se activa un timer por medio de la fotorresistencia 1 (I1.2) que empieza a guardar como dato el tiempo de parqueo que dicho vehículo ha permanecido en el estacionamiento. De esta manera se lleva un control del tiempo total utilizado para estacionar por cada uno de los puestos. (Ver figura 43)

5.8 Programación Software

En este Punto se explicará que comandos se utilizaron para realizar el SCADA y sobre como como se pudo realizar la interfaz gráfica del parqueadero inteligente.

5.8.1 Descripción de las variables

Primeramente, antes de comenzar a programar el Software hay que saber a dónde queremos llegar y qué queremos mostrar en la interfaz gráfica. Se dividen las variables y las actividades en 3 grandes grupos que son:

- ❖ Interfaz gráfica para el operador que permita visualizar:

- Ocupación de los diferentes puestos de parqueadero
 - Movimientos de las plataformas del parqueadero
 - Estado de detección de los sensores finales de carrera
 - Identificación del vehículo como pesado o liviano
 - estado de los semáforos, pulsadores, motores.
- ❖ Herramienta que genere automáticamente reportes diarios, escritos indicando:
 - Ocupación de cada puesto de parqueadero y del conjunto
 - Cantidad de vehículos livianos y pesados por día
 - Tiempo promedio para estacionarse
 - ❖ base de datos de alarmas, considerando las siguientes:
 - Parqueadero Lleno
 - Demora anormal para estacionarse
 - Demora anormal para salir del parqueadero

5.8.2 Métodos de programación y comandos utilizados

Para la programación de la interface se utilizará los comandos disponibles en el software Indusoft. La programación es booleana mediante tags los mismos que tienen que ser creados antes de ser utilizados. En la herramienta de gráficos se encontrarán varias opciones para el diseño de la interfaz gráfica, la misma que por la complejidad de la estructura del parqueadero utiliza herramientas muy simples para formar una interfaz fácil de entender y que cuente con todos los beneficios para la visibilidad del operador.

De esta manera primero se procede a realizar las plataformas las cuales para hacerlas más vistosas se las realizó con proyecciones 3D, para realizar esto se partió de líneas lo cual es el elemento principal de las plataformas. De la misma forma se realizó un prototipo de semáforo para que el operador también tenga la oportunidad de observar cada detalle de la semaforización. De la librería Symbols de Indusoft se procede a utilizar objetos que nos servirán de mucho para la representación de nuestras variables instaladas en el parqueadero. Como por ejemplo: carros, motores, flechas y así mismo tags que nos proporciona la hora y fecha actual la misma que está sincronizada con el timer de la computadora.

5.8.3 Efectos de Movimiento

Para lograr los efectos de movimiento los cuales son aplicadas en las plataformas vacías, plataformas con vehículos y en sí todos los detalles añadidos a las plataformas que se moverán, se utiliza la librería animaciones la cual nos permite realizar varios efectos posibles y la complejidad dependerá de la programación que se realice en cada ventana existente en cada comando. El comando utilizado para lograr los efectos de movimiento es **Posición**:

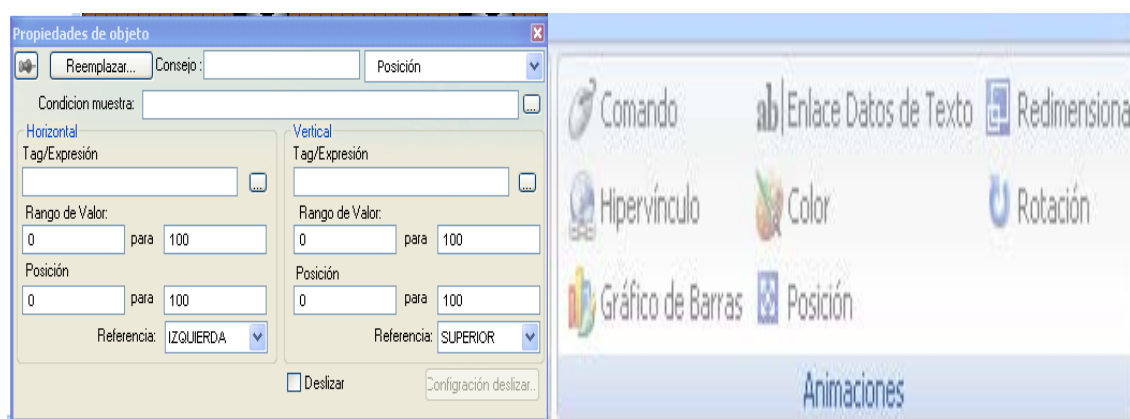


Figura 44. Propiedades de Objeto.

El comando posición lo que hace es darle vida al objeto, variando sus características iniciales para esto se selecciona el objeto y se le da click en posición una vez realizado esto se da doble click en el objeto permitiendo que así se abra la ventana mostrada en la figura (45). Hay varios parámetros a configurar en la pantalla “propiedades de objeto”. Existe la opción de mover al objeto en forma horizontal y vertical, dependiendo de esto se configura las coordenadas que son la posición y el rango de valor que es la escala. La programación se ubicará en la parte de tag/Expresión. Este punto es el más importante del movimiento ya que aquí se encuentran las funciones existentes para dicha programación. Haciendo un click en el recuadro pequeño de la derecha aparecen las funciones que se pueden utilizar en esta programación. Ver figura 45:

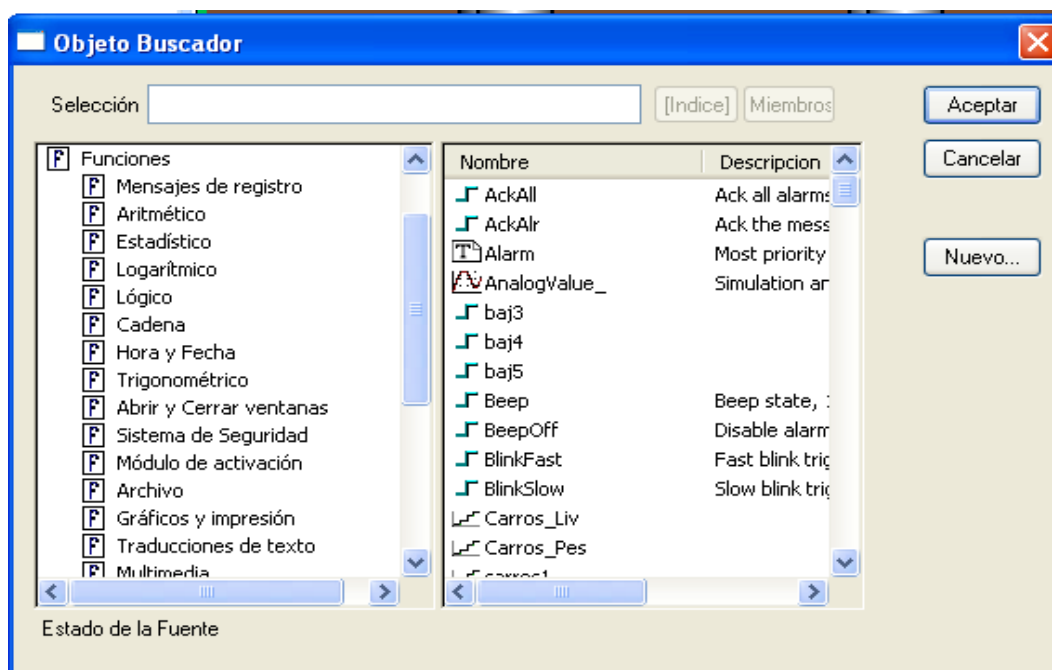


Figura 45. Etiquetado de las Variables

Luego de escoger las funciones necesarias el programa le dirá si es que la sentencia está bien escrita o no, para la ejecución del tags. Se puede realizar varias conjugaciones entre ellas para lograr efectos que a simple vista no hubieran sido posibles.

Para el movimiento de las plataformas se plantean 3 estados, estos son: arriba, mitad o abajo, para las plataformas de arriba o izquierda mitad derecha para las plataformas de abajo.

Código utilizado para lograr movimiento de tres estados:

```
“if (Tope_arriba_4 and toggle(Tope_abajo_4),0, if (toggle(Tope_arriba_4) and
Tope_abajo_4,1, 0.5))*100”
```

5.8.4 Descripción del programa

F	If(numCondition , nu...	Ejecución condicional
F	Toggle(numValue)	Devuelve el valor alternado.

Figura 46. Formato de Programación.

- And es un operador lógico
- Tope_arriba_4 tag de la posición de arriba del puesto 4
- Tope_arriba_4 tag de la posición de abajo del puesto 4
- *100 es para aumentar la escala.

Así se logra un movimiento de 3 estados. Lo cual fue aplicado a todos los objetos que se considera que van a realizar desplazamientos tanto en eje x como en eje y.

5.8.5 Cambio de colores.

Esta animación se la utiliza en varios objetos los cuales mediante tag, y las variables leídas al PLC se puede obtener cambios de colores en los objetos, como la lógica es booleana, es decir, binaria sólo podemos tener dos cambios de colores y para la interface del parqueadero se utilizará colores estándar los cuales son rojo para apagado y verde para encendido la descripción de los objetos a los cuales se les aplica esta animación se describe a continuación:

Objeto	estado	color
Motores	encendido	luz verde
	apagado	luz roja
Semaforos	encendido	luz verde
	apagado	luz roja
barras o tornillos sin fin	en movimiento	luz verde
	sin movimiento	luz roja
fines de carrera	activado	luz verde
	desactivado	luz roja
pulsadores	activado	luz verde
	desactivado	luz roja
flechas	entrando vehiculo	luz verde
	saliendo vehiculo	luz roja
	subiendo , bajando plataforma	luz amarilla

Tabla 4.Estado de las variables a controlar.

Para la configuración de la animación de cambio de color se utiliza el comando color. Para activar este comando se selecciona el objeto y luego se da click en el comando color para así cambiar sus propiedades para el cual hay que dar doble click sobre el objeto.Y se abrirá la pantalla mostrada en la figura 47

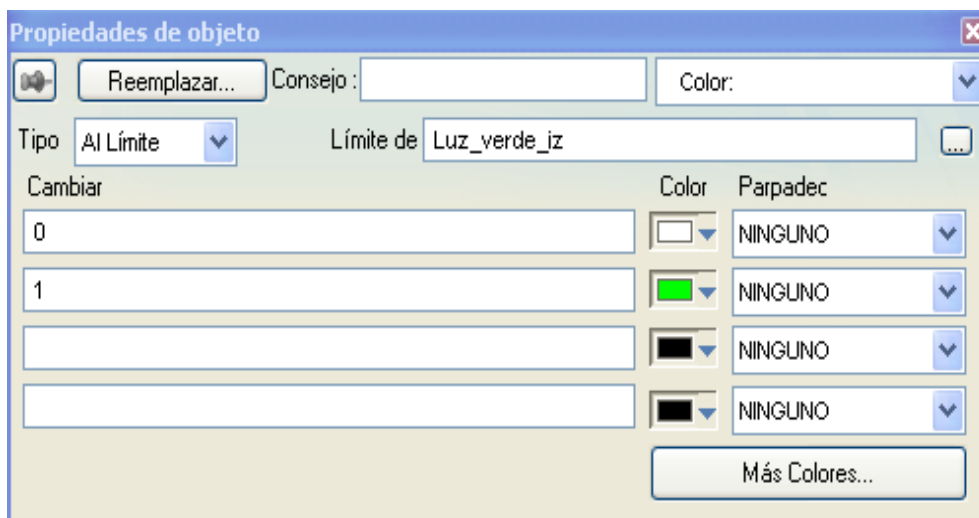


Figura 47. Configuración para cambiar de color.

En la esquina arriba derecha se encuentra las opciones o animaciones que están cargadas en el objeto, hay que seleccionar la opción color si es que el objeto tiene varias animaciones. Un ejemplo se muestra en la figura 48:

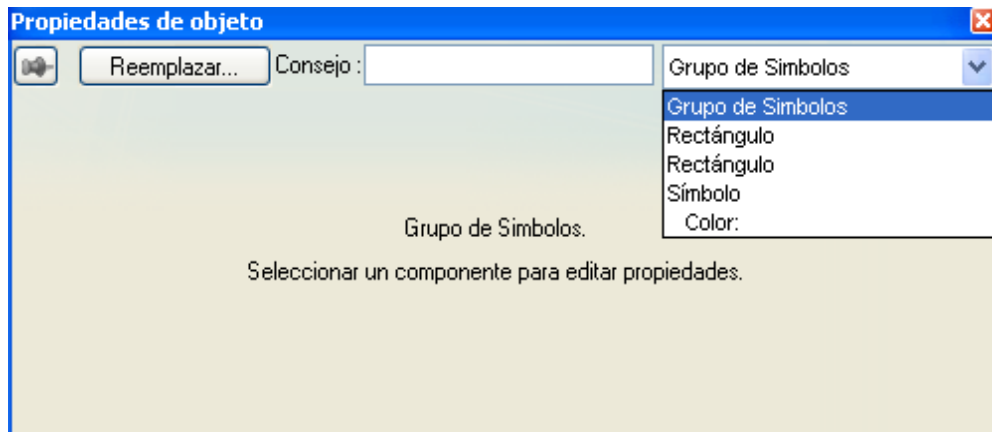


Figura 48. Propiedades de animaciones.

Para la configuración de la propiedad color se utiliza los estados del tag los cuales son 0 o 1 y solamente se designa el color que se requiera y lo más importante indicar cuál es el tag que representará a ese objeto.

5.8.6 Aparición y Desaparición de Objetos

Esta animación se utiliza para realizar los efectos de presencia del vehículo en la plataforma. Para esto se utiliza el comando modificar el cual se encuentra en la barra de animaciones. Para la configuración de este efecto vamos hacer que el vehículo siempre esté en la plataforma y con el comando modificar a este lo multiplicamos ya sea por 0 para desaparecer o por 1 si es que se necesita aparecer el vehículo, esta es la lógica que se maneja para esta animación. La tabla de configuración del objeto a desaparecer se muestra en la figura 49:



Figura 49. Aparición de objetos.

5.8.7 Alarmas

Las alarmas son configuradas en el mismo software Indusoft , es decir se utiliza las alarmas que genera el mismo programa con nuestras configuraciones y condiciones. Para dicha configuración se utiliza en el menú tareas la opción alarmas y mediante eso se abre la pantalla que se muestra en la figura 50.

	Nombre de Tag	Tipo	Limit	Mensaje	Prioridad	Selección
1	stop	Hi	1.000000	Stop process	0	
2	continue	Hi	1.000000	Reanudación exitosa	0	
3	lleno	Hi	1.000000	Parqueadero LLeno	0	
4	t_estacionarse	HiHi	45000.0...	Demora anormal para est...	0	
5	demora_salida1	HiHi	45000.0...	Demora salida puesto 1	0	
6	demora_salida3	HiHi	45000.0...	Demora salida puesto 3	0	

Figura 50. Configuración alarmas.

Las alarmas de igual forma que todos los comandos implementados en el Indusoft, funcionan mediante tags, es así que se pueden obtener variables binarias, enteras, es así que mediante la condición que se le ponga , la alarma se ejecuta automáticamente. La condición es el cambio de valor del tags asociado a esa alarma. Las alarmas son muy importantes en la implementación del SCADA ya que nos permiten hacer históricos o prevenir accidentes mediante señales de que algo no está bien en la planta. Así el operador puede tomar decisiones tan solo viendo la interface del computador.

Dependiendo del nivel de la alarma en algunos casos es necesario programar en la pantalla, sub comandos para que el resultado de la operación programada sea la alarma. Un caso utilizado en este parqueadero es la utilización de la alarma ‘parqueadero lleno’ el cual se programa de la siguiente manera:

```
``Este procedimiento es ejecutado continuamente mientras esta pantalla está abierta.
```

```
SubScreen_WhileOpen()
```

```
If $Fotoresistencia1 And $Fotoresistencia2 And $Fotoresistencia3 And $Fotoresistencia4 And
    $Fotoresistencia5=1 Then
    $lleno=1
    Else
    $lleno=0
    EndIf``
```

5.8.8 Reportes

Para la sección de reportes se manejan dos opciones: el reporte automático, el cual consiste en definir una hora establecida para la generación del mismo, o la segunda opción que se define como el reporte inmediato , el cual mediante un pulsador ya sea el operador genera el reporte inmediatamente.

```
``Este procedimiento es ejecutado continuamente mientras esta pantalla está abierta.
```

```
SubScreen_WhileOpen()
```

```

    If $Hour=$In_horaThen
    If $Minute=$in_minutoThen
    If $Second=$in_segundoThen
    $Report( "Disk:Report1.rep" )

        EndIf
        EndIf
        EndIf

    EndSub``

```

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

El objetivo final del proyecto consiste en fabricar un modelo a escala de un sistema de parqueadero elevador automático con su respectivo sistema SCADA. Pero para conseguir este objetivo, primero se logró obtener los siguientes objetivos específicos.

- Se fabricó el modelo a escala con 5 puestos en 2 niveles.
- Se implementó el sistema que permita movilizar los vehículos verticalmente y horizontalmente.
- Se instaló sensores que permitan determinar la ocupación de cada puesto.
- Se calibró un sensor de peso que permita distinguir entre vehículos livianos y pesados.
- A implementación de un sistema de semáforos a la entrada para determinar si existe o no puestos libres.
- Programación del funcionamiento automático mediante un PLC.
- Se puso a punto el modelo a escala de parqueadero automático, implementar un sistema SCADA de comunicación y monitoreo del estado del sistema.

- Se configuró la comunicación entre una computadora fija y un PLC Siemens S7-200, también la comunicación desde el software Indusoft Web Studio hacia el equipo de control
- Para una mejor apreciación se diseñó una interface gráfica para el operador que permita visualizar
 - Ocupación de los diferentes puestos de parqueadero
 - Movimientos de las plataformas del parqueadero
 - Estado de detección de los sensores finales de carrera
 - Identificación del vehículo como pesado o liviano
- También se creó una herramienta que genere automáticamente reportes diarios, semanales, escritos indicando:
 - Ocupación de cada puesto de parqueadero y del conjunto
 - Tiempo promedio de ocupación de parqueadero
 - Cantidad de vehículos livianos y pesados por día
 - Número de horas de encendido de los motores
- Y por último se diseñó una base de datos de alarmas, considerando las siguientes:
 - Parqueadero Lleno
 - Demora anormal para estacionarse
 - Demora anormal para salir del parqueadero

Se logró el control total automatizado mediante la programación en el MicroWin, que es el software que trabaja en conjunto con el PLC Siemens. Este tipo de PLC's son muy utilizados en aplicaciones industriales, gracias a su diseño compacto y capacidad de ampliar

sumemoria, mucho más en automatizaciones específicas, que no requieran de un nivel avanzado de conocimientos.

La comunicación entre el PLC y el Indusoft, el cual es utilizado para la programación e interfaz gráfica del sistema SCADA, fue directa de acuerdo a los parámetros del protocolo utilizado, que en este caso es el Modbus y con el cable PC/PPI que ya viene diseñado para este tipo de PLC's donde se puede configurar directamente la velocidad de comunicación a la que se quiere transmitir los datos dentro de la automatización, ya sea para la adquisición de datos o el control del parqueadero.

Se realizó un efectivo comando manual de cada una de las plataformas, en el caso de que se requiera mantenimiento de las mismas, o por alguna otra emergencia en la cual se necesite mover determinada plataforma a una cierta distancia; esto es más para el caso de seguridad que por algún motivo en particular pueda suceder algún daño en la automatización y así poder mover cada plataforma manualmente, para que los vehículos que se encuentren dentro del parqueadero puedan salir.

En conclusión, el parqueadero automatizado es capaz de estacionar 5 vehículos en donde normalmente caben 3, esto quiere decir que este modelo a escala optimiza un espacio del 40% y por lo tanto es un proyecto factible que se puede aplicar en la realidad. Hay que tener en cuenta que tanto la estructura mecánica como la electrónica, deben estar perfectamente configuradas para que no exista ningún contratiempo posteriormente y así mantener alineadas y controladas cada una de las plataformas, ya sea en sus motores o en los diferentes sensores que contienen cada una de ellas.

Las interfaces gráficas del sistema SCADA ayudan al operador a visualizar de una mejor manera todo el funcionamiento y la automatización del parqueadero, así como también

sus diferentes alarma y reportes que se crean a diario, para un control eficiente del parqueadero. Hoy en día los sistemas SCADA son muy utilizados en el campo de la industria ya que de esta manera se puede llevar un control total y absoluto de la producción, así como también advertir de ciertas fallas que se puedan ocasionar en diferentes puntos de la planta en la cual se encuentre automatizada.

6.2 Recomendaciones

La utilización de guías para cada una de las plataformas y sensores es muy útil, puesto que brindan seguridad a la dinámica del sistema evitando que la estructura móvil se salga de su camino, y así teniendo una estabilidad completa de la estructura mecánica.

Como se utilizaron todas las entradas y todas las salidas digitales del PLC, es muy importante que se lleve un etiquetado de todo el cableado estructurado, ya que de esa manera se tendrá una mayor facilidad al momento de realizar las conexiones hacia el PLC y así poder actualizar o hacer algún tipo de mantenimiento al mismo, de esta manera se optimizará el tiempo.

Como se sabe, los sensores de presencia de cada una de las plataformas dependen de la cantidad de luz que pueden captar, por lo tanto antes de poner en funcionamiento el parqueadero, se necesita de una calibración de los mismos, de tal manera que de acuerdo al cuarto donde se encuentre y la luz que rodee ese cuarto pueda ser captada por los sensores para así tener una lectura correcta de la cantidad de luz que existe cuando una plataforma está ocupada con un vehículo y cuando está desocupada.

La comunicación entre el PLC y la computadora es muy importante dentro de la automatización ya que constantemente se están enviando y recibiendo datos de las diferentes variables importantes que se manejan. Razón por la cual se necesita entender el tipo de

protocolo que se está utilizando, que en este caso es Modbus, y también conocer la forma o la dirección del PLC a utilizar ya que sin este dato no se puede realizar una comunicación exitosa, esto se lo realiza dentro de la programación en Ladder o MicroWin, en un bloque llamado comunicación. Dicho sea de paso, los PLC Siemens son fáciles de implementar la comunicación.

REFERENCIAS

- ✓ [1] SIEMENS. (2000). *Manual de sistema de Automatización S7-200*. Milan: De Lorenzo.
- ✓ [2] Stenerson, J. (2003). *Industrial Automation and Process Control*. New Jersey : Prentice Hall.
- ✓ [3] David Bailey, Edwin Wright. (2003). *Practical SCADA for industry*. Gran Bretaña: Elsevier.
- ✓ [4] Grodon Clarke, Deon Reynders. (2004). *Practical Modern SCADA Protocols*. Gran Bretaña: Elsevier.
- ✓ [5] Dagoberto Montero, David Barrantes, Jorge Qirós. (1 de junio de 2004). *Scribd*. Recuperado el 3 de Mayo de 2013, de Introducción a los Sistemas SACADA: <http://es.scribd.com/doc/13473499/Introduccion-a-Los-Sistemas-SCADA>
- ✓ [6] Machaca, R. (17 de Diciembre de 2007). *Slideshare*. Recuperado el 07 de Mayo de 2013, de Circuitos Integrados: <http://www.slideshare.net/noche/circuitos-integrados-206123>
- ✓ [7] Ruiz, J. (10 de Junio de 2010). *Universidad Distrital Jose Caldas*. Recuperado el 07 de Mayo de 2013, de Tecnología de los Circuitos Integrados: <http://200.69.103.48/comunidad/profesores/jruiz/jairocd/texto/usm/ci/1504tci.pdf>
- ✓ [8] Nashelsky, B. (2003). *Teoria de circuitos y dispositivos electrónicos 8 edición*. México: Prentice Hall.
- ✓ [9] Ned Mohan, Tore M. Undeland, William P. Robbins. (2009). *Electronica de Potencia, Convertidores, aplicaciones y diseño, Edición 3*. México: Mc Graw Hill.
- ✓ [10] Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss. (2007). *Sistemas Digitales Principios y Aplicaciones Edición 10*. México: Prentice Hall.

- ✓ [11] Gonzales, J. M. (Febrero de 2004). *Automatizacion de Procesos Industriales*. Recuperado el 10 de Mayo de 2013, de <http://www.araba.ehu.es/depsi/jg/APIslides.pdf>
- ✓ [12] Cassá, J. O. (Junio de 2005). *Automatización de Procesos Industriales*. Recuperado el 10 de Mayo de 2013, de <http://www.iit.upcomillas.es/pfc/resumenes/42bbf80238332.pdf>
- ✓ [13] Datasheet L298
- ✓ [14] Datasheet AD620
- ✓ [15] Datasheet TL084

- ✓ [16] Buendía, M. (Junio de 2009). *Comunicaciones Industriales*. Recuperado el 16 de Mayo de 2013, de http://wiki.pachecolopez.es/lib/exe/fetch.php?media=electronica:manual_protocolo_modbus_castellano_.pdf