

UNIVERSIDAD SAN FRANCISCO DE QUITO

COLEGIO DE CIENCIAS E INGENIERÍA

**PLANTA DE CLASIFICACIÓN Y ORGANIZACIÓN AUTOMÁTICA: PUENTE
GRÚA AUTÓNOMO Y SISTEMA DE SUPERVISIÓN Y CONTROL
UTILIZANDO UN PLC CUBLOC**

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
ELECTRÓNICO

Paúl Alejandro Frutos Galarza

Carlos Santiago Garzón Ibarra

Omar Aguirre, M.Sc. Director de Tesis

Quito, Enero 2013

Universidad San Francisco de Quito
Colegio de Ciencias e Ingeniería Politécnico

HOJA DE APROBACIÓN DE TESIS

PLANTA DE CLASIFICACIÓN Y ORGANIZACIÓN AUTOMÁTICA: PUENTE GRÚA AUTÓNOMO Y SISTEMA DE SUPERVISIÓN Y CONTROL UTILIZANDO UN PLC CUBLOC

Paúl Alejandro Frutos Galarza

Carlos Santiago Garzón Ibarra

Omar Aguirre, M.Sc.
Director de la Tesis

René Játiva, D.E.A.
Miembro del Comité de Tesis

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

Ximena Córdova, Ph.D.
Decana del Colegio de Ingeniería Politécnico

Quito, Enero 2013

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy 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, autorizo 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: Paul Alejandro Frutos Galarza

C. I.:171927175-9

Fecha: 2013/01/15

Firma:

Nombre: Carlos Santiago Garzón Ibarra

C. I.:171244470-0

Fecha: 2013/01/15

AGRADECIMIENTOS

A nuestras familias por su amor, comprensión y apoyo a lo largo de todo el camino recorrido.

A Nelson Herrera y Alicia Rivera por toda la ayuda brindada durante nuestra carrera en la universidad.

A los profesores de Ing. Electrónica, Física y Matemática y a los amigos físicos e ingenieros

DEDICATORIA

A mis padres, abuelos y hermanos por su gran amor y comprensión
A Raquel, por sus enseñanzas y mostrarme que con esfuerzo y dedicación se puede alcanzar
cualquier meta.

Paúl Alejandro Frutos Galarza

A mi esposa, a mis hijos, a mis padres por su amor, comprensión y apoyo en todo este tiempo.

A la Policía Nacional del Ecuador por la oportunidad de superación que será retribuido a tan
noble institución.

Carlos Santiago Garzón Ibarra.

Resumen

El presente proyecto tiene por finalidad desarrollar un sistema automático de clasificación y organización de elementos. La planta diseñada es un puente grúa automatizado, el cual está gobernado por un sistema SCADA, además se cuenta con un sistema de visión y memoria.

La planta diseñada primeramente diferencia y clasifica los elementos o fichas dependiendo de su color mediante el sistema de visión, para luego organizarlos y posicionarlos en un pallet tipo tablero de ajedrez. Todo este proceso de clasificación y organización es automático y está gobernado por un PLC CUBLOC. La planta no cuenta con sensores en el pallet que indiquen la presencia o no de un objeto en las distintas posiciones, pero en cambio se encuentra implementado un sistema de memoria que recuerda el lugar donde se ubicaron los elementos con el fin de permitir que se desarrolle de manera óptima el proceso de organización. El sistema de supervisión y control que se encuentra instalado tiene la misión de verificar y controlar parcialmente las actividades que realiza la planta, de esta manera, en ciertos casos se requieren de autorizaciones por parte de un operador o un elemento externo para continuar con el proceso de clasificación y organización, evitando así los errores.

El presente proyecto muestra detalladamente la programación y la configuración de la comunicación entre el PLC de la marca CUBLOC, utilizado para la automatización, y el ordenador.

Abstract

This project shows the design and construction process of an automated station that can perform classification and organization of elements. The station is an automated overhead crane, which is governed by a SCADA system. The station also has a vision system and a memory system.

First, the automated station identifies and classifies the elements depending on the color using the vision system; then, the plant organizes and places the elements in a pallet. This whole process is automatic and is governed by a CUBLOC PLC. The plant has no sensors in the positions of the pallet to indicate the presence or absence of an object, but instead it is implemented a memory system that can remember where the elements were placed. The monitoring and control system has the task of verifying and controlling the activities of the station, thus, in certain cases the plant require an authorization by the operator or an external device to continue its activities.

This project shows in detail the programming and configuration of the communication between the CUBLOC PLC and the computer.

TABLA DE CONTENIDO

1. CAPÍTULO I.....	1
1.1. Introducción.....	1
1.2. Objetivos.....	2
1.2.1. Objetivo General.....	2
1.2.2. Objetivos Específicos.....	2
2. CAPÍTULO II - MARCO TEÓRICO.....	3
2.1. Sistemas de Control.....	3
2.2. Definiciones básicas previas a los Sistemas de control.....	3
2.2.1. Variables.....	3
2.2.2. Plantas.....	3
2.2.3. Procesos.....	3
2.2.4. Sistemas.....	3
2.2.5. Señales.....	4
2.3. Definición Sistema de Control.....	4
2.4. Automatización industrial.....	5
2.4.1. Concepto.....	5
2.4.2. Niveles de Automatización. Sistema SCADA.....	5
2.5. Sistema SCADA.....	8
2.5.1. Definición.....	8
2.5.2. Protocolo de Comunicación.....	8
2.5.3. OPC Server.....	8
2.5.4. Partes de un Sistema SCADA.....	9
2.5.5. Implantación de sistema SCADA.....	10
2.5.6. Funciones de un SCADA.....	10
3. CAPÍTULO III - PLANTA, DESCRIPCIÓN, DISEÑO E IMPLEMENTACIÓN.....	11
3.1. Sistema Electromecánico.....	11
3.1.1. Estructura.....	11
3.1.1.1. Estructura Base o Estática.....	11
3.1.1.2. Pallet.....	12
3.1.1.3. Estructura Móvil.....	13
3.1.2. Motores de corriente continua.....	15
3.1.3. Motores pasó a paso unipolares.....	15
3.1.4. Electroimán.....	17
3.2. Elementos Electrónicos.....	17
3.2.1. Sensores Reed Switch.....	17
3.2.2. Controlador Lógico Programable.....	18
3.2.3. Drivers de Control Motores DC.....	19
3.2.4. Drivers de Control Motores Paso a Paso.....	21
3.2.5. Diagramas de los circuitos y diseño de las placas PCB.....	22
3.2.6. Descripción del Cableado.....	24

4. CAPÍTULO IV - Programación y Automatización.....	25
4.1. Sistema de Visión.....	25
4.2. Interfaz de conexión al PLC.....	30
4.3. Lógica de la Programación.....	31
4.4. Descripción del Programa.....	33
4.4.1. Sentencias Especiales.....	33
4.4.2. Inicialización de Variables.....	34
4.4.3. Subrutinas.....	35
4.4.3.1. Parada.....	35
4.4.3.2. Adelante.....	35
4.4.3.3. Atrás.....	36
4.4.3.4. Recoger Producto.....	36
4.4.3.5. Entregar Producto.....	37
4.4.4. Clasificación de Objetos.....	38
4.4.5. Ciclo de Ejecución.....	38
5. CAPÍTULO V - SCADA.....	41
5.1. Conexión Física.....	41
5.2. Protocolo MODBUS.....	42
5.3. Configuración de la Comunicación.....	44
5.3.1. Programación en el PLC.....	44
5.3.2. Programación del Software.....	48
5.3.2.1. Consideraciones Generales.....	48
5.3.2.2. Indusoft.....	50
5.3.2.3. NI OPC Servers.....	54
5.3.3. Monitoreo y supervisión, HMI.....	56
6. Conclusiones y Recomendaciones.....	60
6.1. Conclusiones.....	60
6.2. Recomendaciones.....	62
BIBLIOGRAFÍA.....	63

LISTADO DE FIGURAS

Figura 1. Señal PWM.....	4
Figura 2. Niveles de automatización.....	5
Figura 3. PLC marca CUBLOC.....	6
Figura 4. OPCServer.....	9
Figura 5. Sistema SCADA.....	10
Figura 6. Diseño AUTOCAD Estructura Base.....	12
Figura 7. Descripción Pallet.....	13
Figura 8. Guías y topes mecánicos.....	14
Figura 9. Elemento Móvil.....	15
Figura 10. Micrometal Gear Motor 100:1.....	15
Figura 11. Esquema motor paso a paso	16
Figura 12. Esquema motor paso a paso	16
Figura 13. Motor paso a paso unipolar.....	17
Figura 14. Actuador Final.....	17
Figura. 15. Funcionamiento del un sensor Reed Switch.....	18
Figura 16. CUSB-22R.....	19
Figura 17. Esquema L293.....	20
Figura 18. Descripción de los Pines del circuito Integrado.....	20
Figura 19. Circuito para cada motor DC con un puente H completo.....	21
Figura 20. Circuito Propuesto para el control de cada motor paso a paso unipolar.....	22
Figura 21. Circuito controlador motores DC.....	23
Figura 22. PCB del controlador motores DC.....	23
Figura 23. Circuito controlador del motor paso a paso.....	23
Figura 24. PCB del controlador del motor paso a paso.....	24
Figura 25. Lógica del programa sistema de visión.....	26
Figura 26. Diagrama de Bloques del programa de identificación de colores en Labview.....	27
Figura 27. Panel frontal del programa.....	29
Figura 28. Máquina de estados del programa del PLC.....	32
Figura 29. Inicialización de Variables.....	34
Figura30.Parada.....	35
Figura 31. Adelante.....	36
Figura 32. Atrás.....	36
Figura 33. Recoger Producto.....	37
Figura 34. Entregar Producto.....	38
Figura 35. Clasificación de Objetos.....	38
Figura 36. Ciclo de Ejecución.....	40
Figura 37. Conexión PLC CUSB-22R, Protocolo Modbus sobre RS-232 en canal 1.....	42
Figura 38. Formato de Frame Modbus RTU.....	42
Figura 39. Declaración Pines de Entrada a ser utilizados en Ladder	46
Figura 40. Declaración registros tipo M con su correspondiente alias.....	47
Figura 41.- Registros para lectura y escritura utilizados como contactores.....	48
Figura 42.- Selección del driver de comunicación.....	50
Figura 43. Propiedades de comunicación MODBUS.....	51
Figura 44. Configuración de los campos RC, RS, WC y WS.....	51

Figura 45. Formato Direccionamiento MODBUS en Indusoft.....	52
Figura 46. <i>Main Driver Sheet</i>	53
Figura 47. Canal y Driver de la comunicación.....	54
Figura 48. Parámetros de comunicación.....	54
Figura 49. Configuración del Dispositivo.....	55
Figura 50. Formato direccionamiento MODBUS en NI OPC Servers.....	55
Figura 51. HMI.....	56

LISTADO DE TABLAS

Tabla 1. Entradas Lógicas, movimientos de los dos motores DC.....	21
Tabla 2. Descripción de I/O Digitales.....	31
Tabla 3. Funciones de Modbus que soporta PLC CUBLOC	43
Tabla 4. Correspondencia entre direcciones Modbus y registros del CPU CB280.....	44
Tabla 5.- Dirección Modbus de los espacios de memoria utilizados en el PLC CUBLOC.....	48

1. CAPÍTULO I

1.1. INTRODUCCIÓN

La automatización es una de las tareas que se realizan muy a menudo en ingeniería. Por este motivo abarca muchos de los conceptos de diferentes áreas tales como instrumentación industrial, sensores, transmisores de campo, sistemas de control y supervisión, sistemas de transmisión y recolección de datos y por supuesto las aplicaciones de software en tiempo real para supervisar, controlar las operaciones de plantas.

En el presente proyecto se describe la construcción de un modelo de un sistema de clasificación y organización de elementos, en el cual se exploran muchas de las áreas que potencialmente conforman un proceso de automatización, desarrollando varias alternativas tecnológicas que podrían ser utilizadas como guías o referentes para cualquier tipo de proyectos que requieran sistemas sensoriales y de control específicos (i.e. detección de color, comunicación entre controladores).

El modelo diseñado e implementado es un puente grúa autónomo que cuenta con su propio sistema de supervisión y control, posee también un sistema de visión que permite diferenciar objetos. Se busca obtener un sistema totalmente automático con capacidad de decisión, y memoria, estas dos características permitirán que el prototipo pueda clasificar y organizar productos o elementos. El sistema SCADA (supervisión, control y adquisición de datos) del prototipo cuenta con una interfaz hombre-máquina, interfaz que puede ser visualizada a través del monitor de una computadora.

En el primer capítulo de este documento se presenta el objetivo general y los objetivos específicos de este proyecto. En el Capítulo 2 se muestran los conceptos básicos que se utilizan en la automatización industrial. A partir del Capítulo 3 empieza la descripción de la construcción, diseño e implementación de la planta, desde los elementos mecánicos que soportan y brindan la facilidad para la dinámica del proceso, los actuadores y sensores, hasta la descripción del Controlador Lógico Programable y los drivers de control. En el Capítulo 4 se

muestra la lógica de programación y se describe el código cargado en el PLC así como también la programación del sistema de Visión. La descripción de la configuración de la comunicación y todos los parámetros que involucra son descritos detalladamente en el Capítulo 5 junto con el proceso de monitoreo y control. Por último se expresan las conclusiones del trabajo y se hace una lista de recomendaciones que podrían ser tomadas en cuenta para proyectos futuros que involucren tareas de automatización.

1.2. OBJETIVOS

1.2.1. OBJETIVO FINAL DEL PROYECTO

Fabricar un modelo a escala de un puente grúa, utilizando control automático regulado por un PLC CUBLOC, un sistema de Visión y Control Scada, que permita organizar elementos en una grilla.

1.2.2. OBJETIVOS ESPECÍFICOS DEL PROYECTO

- Fabricar el modelo a escala de un puente grúa que permita movimientos X-Y-Z con electroimán para recoger los productos.
- El puente grúa se desplazará encima de un arreglo cuadrado (tipo ajedrez).
- Implementar un sistema de visión que permita visualizar el tipo de elemento.
- Implementar una base, grilla, en la cual se ubicarán los elementos.
- Programar el funcionamiento automático con el PLC CUBLOC.
- Desarrollar un sistema de memoria que reemplace a los sensores de presencia.
- Realizar la Comunicación entre el PLC CUBLOC y el Ordenador
- Implementar un sistema de control SCADA.

2. CAPÍTULO II - MARCO TEÓRICO

2.1. Sistemas de control

Se tiene conocimiento que uno de los primeros trabajos de control automático fue el que realizó James Watt al controlar la velocidad de su máquina de vapor [1]. Desde el siglo XVIII se inició con esta ola de desarrollo tecnológico.

2.2. Definiciones básicas previas a los sistemas de control.

2.2.1. Variables

Existe una clasificación para las variables: controlada y manipulada, la primera corresponde a una cantidad o condición de una señal a controlar a través de un controlador. La variable manipulada está ligada a la primera ya que al controlar y medir la variable controlada se puede utilizar esta información para aplicar a la variable manipulada.

2.2.2. Plantas

Puede considerarse a una planta a un conjunto de elementos de una máquina o robot que trabajan juntos para realizar una acción o trabajo particular [1]. En la presente tesis la planta corresponde al puente grúa en sí.

2.2.3. Procesos

Corresponde a un conjunto de acciones organizadas, y sistemáticas que desencadenarán en un fin u objetivo.

2.2.4. Sistemas

Oppenheim en su libro *Señales y Sistemas* define a un sistema de la siguiente manera: “Un sistema puede considerarse como un proceso en el cual las señales de entrada son transformadas por el sistema o provocan que éste responda de alguna forma, lo que da como resultado otras señales como salidas.”[2]

2.2.5. Señales

- **PWM**

Según Houcine Hassan, profesor de la Universidad Politécnica de Valencia, “PWM (Pulse Width Modulation o Modulación por ancho de pulso) es un método de conmutación sencillo y que consiste en aplicar todo el voltaje de alimentación a la carga y después retirarlo. Los tiempos de nivel alto y bajo son controlados y el resultado en la carga es el mismo que si se hubiese aplicado una menor tensión de forma continua. Variando los tiempos en los que se aplica la tensión podremos controlar la velocidad de un motor” [3] para el presente proyecto se utilizan motores DC que serán controlados desde el PLC CUBLOC que nos da la facilidad de las señales PWM.

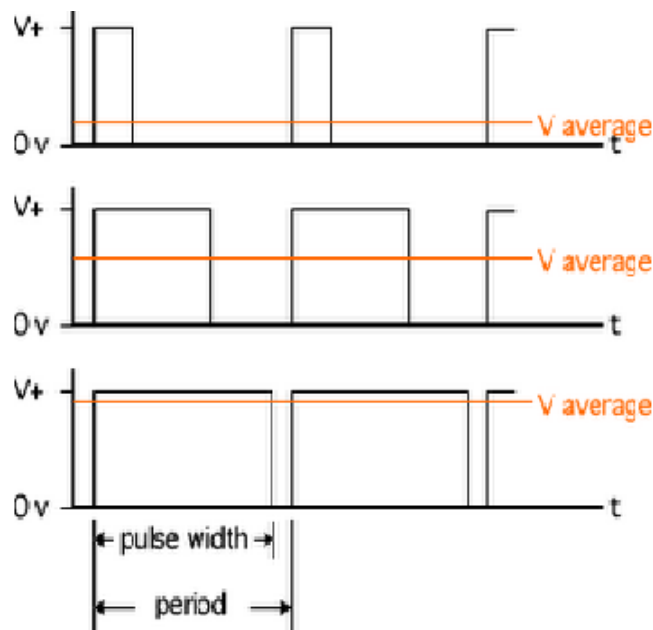


Figura 1. Señal PWM [3]

- **Señales a nivel lógico TTL**

Estas señales se utilizan en los circuitos de control, y pueden tomar valores únicamente de $0VDC$ (nivel lógico bajo) y $5VDC$ (nivel lógico alto)

2.3. Definición Sistema de Control

Una vez analizados conceptos previos se puede definir a un sistema de control como un sistema o un subsistema que tiene un conjunto de componentes o elementos que en forma sistemática y

organizada le permiten controlar su comportamiento en forma continua con la única finalidad de cumplir con una meta previamente establecida.

2.4. Automatización industrial

2.4.1. Concepto

Se puede decir que la automatización industrial dentro de un proceso utiliza sistemas electromecánicos y computarizados que controlan a una máquina que busca remplazar al hombre en ciertas tareas, según Emilio García Moreno, en su libro *Automatización de procesos industriales*, “El concepto de automatización lleva implícitamente la supresión total o parcial de la intervención humana en la ejecución de diversas tareas industriales. La aplicación de la automatización va desde las tareas más sencillas como la regulación de la temperatura de un horno, hasta las aplicaciones más complejas como la regulación a través de un sistema computarizado de un sistema bancario” [4]

2.4.2. Niveles de Automatización.

Aunque en este proyecto se describe el diseño y la implementación de los niveles 0, 1, 2 que se muestran en la Figura 2, en esta sección se realiza una revisión general de todos los niveles de automatización.

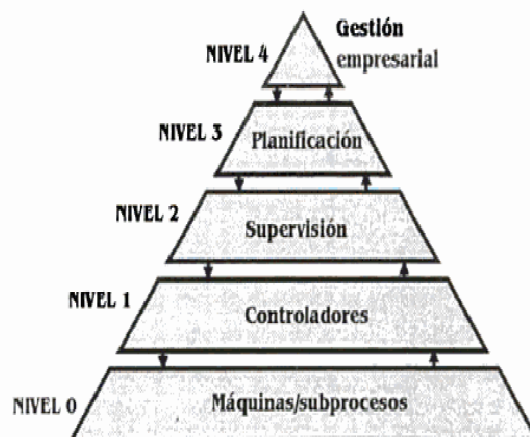


Figura 2. Niveles de automatización [4]

Los niveles de automatización dependen del objetivo a alcanzar, los diferentes niveles que se conocen son:

a. Nivel Cero

Conformado por los dispositivos o equipos que reciben algún tipo de señal física y la transforman en otro tipo de señal como los sensores, en este nivel también se encuentran los actuadores con sus respectivos drivers. Este nivel es el de más bajo rango en la estructura de los niveles de automatización

b. Nivel 1.

Este es el nivel de control, y corresponde a la parte de la programación del PLC, de donde se espera que los actuadores ejecuten una tarea previamente deseada en forma independiente y automática. En el presente proyecto se utiliza un PLC marca CUBLOC® ya que nos facilita una mayor cantidad de salidas PWM que serán aprovechadas por los actuadores que hacen que la estructura pueda moverse en diferentes direcciones.



Figura 3. PLC marca CUBLOC [6]

c. Nivel 2.

Este nivel corresponde al nivel de supervisión y control, nivel en el cual se realizan las siguientes acciones:

- Adquisición de datos así como análisis de los mismos.
- Monitoreo.
- Tratamiento de alarmas y asistentes.
- Monitoreo
- Seguimiento de las órdenes previamente programado.

En este nivel como nos dice Emilio García, “Depende de la filosofía de control de la empresa, a este nivel se emite ordenes de ejecución al nivel 1 y recibe situaciones de estado de dicho nivel. Igualmente recibe programas de producción, calidad, mantenimiento, etc. Del Nivel 3 y realimenta dicho nivel con las incidencias (estado de órdenes de trabajo, situación de máquinas, estado de la obra en curso, etc) en la planta” [4], es decir en otras palabras como su nombre lo indica es un nivel meramente de supervisión.

d. Nivel 3

Este nivel corresponde al nivel de planificación, como su nombre lo indica corresponde a un nivel donde se planifica la programación de la producción, se hace el requerimiento de los materiales necesarios para realizar un producto, así como a los diferentes análisis financieros, etc. En este nivel se ubica la gestión de distribución, es decir en este nivel se programa las órdenes que el nivel 2 deberá ejecutar.

e. Nivel 4.

Este es un nivel de gestión empresarial que se dedica única y exclusivamente a la parte administrativa de la industria, entre las actividades más relevantes que se desarrollan en este nivel podemos mencionar: Planificación estratégica, ingeniería de procesos, gestión tecnológica, Investigación y desarrollo.

2.5. Sistema SCADA.

2.5.1. Definición

SCADA (Supervisión, Control y Adquisición de Datos). Un SCADA es un sistema basado en computadoras que permite supervisar y controlar a distancia una instalación de cualquier tipo. Los sistemas SCADA son controlados por el operador. La ventaja de la utilización de un sistema SCADA es que la medición de las variables se las puede hacer en tiempo real y pueden ser monitoreadas a través de un HMI.

2.5.2. Protocolo de comunicación

Un protocolo es un conjunto de reglas de comunicaciones entre dispositivos. Los protocolos gobiernan el formato, sincronización, secuencia y control de errores. Sin estas reglas, los dispositivos no podrían detectar la llegada de bits.

2.5.3. OPC Server

Del acrónimo en inglés, OLE for Process Control, se puede definir como sistema de comunicación estándar para control y supervisión de procesos [5]. Utiliza una arquitectura Cliente – Servidor, donde el esclavo corresponde al servidor y el maestro corresponde al cliente, es bidireccional donde el maestro o cliente puede escribir en los esclavos. Un OPC Server es una aplicación que actúa como una interface entre el PLC y el ordenador. El OPC Server se conecta al PLC e interactúa (leer y escribir) con la memoria del mismo, envía los datos en el formato estándar OPC. En este proyecto se utiliza el OPC Server de National Instruments.



Figura 4. OPC Server [5]

2.5.4. Partes de un Sistema SCADA.

- **Sensores.**

Los sensores son los encargados de transformar una señal física en una señal eléctrica, las señales eléctricas más utilizadas son: voltaje, corriente, carga, etc.

- **Acondicionadores de señales.**

Se encargan de interpretar y tratar la señal eléctrica para que sea entendida por el computador.

- **Conversión de datos.**

Cuando los datos han sido acondicionados, la señal se convierte en un valor digital equivalente, esta función es tratada por un conversor análogo/digital. Estos datos son almacenados en la computadora para su posterior análisis.

- **Presentación de la información.**

Uno de los requisitos del sistema SCADA es presentar la información en forma clara, los datos tienen que ser de fácil interpretación para el operador.

- **Salidas de Control.**

Si bien es cierto el operario decide las acciones de la planta, se generan salidas de control que accionarán diferentes elementos electromecánicos como relés, etc.

2.5.5. Implantación de sistema SCADA.

Previo a la instalación de un sistema SCADA, Molina José nos dice que una planta debe cumplir los siguientes requerimientos:

- El número de variables del proceso que se necesita monitorear es alto.
- El proceso está geográficamente distribuido. Esta condición no es limitativa, ya que puede instalarse un SCADA para la supervisión y control de un proceso concentrado en una localidad.
- La información del proceso se necesita en el momento en los cambios que se producen en el mismo, o en otras palabras, la información se requiere en tiempo real.
- La necesidad de optimizar y facilitar las operaciones de la planta, así como la toma de decisiones, tanto gerenciales como operativas.
- Los beneficios obtenidos justifica la inversión de un sistema SCADA.
- La complejidad y velocidad del proceso permiten que la mayoría de las acciones de control sean iniciadas por un operador.

2.5.6. Funciones de un SCADA.

Las funciones principales de un sistema SCADA son: almacenar, mostrar la información, mostrar acciones de control, alarmas de alerta al operador, y aplicaciones en general dependiendo de la información obtenida de la variable analizada.

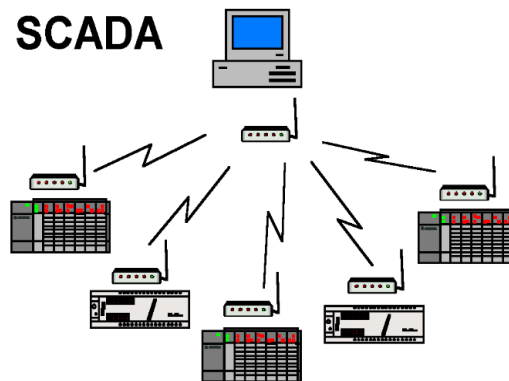


Figura 5. Sistema SCADA [5]

3. CAPÍTULO III - PLANTA, DESCRIPCIÓN, DISEÑO E IMPLEMENTACIÓN

3.1. Sistema Electromecánico

3.1.1. Estructura

La estructura sobre la cual se desarrollan las actividades de clasificación y organización de los objetos se basa en dos elementos, una estructura estática, sobre la cual se sostienen todos los dispositivos del sistema, y una estructura móvil que está asentada sobre la primera y la cual es capaz de realizar movimientos en los ejes X, Y y Z.

3.1.1.1. Estructura Base o Estática

La estructura base que se necesita debe cumplir con ciertas condiciones que faciliten la tarea de clasificar organizar las piezas en una cuadrilla. La estructura base consiste en las aristas de un paralelepípedo. En la superficie inferior se encuentra asentada la cuadrilla o pallet, que es una superficie dividida en un arreglo tipo tablero de ajedrez, la cual se encuentra particionada en varios segmentos donde se colocan las piezas desde la parte superior. Como ya se ha mencionado el sistema tiene por tarea moverse en los 3 ejes espaciales de las coordenadas cartesianas, así los ejes X e Y son utilizados para posicionar los productos y el eje Z sirve para elevar el producto, recogerlo, transportarlo y entregarlo. De ahí que en las aristas superiores de la estructura se asienta el elemento móvil, el cual permite el movimiento en el eje X y en el eje Y del actuador final, el cual también tiene un movimiento en Z para tomar la ficha seleccionada. En este proyecto se utiliza un electroimán como actuador final, el cual toma las piezas de la parte inferior para ser elevadas y transportadas en el plano XY para posteriormente ser colocadas en una posición indicada. En la Figura 6 se muestra el diseño de la estructura en AUTOCAD en el cual se pueden apreciar las características mencionadas anteriormente.

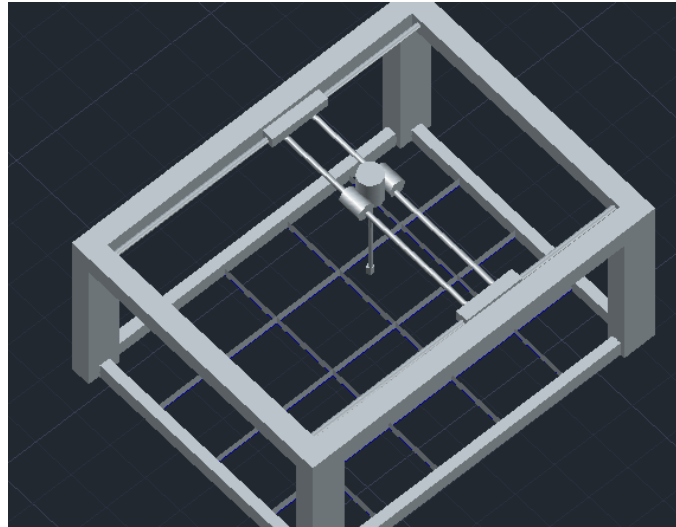


Figura 6. Diseño AUTOCAD Estructura Base

Creado por: Grupo diseñador

Las dimensiones de la estructura son 50 cm de largo, 40 cm de ancho y 45cm de altura. Como se puede apreciar en la Figura 6 hay una grilla en la cual se organizarán los elementos.

3.1.1.2. Pallet

La Figura 7 muestra la descripción del pallet que se encuentra ubicado en la parte inferior de la estructura base. El pallet esta referenciado por los dos ejes cartesianos. El eje X posee 5 subdivisiones y el eje Y posee 3 subdivisiones, de ahí que se tienen 15 segmentos en total. A cada una de las visiones del eje X y del eje Y se les asignó un nombre según su orden, además cada segmento que se muestran en la Figura 7 se ha nombrado con las siglas SXY, donde la letra S, corresponde únicamente a la palabra en inglés *spot*, seguida por el par ordenado XY, que especifica la posición en los ejes cartesianos.

Pallet			
X4	S41	S42	S43
X3	S31	S32	S33
X2	S21	S22	S23
X1	S11	S12	S13
X0	S01	S02	S03
	Y1	Y2	Y3

Figura 7. Descripción Pallet

Creado por: Grupo diseñador

3.1.1.3. Estructura Móvil

Como ya se pudo notar este componente se asienta en la superior sobre las vigas de la estructura base, y es donde se encuentra el actuador final (electroimán). La estructura móvil es similar a un vehículo, posee cuatro llantas que permiten el desplazamiento en el eje X, es decir hacia adelante o hacia atrás únicamente. Las llantas posteriores están acopladas a dos motores de corriente directa con reducción mecánica, los cuales son encargados de proveer el torque para mover la estructura, por su parte las llantas delanteras forman un mecanismo de ayuda para el soporte y el transporte de la estructura, estas llantas no están conectadas a ningún motor. Con el fin de evitar que la estructura móvil no pierda su trayectoria se tiene dos guías para las ruedas de la izquierda y para las ruedas de la derecha, así se evita que la estructura pueda cambiar de dirección. Al final de cada guía se colocaron dos topes mecánicos que evitan que se traspasen los límites de operación.



Figura 8. Guías y topes mecánicos

Creado por: Grupo investigador

En el elemento móvil se montó un motor paso a paso el cual mueve un contenedor plástico en Y. En este contenedor se encuentra colocado otro motor paso a paso que permite el movimiento del actuador final en Z. Este motor está conectado a un eje que se acopla a un carrete que envuelve una cuerda que sostiene al electroimán el cual se acciona para tomar las fichas metálicas. En la Figura 9 podemos apreciar el elemento móvil con todos los componentes y actuadores que hemos mencionado.

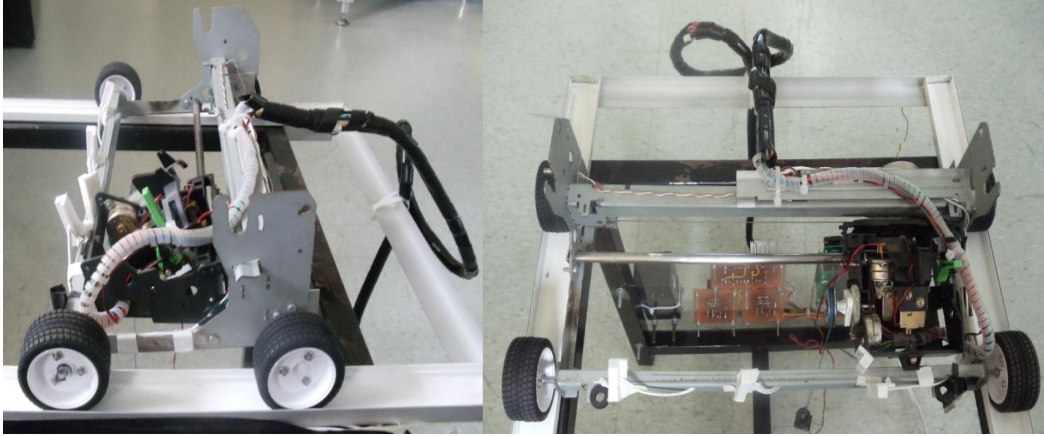


Figura 9. Elemento Móvil

Creado por: Grupo diseñador

3.1.1.4. Motores de corriente continua

Como ya se mencionó cuando se describió la estructura móvil, para los movimientos en el eje X se cuenta con dos motores de corriente continua con reducción de 100:1. A voltaje nominal de 6 voltios, su velocidad es 320 RPM con un torque de $2.2 \frac{Kg}{cm^2}$ [8]. Estos motores son los que se muestran en la Figura 10, su rango de operación está entre los 3 y 9 voltios y son controlados mediante señales PWM (*Pulse width modulation*), las cuales se obtienen de las salidas del PLC.



Figura 10. Micrometal Gear Motor 100:1 [8]

3.1.1.5. Motores paso a paso unipolares

En la estructura móvil se encuentran incorporados dos motores paso a paso unipolares. En la Figura 11 se puede ver el esquema de un motor paso a paso. Estos motores poseen cuatro

bobinas, la corriente que circula por estas bobinas lo hace en el mismo sentido. Posee seis cables externos, uno para cada bobina y dos cables comunes para la fuente de alimentación de cada uno de los pares de bobinas.

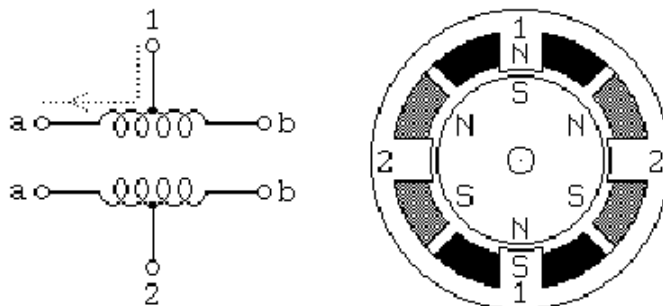


Figura 11. Esquema motor paso a paso [9]

Se utilizan varias secuencias para manejar un motor paso a paso, la secuencia de control que se ha utilizado en este caso es la secuencia paso simple mostrada en la Figura 12, que consiste en encender una bobina a la vez y por separado [9]. Así se logra un paso cada vez que se acciona una bobina y el sentido de giro depende del orden de encendido de las bobinas.

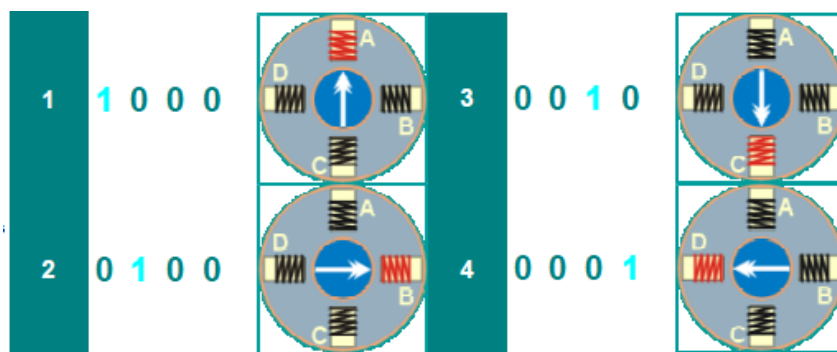


Figura 12. Esquema motor paso a paso [9]

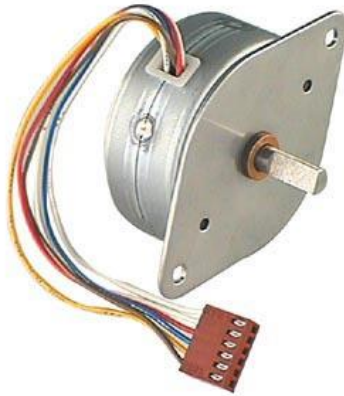


Figura 13.- Motor paso a paso unipolar

Creado por: Grupo diseñador

3.1.1.6. Electroimán

Para actuador final se utiliza un electroimán. Este elemento se obtuvo fácilmente desarmando un relé.

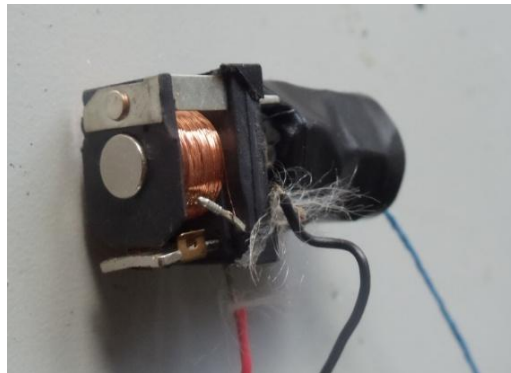


Figura 14.- Actuador Final

Creado por: Grupo diseñador

3.2. ELEMENTOS ELECTRÓNICOS

3.2.1. Sensores Reed Switch

En la planta se utilizan 8 sensores Reed Switch, 5 de ellos están colocados a lo largo del eje X y tres en el eje Y. Se utiliza este tipo de sensor para detectar la posición en el eje X e Y. La ventaja es la utilización de imanes para activar el switch, lo cual brinda mayor certeza en la

detección de objetos en movimiento. En la Figura 15 se muestra el funcionamiento básico de ese tipo de sensores los cuales son activados por un campo magnético.

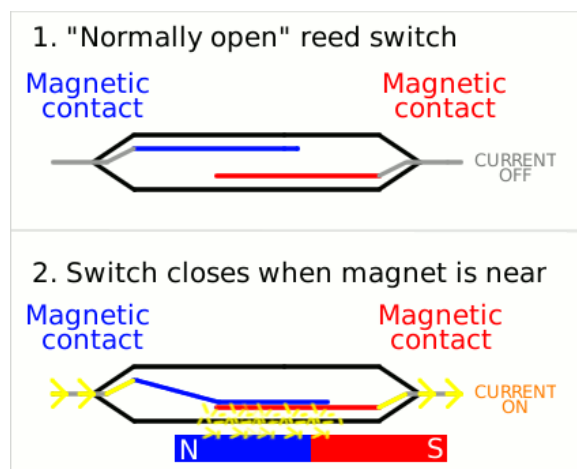


Figura 15. Funcionamiento de un sensor Reed Switch [10]

3.2.2. Controlador Lógico Programable

EL PLC que se utiliza en este proyecto es el equipo CUSB-22R de la marca CUBLOC, el cual posee un procesador CB280. Este dispositivo tiene 6 salidas tipo relé, 11 entradas aisladas por optoacopladores, 6 salidas PWM, 4 entradas TTL, dos contadores de alta velocidad, 6 entradas análogas, 2 puertos RS232 y un puerto CuNET propio de la línea CUBLOC. Se decidió utilizar este PLC ya que se tiene la ventaja de programar en Ladder o en Basic, además se aprovechan todas sus salidas PWM, para el control de los 4 motores. La figura siguiente muestra el esquema CUSB-22R, en donde se pueden apreciar las salidas y entradas.

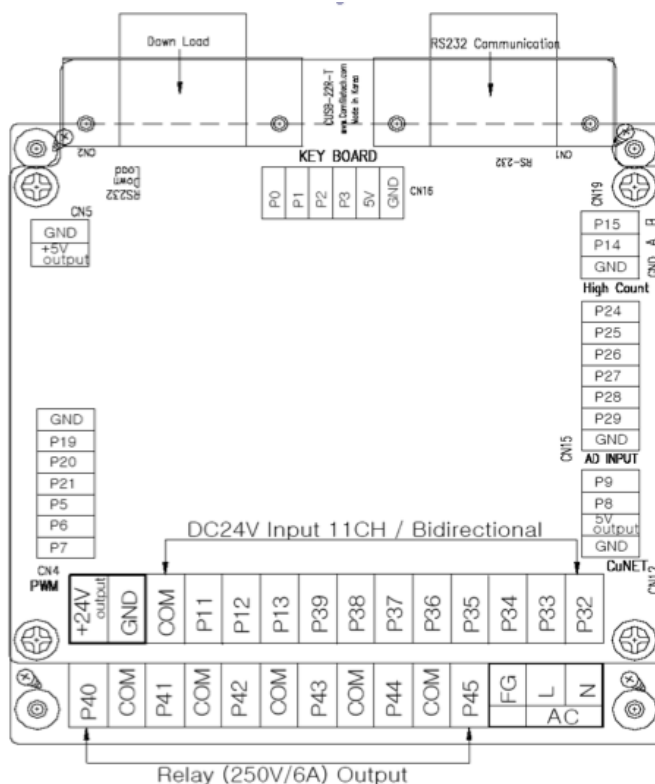


Figura 16. CUSB-22R [6]

3.2.3. Drivers de Control Motores DC

Para el control de los motores DC se utiliza un circuito integrado L293, el cual es un doble puente H. Este circuito integrado actúa como un amplificador de corriente ya que con una señal de control de corriente baja provee una señal de corriente alta con la que se alimenta y se controla el motor. Como ya se mencionó el integrado L293 tiene dos puentes H por lo que basta utilizar uno de estos en el proyecto para los dos motores DC. El modo de operación de los motores se los controla con niveles de voltaje TTL en los pines 2, 7 y 10, 15. En las siguientes figuras se muestra el esquema de un L293, y la descripción de sus pines.

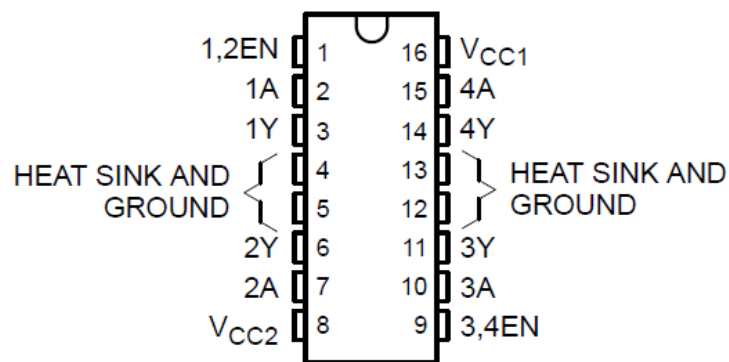


Figura 17. Esquema L293 [11]

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc 2
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc 1

Figura 18. Descripción de los Pines del circuito Integrado L293 [11]

Para cada puente H completo, se necesitan 4 diodos estándar de protección para corriente, conectados en paralelo que permitan que las corrientes circulen en sentido inverso cada vez que conmute el voltaje, ya que el motor DC está compuesto por bobinados que durante periodos de tiempo se opondrán a que la corriente varíe. En la Figura 19 se muestra el circuito propuesto para cada motor DC.

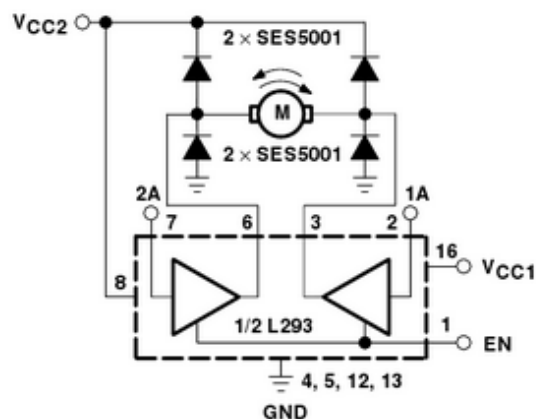


Figura 19. Circuito para cada motor DC con un puente H completo [11].

En este proyecto para el control PWM del puente H se debe tomar en cuenta la siguiente tabla de verdad para los movimientos del robot. En los cuales se definirá si se quiere avanzar, retroceder o parar.

1A(Pin 2)	2A(Pin 7)	3A(Pin 10)	4A(Pin 15)	Acción
1	0	1	0	Avance
0	1	0	1	Retroceso
0	0	0	0	Parada

Tabla 1. Entradas Lógicas, movimientos de los dos motores DC

Creado por: Grupo diseñador

3.2.4. Drivers de Control Motores Paso a Paso

Para el caso del motor paso a paso al ser unipolar, se tuvo en cuenta que era suficiente un amplificador Darlington para la salida del controlador. El controlador utilizado se encuentra en el mercado, es un L297, el cual provee una máquina de estados en el que se encuentra la secuencia de movimiento del motor, para el giro en counter clockwise (ccw) o clockwise (cw). Una vez encontrados los controladores, en la hoja técnica de éstos se puede encontrar circuitos de ejemplo con sus debidas protecciones para probar el control de cada motor.

Para el motor paso a paso utilizamos un L297 y un ULN2003 el cual es un conjunto de amplificadores Darlington dentro de un integrado. Este circuito al no conmutar el voltaje no se espera un cambio de circulación de corriente por lo que no necesita protecciones adicionales.

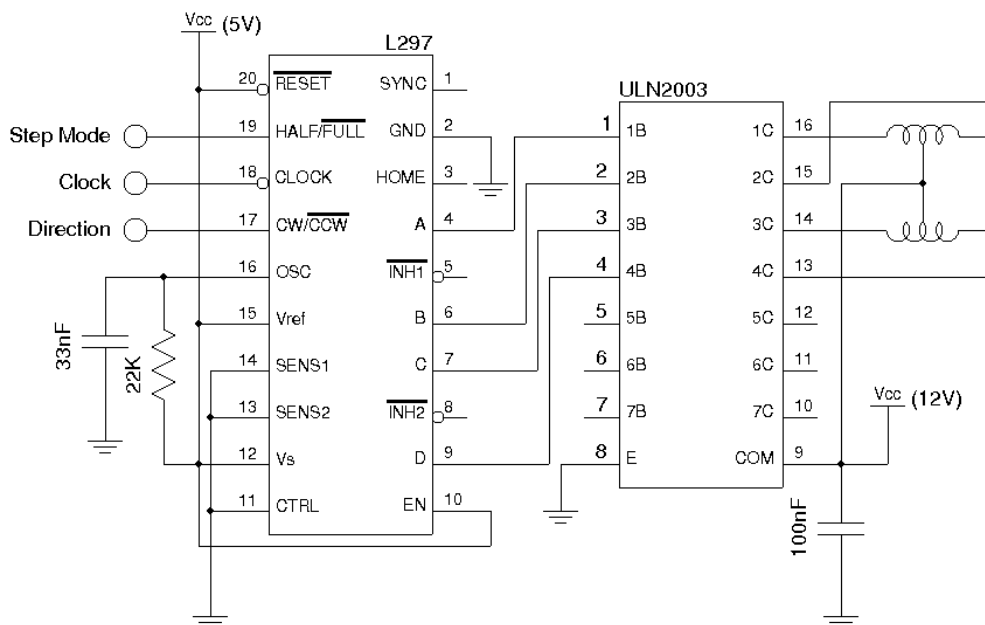


Figura 20. Circuito propuesto para el control de cada motor paso a paso unipolar [9].

3.2.5. Diagramas de los circuitos y diseño de las placas PCB

La alimentación de los motores y el control de los mismos se encuentran separados, por ello se utilizan dos reguladores de voltaje, uno para bajar el voltaje de entrada a nivel de potencia 9VDC y otro para nivel TTL de control 5VDC.

Para los controladores de los motores se hicieron placas PCB con el fin de facilitar la conexión y tener un trabajo profesional. Los circuitos y PCBs fueron diseñados en el software Proteus. Se utilizó borneras y conectores de tipo espada para una conexión segura y confiable. Las siguientes figuras muestran los circuitos los drivers de control y el diseño de los PCBs

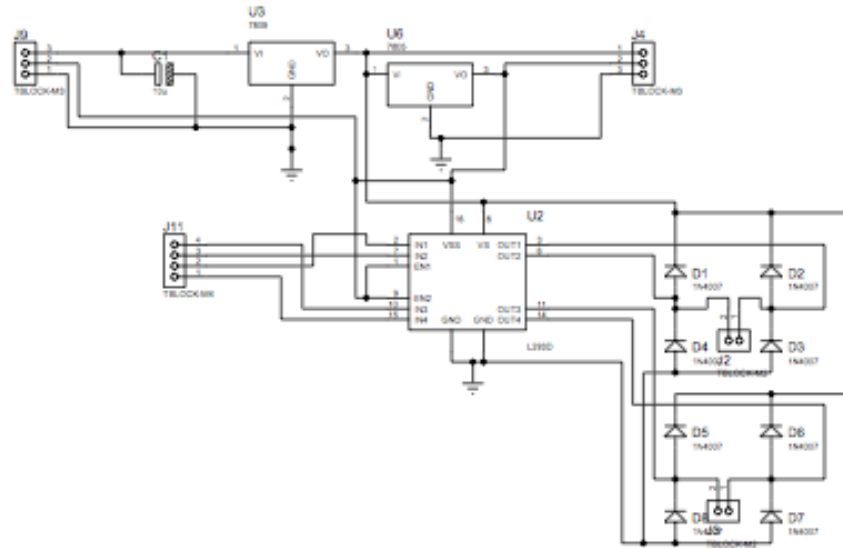


Figura 21. Circuito controlador motores DC.

Creado por: Grupo diseñador

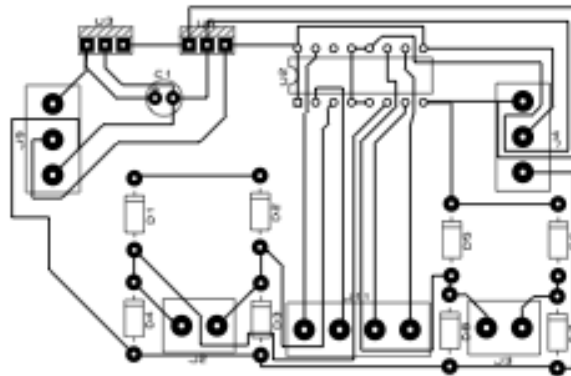


Figura 22. PCB del controlador motores DC

Creado por: Grupo diseñador

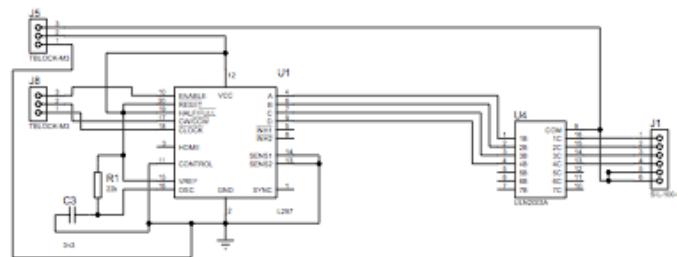


Figura 23. Circuito controlador del motor paso a paso.

Creado por: Grupo diseñador

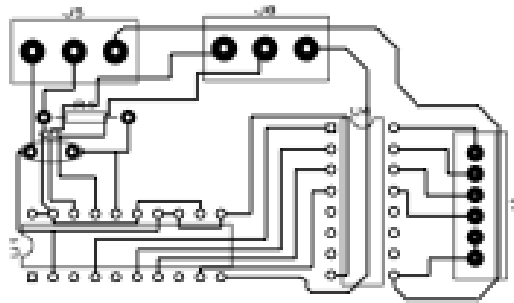


Figura 24. PCB del controlador del motor paso a paso.

Creado por: Grupo diseñador

3.2.6. Descripción del Cableado

Para las entradas de los sensores se utilizó el común en 24VDC del PLC. Uno de los terminales de los sensores tipo reed switch se conectan a las entradas y el otro terminal de los sensores se conecta a tierra (GND). La descripción de las entradas del PLC se la menciona en la siguiente sección.

Adicionalmente se utilizaron 3 resistencias pull-down para las entradas TTL al PLC que se conectan a la tarjeta de adquisición de Labview, la cual nos indica que color encontró. Se utilizó una fuente externa de potencia, ya que el PLC solo realiza trabajos de control.

4. CAPÍTULO IV - PROGRAMACIÓN Y AUTOMATIZACIÓN

En la sección anterior se describieron los elementos mecánicos, eléctricos y electrónicos que componen la planta, por lo que se abordaron conceptos que corresponde a la potencia y a los circuitos electrónicos que permiten el control de los actuadores. En esta sección trata acerca de las señales de control que se envían desde el PLC a los drives que manejan los motores. A continuación lleva a cabo únicamente la descripción del control de la planta, es decir del programa diseñado y cargado en la CPU así como su acción sobre las entradas y salidas digitales del PLC. Esta sección inicia describiendo la programación del sistema de visión que utiliza la planta para reconocer y clasificar los objetos.

4.1. Sistema de Visión

El puente grúa con el fin de cumplir con su objetivo necesita tener un mecanismo mediante el cual pueda distinguir y diferenciar objetos de diferentes colores, en este caso el programa de visión deberá ser capaz de identificar objetos de 3 diferentes colores, y para este caso se ha decidido que sean Rojo, Verde y Azul.

Para la implementación de la parte de visión se creó una aplicación en Labview de tal manera que se pueda identificar de entre estos 3 colores. Para la extracción de la imagen a tiempo real se utilizó una cámara web. Se utilizó Labview debido a su gran versatilidad y facilidad que brinda mediante alguna de sus librerías para manejar y procesar imágenes, además mediante la tarjeta de adquisición de datos DAQ6080 se permite interactuar a tiempo real con los circuitos electrónicos y de esta manera enviar las señales al PLC.

Se seleccionaron estos colores debido a que son los colores primarios, es decir que a partir de estos se pueden crear el resto de colores, además LabView cuenta con filtros RGB, lo cual facilita la identificación específicamente de estos tres colores. La lógica del programa se muestra en la siguiente figura.

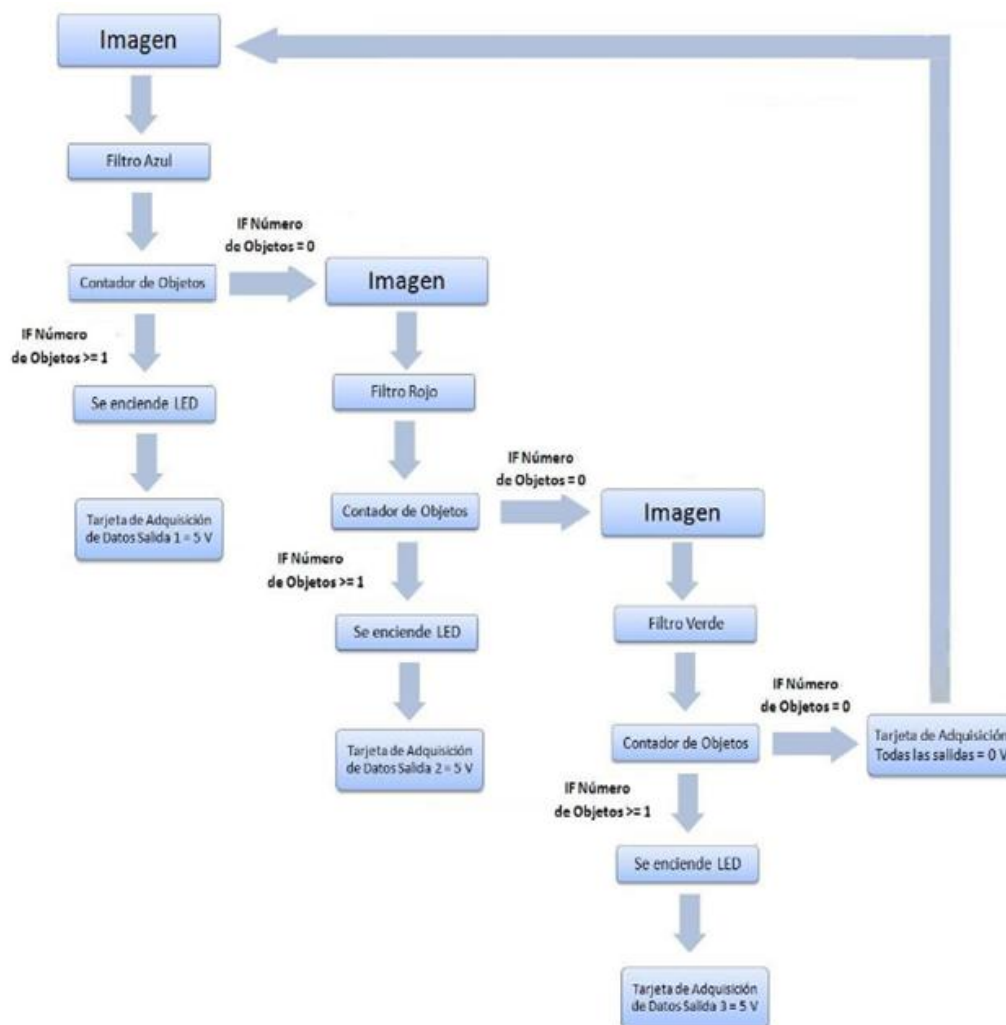


Figura 25. Lógica del programa sistema de visión

Creado por: Grupo diseñador

Mediante una cámara web la imagen, entra primeramente en un filtro para detectar el color azul, si detecta un objeto de este color, enciende una luz de aviso y da una señal de salida en el puerto 1 de la tarjeta de adquisición de datos, si no detecta ningún objeto azul, la imagen pasa por un filtro para detectar objetos de color rojo, realiza el mismo proceso que con el color azul, si detecta un objeto de color rojo da una señal de salida en el puerto 2 de la tarjeta de adquisición de datos, si no encuentra ningún objeto de color rojo, la imagen entra en un filtro verde y es procesada de igual manera que en el caso anterior, en el caso de encontrar un objeto verde se enviaría una señal al puerto 3 de la tarjeta de adquisición de datos, y si no hubiera objeto de color verde todas las salidas en la tarjeta serían 0, con esta lógica además se evita que hayan 2 colores

activados al mismo tiempo, lo cual provocaría una confusión en la lógica del PLC. En la siguiente figura se muestra el diagrama de bloques del programa realizado en Labview.

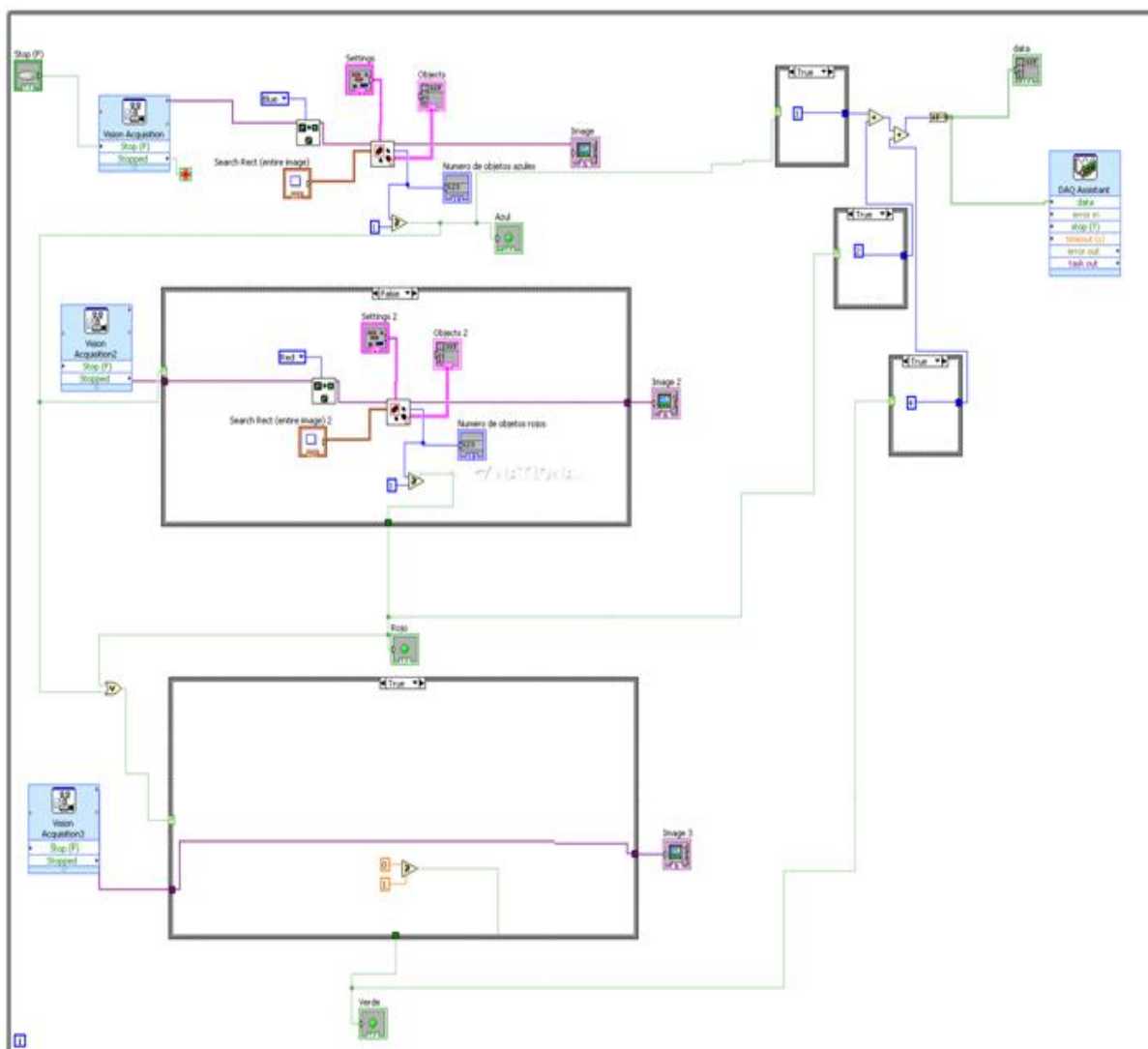


Figura 26. Diagrama de Bloques del programa de identificación de colores en Labview

Creado por: Grupo diseñador

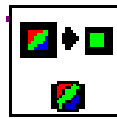
Los principales bloques en el diagrama anterior son:

- Vision Acquisition



Este bloque permite extraer la imagen a ser procesada, en este caso se seleccionara la cámara web como dispositivo, que captura las imágenes y se lo realizará en tiempo real.

- IMAQ ExtractSingleColorPlane VI



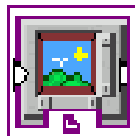
Este bloque permite generar filtros por colores y tonalidades para la imagen extraída con el bloque anterior, en este caso se realizará un filtro para azul, otro para rojo y otro para verde.

- IMAQ Count Object 2 VI



Este bloque permite contar cuantos objetos, del color del filtro aplicado existen en el plano.

- Image



Este bloque permite visualizar la imagen ya procesada (si es el caso) en el panel frontal

- DAQ Assitant Express VI



Este bloque permite conectar la salida con la tarjeta de adquisición de datos, la cual se conectará al PLC.

La siguiente figura muestra el panel frontal del programa, cuando se identifica un objeto de color rojo.

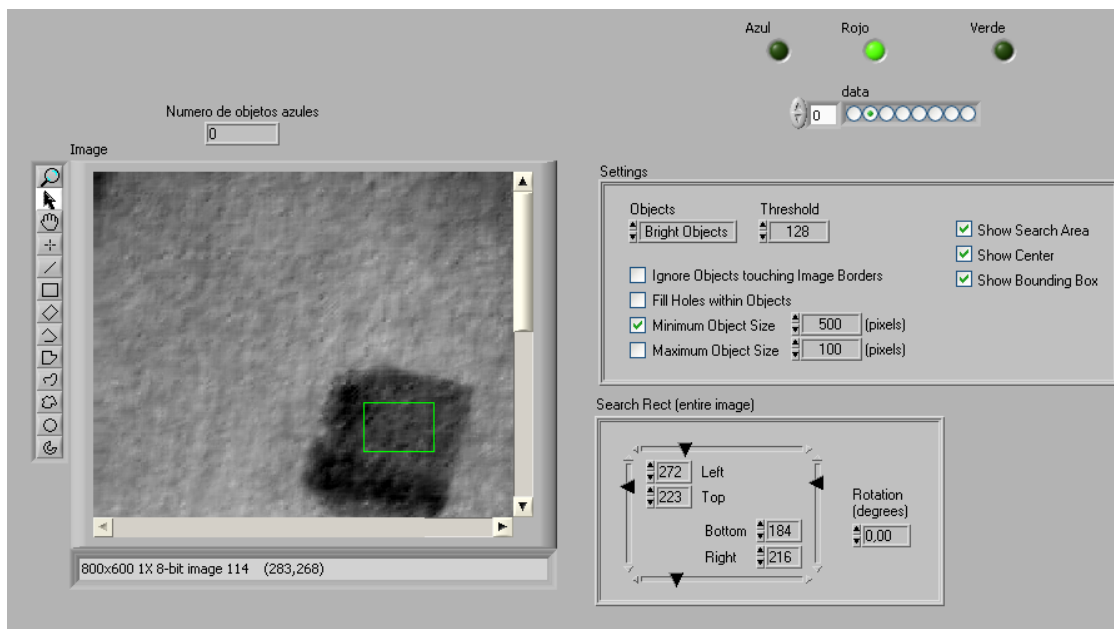


Figura 27. Panel frontal del programa

Creado por: Grupo diseñador

El programa en diseñado en LabVIEW es capaz de reconocer y diferenciar entre los colores rojo, azul y verde mandar una señal de control a la tarjeta de adquisición de datos. Los puertos de la DAQ que a su vez dan valores TTL de voltaje de 0 o 5 V, dependiendo del color que identifique, en sus puertos 1,2 y 3. Estas salidas se conectan directamente al PLC.

4.2. Interfaz de conexión al PLC

Las acciones de la planta y el proceso de clasificación y organización se controlan desde el CPU CB280 del PLC, de esta manera el controlador lógico programable recibe las señales de los sensores en las entradas digitales, con esa información procesa los datos y utilizando las salidas digitales envía las señales de control a los drivers que manejan los actuadores. A continuación se muestra la tabla que especifica la utilidad de las entradas y salidas digitales en el PLC CUSB-22R.

I/O Digital	Función	Descripción
P1	Señal Entrada	Presencia Color Azul
P2	Señal Entrada	Presencia Color Rojo
P3	Señal Entrada	Presencia Color Verde
P11	Señal Entrada	Sensor 0, eje X
P12	Señal Entrada	Sensor 1, eje X
P13	Señal Entrada	Sensor 2, eje X
P39	Señal Entrada	Sensor 3, eje X
P34	Señal Entrada	Sensor 4, eje X
P37	Señal Entrada	Sensor 1, eje Y
P33	Señal Entrada	Sensor 2, eje Y
P35	Señal Entrada	Sensor 3, eje Y
P40	Salida	CW/CCW Stepper 1 (Z)
P41	Salida	Habilita/Deshabilita Stepper 1(Z)
P42	Salida	CW/CCW Stepper 2 (Y)
P43	Salida	Habilita/Deshabilita Stepper 2 (Y)
P44	Salida	Enciende/Apaga Electroimán
PWM Port	Función	Descripción
PWM0/P5	Señal PWM	Control Motor DC 1
PWM1/P6	Señal PWM	Control Motor DC 2
PWM2/p7	Señal PWM	Control Motor DC 2
PWM3/P19	Señal PWM	Señal de Reloj Motor Stepper 1 (Z)
PWM4/P20	Señal PWM	Señal de Reloj Motor Stepper 2 (Y)
PWM5/P21	Señal PWM	Control Motor DC 1

Tabla 2. Descripción de I/O Digitales

Creado por: Grupo diseñador

4.3. Lógica de la Programación

En el sistema de control existen dos condiciones principales, que gobiernan la lógica de la programación. La primera tiene que ver con la clasificación del producto, es decir está relacionada con el tipo de objeto, la segunda condición está ligada a la organización del producto y tiene que ver con la memoria programada en PLC, la cual recuerda las posiciones que están disponibles en el pallet para ubicar los objetos.

La tarea de la planta empieza con el reconocimiento del tipo de objeto según el color, lo cual se lo obtiene mediante el sistema de visión. La tarjeta de adquisición de datos envía una señal lógica que indica el tipo de objeto que está observando. Según esta señal se puede distinguir si se tiene un elemento de color azul, rojo o verde. Es importante recalcar que el objeto se encuentra en un principio en la posición inicial, la cual corresponde a la posición S01 del pallet (la descripción de las posiciones de la grilla se encuentran descritas en la sección 2.1). Dependiendo del color del objeto, se elige una sección en eje Y. Las piezas de color azul serán ubicadas en las posiciones asociadas con Y1, las piezas de color rojo serán ubicadas en las posiciones asociadas con Y2, y las piezas verdes serán ubicadas en las posiciones asociadas con Y3.

A continuación se habilitan los motores DC que mueven el elemento móvil a lo largo del eje X, es aquí cuando entra a funcionar el sistema de memoria que recuerda las posiciones ocupadas y ubica los objetos en los espacios vacíos. La ubicación de los objetos empieza en la posición más lejana del eje X, es decir todas las posiciones asociadas con X4, y se van llenando los espacios hasta que se llenan las posiciones en X1. El sistema de memoria implementado está compuesto por un contador, y varias condiciones, el contador se inicia en 4, y cada vez que pasa por el ciclo de ejecución (que consiste en un proceso desde tomar el objeto hasta ubicarlo en la posición X correspondiente al valor del contador) éste se disminuye en una unidad, y por ello la siguiente vez que se vuelva a ejecutar el ciclo, el valor actual del contador será menor en una unidad. Con esto se permite organizar los productos en las posiciones 4 3 2 1 del eje X y si estas posiciones se terminan se puede programar un control para que no reciba más de estos productos o también se puede reiniciar el conteo a la posición 4 y hacer que se reinicie todo el proceso de organización.

Como se está utilizando tres tipos de objetos diferentes el programa cuenta con tres lazos de ejecución, y con tres sistemas de memoria diferentes.

Para los motores DC el control está hecho por medio de señales PWM. Cada uno de los motores paso a paso requieren una señal de reloj para su control junto con dos señales digitales de salida para habilitar/deshabilitar el motor e indicar la dirección de giro CW/CCW. Con el fin de optimizar el código se utilizaron subrutinas para el movimiento en el eje X, para avanzar, retroceder y parar. También se crearon la subrutina para recoger el producto y para entregar el producto. Estas subrutinas se describen en la siguiente sección.

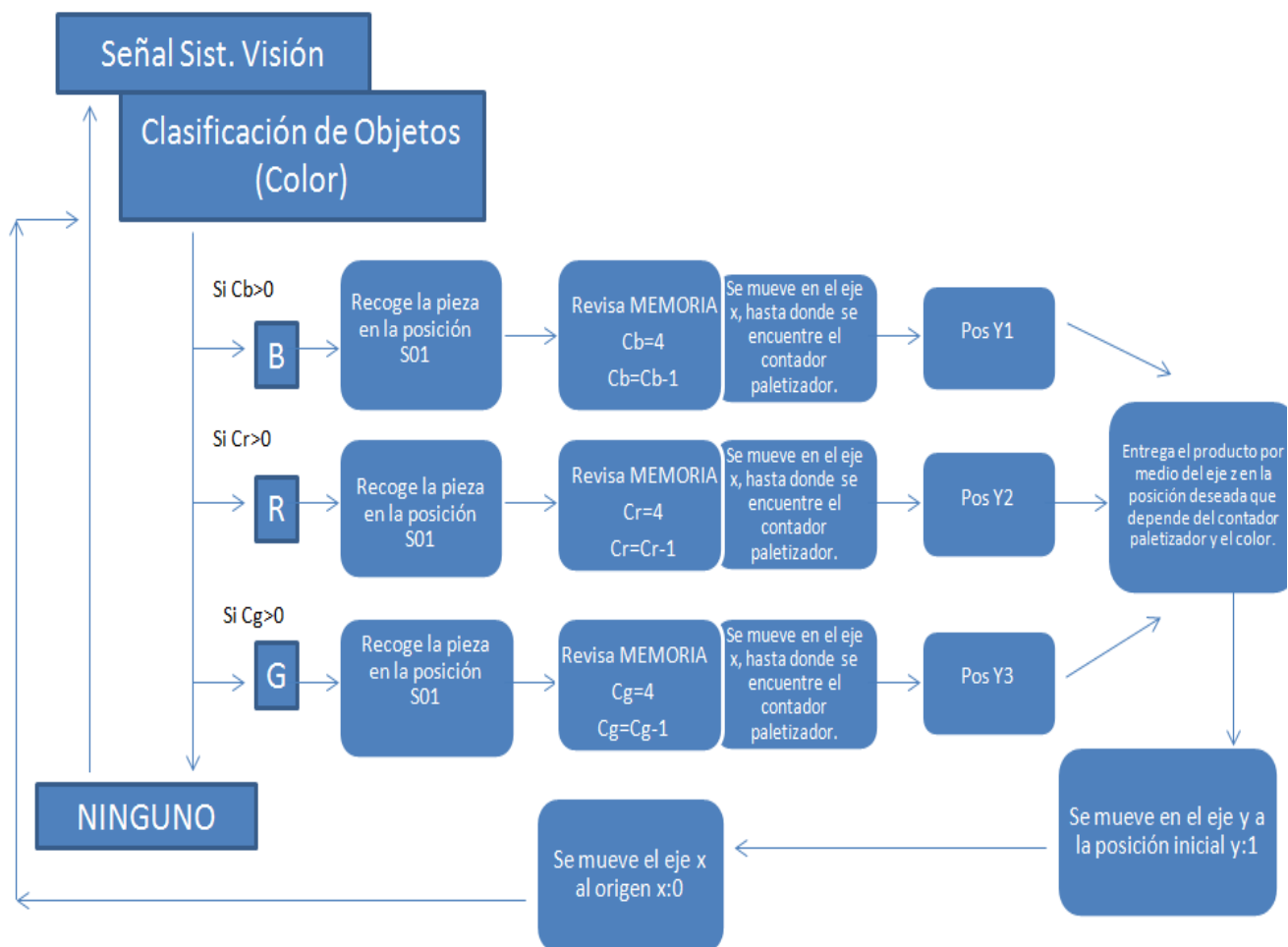


Figura 28. Máquina de estados del programa del PLC

Creado por: Grupo diseñador

4.4. Descripción del Programa

Para la programación del PLC CUBLOC se utiliza el software “Cubloc Studio”, el cual permite escribir código en lenguaje Basic y Ladder. El código que se muestra a continuación ejecuta la acción de clasificación y organización automática y está elaborado en lenguaje Basic, existe otra parte la cual se programó en lenguaje Ladder, pero se utiliza para el sistema SCADA por lo cual no se describe en esta sección. En el código se utilizan los registros de memoria tipo M del CPU CB280, los cuales tampoco se describen en esta sección puesto que son utilizados para monitoreo. Estos registros no interfieren con la lógica del programa que se ya se describió anteriormente y por ello serán tratados en el siguiente capítulo.

4.4.1. Sentencias Especiales

Antes de analizar el código de programación es necesario conocer algunas sentencias especiales que se utilizaron en la programación en lenguaje BASIC. La primera sentencia que se utiliza en el código es “Const Device” la cual que se muestra a continuación:

```
6 Const Device = CB280
```

Esta sentencia especifica el modelo de CPU el cual se está utilizando.

La sentencia “PWM” es la sentencia que utilizamos para activar las señales PWM, el CPU CB280 posee 6 salidas PWM, las cuales se utilizan el control de los motores. Esta sentencia sigue el siguiente formato [6]:

PWM channel, duty, period

Channel: Especifica el canal PWM que se va a utilizar, para el CB280 este valor va del 0 al 5

Duty: Se especifica la duración del ancho pulso (debe ser menor al periodo)

Period: Se especifica la duración del periodo, se puede escribir un valor máximo de 65535.

Mientras más grande sea el número en el periodo la frecuencia va a ser menor. Para conocer la frecuencia a la cual está operando la señal PWM se debe utilizar la relación que nos provee el manual de usuario del CUBLOC [6] que es:

$$\text{Frecuencia} = 2304000 / \text{Duración del Periodo}$$

4.4.2. Inicialización de Variables

El código a continuación muestra la creación e inicialización de las variables tipo byte (0-255), las cuales almacenan el estado de los sensores. Como se puede apreciar en el código a continuación primero se declaran las variables a utilizarse (inicialmente estas variables se establece en 0), luego se inicializan las señales PWM, que sirven de reloj para los motores paso a paso, y como control de los motores DC, a continuación se deshabilitan los dos motores paso a paso mandando una señal lógica alta a los salidas P41, y P3. Por último se deshabilita el electroimán utilizando la salida P44. Es importante notar que en el código se crearon unas variables llamadas tipo bandera, las cuales son útiles al momento de realizar la organización de los productos, lo cual se describe más adelante. También se ejecuta la subrutina *parada*, que mantiene los motores DC apagados, y la cual se describe más adelante. Toda esta inicialización se ejecuta 1 sola vez al encender el PLC.

```

68 Dim azul As Byte
69 Dim rojo As Byte
70 Dim verde As Byte
71 Dim conta As Byte
72 Dim contb As Byte
73 Dim contc As Byte
74 Dim sx0 As Byte
75 Dim sx1 As Byte
76 Dim sx2 As Byte
77 Dim sx3 As Byte
78 Dim sx4 As Byte
79 Dim sy1 As Byte
80 Dim sy2 As Byte
81 Dim sy3 As Byte
82 Dim flagx As Byte
83 Dim flagy As Byte
84 Dim flagz As Byte
85
86 'Inicializacion Senal de Rejor Steppers
87 Low 19 'stepper
88 Pwm 3,5000,10000
89 Low 20 'stepper
90 Pwm 4,5000,10000
91 'Inicializacion Pwm motor DC
92 Low 21
93 Low 5
94 Low 6
95 Low 7
96 parada
97
98 High 41 'Disable Stepper 1
99 Stepper1=0
100 High 43 'Disable Stepper 2
101 Low 44 'bobina desactivada
102 OnBobina=0
103 High 45 'comun sensores tierra
104
105 azul = 0
106 rojo = 0
107 verde = 0
108 sx0 = 0
109 sx1 = 0
110 sx2 = 0
111 sx3 = 0
112 sx4 = 0
113 sy1 = 0
114 sy2 = 0
115 sy3 = 0
116 flagx = 0
117 flagy = 0
118 flagz = 0
119
120 conta = 4
121 contb = 4
122 contc = 4

```

Figura 29. Inicialización de Variables

Creado por: Grupo diseñador

4.4.3. Subrutinas

4.4.3.1. Parada

En la Tabla 1 de la sección 2 se muestran los estados lógicos de los pines de circuito integrado L293 que gobiernan las acciones de los motores DC, ahí se puede apreciar que la parada de los dos motores se consigue desactivando todas las señales de entrada (nivel lógico bajo). Por ello en el código a continuación se aprecia que ancho de pulso de las señales PWM que se envían es 0, con ello lo que estamos haciendo es mandar un voltaje constante de 0V a los pines de control respectivos del puente H.

```
500 Sub parada()  
501 Stop=1  
502 AdelanteX=0  
503 AtrasX=0  
504 Pwm 5,0,10000 'Motor DC 1  
505 Pwm 0,0,10000  
506 Pwm 1,0,10000 'Motor DC 2  
507 Pwm 2,0,10000  
508 End Sub
```

Figura 30. Parada

Creado por: Grupo diseñador

4.4.3.2. Adelante

El movimiento hacia adelante se consigue con la combinación 1, 0, 1,0, en los pines de control del puente H. Por ello en el código a continuación activa el PWM 5, y desactiva el PWM0 con el fin que el motor DC 1 avance, con la misma lógica se activa el PWM 1, y desactiva el PWM2 para que motor DC 2 gire en la misma dirección que el primero. Con ello se logra que la estructura móvil se mueva hacia adelante. Es importante mencionar que dependiendo del ancho de pulso se controlará la velocidad del motor. Cabe recalcar que cuando mencionamos que se desactiva un PWM en este caso nos referimos a que su ancho de pulso es cero.

```
510 Sub adelante()  
511 Stop=0  
512 AdelanteX=1  
513 AtrasX=0  
514 Pwm 5,4000,10000 'Motor DC 1  
515 Pwm 0,0,10000  
516 Pwm 1,4000,10000 'Motor DC 1  
517 Pwm 2,0,10000  
518 End Sub
```

Figura 31. Adelante

Creado por: Grupo diseñador

4.4.3.3. Atrás

Para hacer que la estructura móvil retroceda simplemente se cambia el estado de los PWM que se utilizaron para avanzar es decir se desactiva el PWM 5, activa el PWM0, desactiva el PWM1, y activa el PWM2.

```
520 Sub atras()  
521 Stop=0  
522 AdelanteX=0  
523 AtrasX=1  
524 Pwm 5,0,10000 'Motor DC 1  
525 Pwm 0,4000,10000  
526 Pwm 1,0,10000 'Motor DC 1  
527 Pwm 2,4000,10000  
528 End Sub
```

Figura 32. Atrás

Creado por: Grupo diseñador

4.4.3.4. Recoger Producto

La subrutina a continuación realiza la tarea de recoger el producto, y es llamada una vez que el actuador final está ubicado en la posición inicial. La subrutina empieza primero estableciendo la rotación del motor paso a paso en Z sentido CW, luego se habilita la acción del motor, y se espera tres segundos hasta que el electroimán baje y alcance el pallet, al final de este tiempo se desactiva la acción del motor. A continuación se activa el actuador y se toma el objeto, posteriormente se cambia la rotación del motor paso a paso a CCW con el fin de elevar el electroimán junto con el objeto, se habilita la acción del motor y se espera asimismo tres segundos hasta que se eleve el objeto, entonces se deshabilita la acción del motor. Al final de esta

subrutina el elemento que se ha recogido se encuentra elevado en la posición inicial listo para ser transportado a su posición final.

```
531 'Recoger Producto
532
533 Sub empz()
534 RecogerProd=1
535 Low 40 'CW Z baja
536 Low 41 'Abilita Stepper 1
537 Stepper1=1
538 Wait (3000)
539 High 41 'Desabilita Stepper 2
540 Stepper1=0
541 High 44 'Prende el electroiman
542 OnBobina=1
543 High 40 'CCW Z sube
544 Low 41 'Abilita Stepper 1
545 Stepper1=1
546 Wait (3000)
547 High 41 'Desabilita Stepper 1
548 Stepper1=0
549 RecogerProd=0
550 End Sub
```

Figura 33. Recoger Producto

Creado por: Grupo diseñador

4.4.3.5. Entregar Producto

Esta subrutina se llama cuando el actuador final se encuentra elevado en la posición final en la que debe depositar el objeto que está sosteniendo. También se puede apreciar claramente que cuando se llama a esta subrutina el electroimán se encuentra encendido. La primera acción que se realiza es especificar la dirección de giro del motor a CW para bajar al actuador junto el objeto, luego se habilita el actuador y se espera tres segundos hasta que el electroimán alcance el pallet, ahí se apaga al electroimán y se suelta el objeto, instantáneamente se cambia la dirección de giro del motor a CCW para subir el actuador final y se habilita el motor para que ejecute esta acción, luego de tres segundo cuando el electroimán se encuentra elevado y es cuando se desactiva la acción del motor.


```
552 'Entregar Producto
553
554 Sub terz()
555 EntregarProd=1
556 Low 40 'CW Z baja
557 Low 41 'Habilita Stepper 1
558 Stepper1=1
559 Wait (3000)
560 High 41 'Desabilita Stepper 1
561 Stepper1=0
562 Low 44 'Apaga Electroiman
563 OnBobina=0
564 High 40 'CCW Z Sube
565 Low 41 'Habilita Stepper 1
566 Stepper1=1
567 Wait (3000)
568 High 41 'Deshabilita Stepper 1
569 Stepper1=0
570 EntregarProd=0
571 End Sub
```

Figura 34. Entregar Producto

Creado por: Grupo diseñador

4.4.4. Clasificación de Objetos

Para la clasificación de los objetos simplemente se guarda el estado de las entradas P0, P1 y P2, en las variables *azul*, *rojo* y *verde* respectivamente. El valor de estas tres variables puede ser 1 o 0 y nunca se va a tener dos variables con un nivel lógico alto debido a la restricción establecida en el sistema visión en la sección 3.1.

```
210 azul = Keyinh(1,100)
211 rojo = Keyinh(2,100)
212 verde = Keyinh(3,100)
```

Figura 35. Clasificación de Objetos

Creado por: Grupo diseñador

4.4.5. Ciclo de Ejecución

Existen tres ciclos de ejecución, uno para cada color. Cada ciclo de ejecución es un proceso en el cual el actuador toma el objeto de la posición inicial y lo deposita en la posición del pallet correspondiente. A continuación se muestra únicamente el código del ciclo de ejecución para los elementos de color rojo, puesto que los otros dos siguen exactamente la misma lógica.

Después de la clasificación el programa entra en uno de los tres ciclos, en cada uno existe el mecanismo de memoria, el cual está compuesto por un contador el cual se inicializa en 4, y va actualizándose cada vez que se entra al ciclo. En este ciclo, primero se ejecuta la subrutina para recoger el producto y luego la subrutina que hace avanzar la estructura móvil, ahí se entra en un el lazo, en donde primeramente se evalúa el estado de cada sensor en X para conocer la posición, a continuación se tienen cuatro condiciones, las cuales hacen la estructura se detenga y se salga del lazo. Las condiciones se ejecutan siempre que el contador coincida con el sensor donde está la estructura, entonces el contador indica en que segmento del eje X en la cual se detiene la estructura para dejar el objeto. A continuación se enciende el stepper 2 que mueve al actuador en Y. Para detener este motor se programa otro lazo el cual se detienen una vez que se alcanza la columna requerida. Posteriormente se llama la subrutina para entregar el producto. Se activa nuevamente el motor para el movimiento en Y pero se cambia de sentido de giro para que el actuador regrese a hasta la primera columna, ahí se deshabilita el motor. Finalmente se retrocede en X hasta alcanzar la posición X0. Al final del ciclo como ya se mencionó únicamente se actualiza el contador para evitar ubicar el objeto en una posición ya ocupada.

```

309 Elseif rojo = 1 And contb > 0 Then
310 'Recoge producto z
311 empz
312
313 'Avanza en X
314 adelante
315
316 Do
317 sx1 = In(12)
318 sx2 = In(13)
319 sx3 = In(39)
320 sx4 = In(34)
321 If sx1 = 1 And contb = 1 Then
322 parada
323 flagx=1
324 Elseif sx2 = 1 And contb = 2 Then
325 parada
326 flagx=1
327 Elseif sx3 = 1 And contb = 3 Then
328 parada
329 flagx=1
330 Elseif sx4 = 1 And contb = 4 Then
331 parada
332 flagx=1
333 End If
334 Loop While flagx = 0
335 flagx = 0
336
337 'avanza eb y
338 Low 42 'cw
339 Low 43 'enable izquierda
340 Stepper2=1
341
342 Do
343 sy2 = In(33)
344 If sy2 = 1 Then
345 High 43 'disable izquierda
346 Stepper2=0
347 flagy = 1
348 End If
349 Loop While flagy = 0
350 flagy = 0
351
352 'Entrega Producto Z
353 terz
354
355 'Retrocede y
356 High 42 'ccw
357 Low 43 'enable izquierda

```

```

359 Stepper2=1
360
361 Do
362 sy1 = In(37)
363 If sy1 = 1 Then
364 High 43 'disable izquierda
365 Stepper2=0
366 flagy = 1
367 End If
368 Loop While flagy = 0
369 flagy = 0
370
371 'Retrocede x
372 atras
373 Do
374 sx0 = In(11)
375 If sx0 = 1 Then
376 parada
377 flagx = 1
378 End If
379 Loop While flagx = 0
380 flagx = 0
381
382 contb = contb - 1

```

Figura 36. Ciclo de Ejecución

Creado por: Grupo diseñador

5. CAPÍTULO V - SCADA

En esta sección se describe la implementación del sistema de supervisión y control de la planta. El SCADA indica en tiempo real y de manera visual el estado del proceso de clasificación y organización de los elementos. Este sistema es capaz de informar la posición del actuador final, además se pueden programar alarmas en caso de requerirse, historiales, entre otras funcionalidades altamente requeridas para la una administración de la planta. La implementación del sistema SCADA se la realizó con el fin de tener un monitoreo permanente del estado de la misma, así como también un control parcial sobre el proceso de organización.

El comportamiento de la planta se lo aprecia a través de un HMI donde el operador es capaz recibir información importante como por ejemplo qué sensores se encuentran encendidos, las posiciones del pallet que están ocupadas, las acciones que están llevando a cabo los actuadores, entre otras cosas más. El SCADA además está compuesto de alarmas, que indican cuando el pallet está lleno, acciones de mando como vaciar el pallet, también se lleva la estadística de cuantas veces se ha vaciado el pallet.

5.1. Conexión Física

El PLC CUSB-22R, de la marca CUBLOC que se está utilizando para realizar este proyecto tiene dos canales de comunicación serial RS-232. El canal 0 maneja niveles de voltaje de +/- 12V. El canal 1 en cambio posee dos tipos de salidas, una con niveles de voltaje TTL (5V/0V) y otra con niveles de voltaje +/- 12V. En este proyecto la comunicación se la realiza mediante RS-232 utilizando el protocolo de comunicación Modbus, el cual puede ser utilizado únicamente en el canal 1 según lo indica el Manual de usuario del CUBLOC [6]. Así en este proyecto se utiliza el canal 0 únicamente para descargar el programa al PLC y el canal 1 para realizar el monitoreo y la escritura en la memoria del CPU.

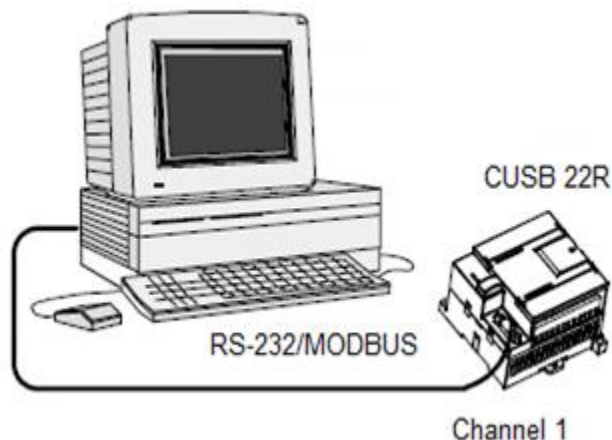


Figura 37. Conexión PLC CUSB-22R, Protocolo Modbus sobre RS-232 en canal 1

Creado por: Grupo diseñador

5.2. Protocolo MODBUS

Modbus es un protocolo desarrollado por MODICOM (actualmente Telemecanique) y utilizado ampliamente en la industria sobretodo en redes de control. Por lo general tiene como red Ethernet o RS-232. En este caso se utiliza RS232. En Modbus se maneja una comunicación del tipo maestro-esclavo. El maestro es quien ordena y pregunta, el esclavo únicamente se limita a responder y ejecutar las acciones. Cada esclavo tiene una dirección única. Modbus maneja dos tipos de transmisión, ASCII y RTU. En este proyecto se utiliza Modbus RTU con control de redundancia cíclica CRC (Cyclic Redundancy Check). La comunicación se realiza enviando y recibiendo unidades “frames”. Cada uno de estos frames, contienen la dirección del equipo, la función Modbus a utilizarse, lo cual se detallara más adelante, los datos, y el control de error. El formato del frame de datos que utiliza Modbus RTU es el que se muestra a continuación, donde se utilizan 4 bytes de separación entre el inicio y el final.

START	SLAVE ADR	FUNCTION	DATA	CRC	END
T1-T2-T3-T4	1 Byte	1 Byte	N Bytes	2 Byte	T1-T2-T3-T4

Figura 38. Formato de Frame Modbus RTU [6]

Modbus se basa en un sistema de direcciones para poder acceder a la memoria de los equipos conectados a la red. Para ello cada fabricante de dispositivos de control que soportan Modbus

provee una tabla de equivalencias, por lo que no se puede hablar de una equivalencia única. En esta sección se hace especial énfasis en las equivalencias con el PLC CUBLOC, sin embargo, la mayoría de ideas expuestas es válida para PLCs de diferentes fabricantes.

A continuación se detallan las funciones Modbus que utiliza el PLC. Estas funciones no son más que lectura y escritura, pero tienen valores diferentes debido a que dependen del tipo de variables que van a afectar, es decir: entradas o salidas, análogas o digitales, registros de memoria del PLC, dimensión de dichos registros, entre otros.

Código Función	Función
01	Lee bits
02	Lee bits
03	Escribe registros (word)
04	Escribe registros (word)
05	Escribe 1 bit
06	Escribe 1 registro (word)
15	Escribe bits múltiples
16	Escribe registros (words) múltiples

Tabla 3. Funciones de Modbus que soporta PLC CUBLOC [6]

En el programa interno del PLC se debe ubicar las variables y sus respectivos espacios de memoria. Una vez que se han ubicado las variables requeridas para la interfaz de usuario, se sigue la convención para escribir las direcciones del PLC como direcciones Modbus tal como se especifica en la tabla siguiente.

Dirección Modbus		Registros CB280	
<i>Bit Units (1 bit)</i>			
DEC	HEX	Nombre del Registro	Rango
0000	0000H	P (Input/Output Register)	P0-P127
4096	1000H	M (Internal Register)	M0-M511
8192	2000H	No Utilizado	-
12288	3000H	No Utilizado	-
16384	4000H	F (Special Register)	F0-F127
<i>Word Units (16 bits)</i>			
DEC	HEX	Nombre del Registro	Rango
20480	5000H	T (Timer)	T0-T99
24576	6000H	C (Counter)	C0-C49
28672	7000H	D (For Storing Data)	D0-D99
32768	8000H	WP (Register P Word Acces)	WP0-WP7
36864	9000H	WM (Register M Word Acces)	WM0-WM31
40960	0A000H	WF (Register F Word Acces)	WF0-WF31

Tabla 4. Correspondencia entre direcciones Modbus y registros del CPU CB280

Creado por: Grupo diseñador

En el caso de la aplicación desarrollada se utilizan los registros tipo P, M, y C por lo que el rango de direcciones Modbus de interés está entre 1-127, 4096-4607 y 24576-246676 (en base decimal). En las siguientes secciones se detallan las direcciones y equivalencias utilizadas en el proyecto.

5.3. Configuración de la Comunicación

Antes de iniciar la programación en cualquiera de los software que se manejan a nivel del ordenador LabVIEW o Indusoft, es necesario incluir variar líneas de código en el programa del PLC CUBLOC con el fin de establecer los parámetros de comunicación que se van a manejar.

5.3.1. Programación en el PLC

Como ya se lo ha mencionado al PLC CUBLOC se lo puede programar tanto en Lenguaje Basic como Ladder. Para tener una comunicación adecuada entre el ordenador y el PLC se debe programar asimismo en ambos lenguajes de programación. En Basic se configura inicialmente

los parámetros básicos la comunicación, como la dirección y la velocidad a la que transmiten los datos, en cambio en Ladder se deben especificar las salidas que se van a leer, puesto que la lectura y escritura de datos se la realiza sobre el programa Ladder.

Lo primero que se realiza en la configuración de la comunicación es la activación del puerto de comunicación que se va a utilizar en el PLC, anteriormente ya se mencionó que únicamente el canal 1 soporta el protocolo Modbus. Utilizando el comando “**Opencom**” se habilita la comunicación RS-232. Esta sentencia sigue el siguiente formato [6]:

Opencom channel, Baudrate, protocol, recvSize, sendSize

Channel: Especifica el canal (puerto serial) RS-232 que se va a utilizar. Para el PLC CUSB-22R, se puede elegir el canal 0 o 1, y ya se ha explicado que se debe seleccionar la segunda opción.

Baudrate: Es la velocidad de transmisión de datos, en este proyecto se utiliza una velocidad de transmisión de 19200 bits/segundo.

Protocol: Este parámetro configura el protocolo de comunicación que se va a utilizar a través del canal seleccionado, así este parámetro establece los bits de datos, la paridad, y el número de bits de stop. El valor “3” es configura una comunicación con 8 bits de datos, paridad ninguna, y 1 bit de stop. Para seleccionar otra configuración posible se debe revisar el manual de usuario del CUBLOC.

RecvSize: Es el Tamaño del buffer de recepción (máx. 1024). Se utiliza 50 para el proyecto

SendSize: Es el tamaño del buffer de envío (máx. 1024). Se utiliza 50 para el proyecto

La declaración utilizada en este proyecto para lograr la comunicación se muestra a continuación:

```
8 Opencom 1,19200,3,50,50
```

Para establecer el protocolo Modbus y definir que el PLC este en modo esclavo se utiliza la sentencia “Set Modbus”. Esta sentencia sigue el siguiente formato [6]:

Set Modbus mode, slaveAddres, returnInterval

Mode: Este parámetro establece si se utiliza Modbus ASCII (0), o Modbus RTU (1)

SlaveAddres: Se especifica una dirección para el PLC actúe como esclavo (1 a 254)

ReturnInterval: Es el tiempo de espera para responder al maestro. (1 a 255)

En este proyecto se utilizó la siguiente declaración:

```
9 Set Modbus 1,1,20
```


Debido a que en este PLC la comunicación se la realiza en combinación con lógica Ladder, se debe definir todos los pines de entrada que son utilizados mediante la sentencia **“UsePin”**

```
11 Usepin 1, In 'Entrada Color 1
12 Usepin 2, In 'Entrada Color 2
13 Usepin 3, In 'Entrada Color 3
14 Usepin 11, In 'Sensor X0
15 Usepin 12, In 'Sensor X1
16 Usepin 13, In 'Sensor X2
17 Usepin 39, In 'Sensor X2
18 Usepin 34, In 'Sensor X4
19 Usepin 37, In 'Sensor Y1
20 Usepin 33, In 'Sensor Y2
21 Usepin 35, In 'Sebsor Y3
```

Figura 39. Declaración Pines de Entrada a ser utilizados en Ladder

Creado por: Grupo diseñador

Debido a que se van a usar registros de memoria auxiliares tipo M, para lectura y escritura de los datos se debe utilizar un alias con el fin de utilizar estos registros.

```

23 Alias M11=VarEncendido
24 Alias M12=completo1
25 Alias M14=completo2
26 Alias M16=completo3
27 Alias M13=ResA
28 Alias M15=ResB
29 Alias M17=ResC
30 Alias M18=RecogerProd
31 Alias M19=EntregarProd
32 Alias M20=AdelanteX
33 Alias M21=AtrasX
34 Alias M22=Stop
35 Alias M23=OnBobina
36 Alias M24=Pos11
37 Alias M25=Pos12
38 Alias M26=Pos13
39 Alias M27=Pos14
40 Alias M28=Pos21
41 Alias M29=Pos22
42 Alias M30=Pos23
43 Alias M31=Pos24
44 Alias M32=Pos31
45 Alias M33=Pos32
46 Alias M34=Pos33
47 Alias M35=Pos34
48 Alias M36=Stepper1
49 Alias M37=Stepper2
50 Alias M50=VarRestA
51 Alias M51=auxiliarA
52 'M52 Reset Contador0
53 Alias M53=VarRestB
54 Alias M54=auxiliarB
55 'M55 Reset Contador1
56 Alias M56=VarRestC
57 Alias M57=auxiliarC
58 'M58 Reset Contador2

```

Figura 40. Declaración registros tipo M con su correspondiente alias

Creado por: Grupo diseñador

La última sentencia requerida en Basic y que permitirá el monitoreo y la escritura de datos es “Set Ladder On”, la cual hace que se ejecute la programación en Ladder al mismo tiempo que se ejecuta el programa en Basic.

```
60 Set Ladder On
```

En programa en Ladder es necesario incluir las entradas digitales (registros tipo P) y los registros tipo M a manera de contactores, con el fin de poder leer el estado de dichos registros desde el ordenador, esto se lo realiza ya que la comunicación se sostiene a través del programa en Ladder.

Todos los registros utilizados en Ladder pueden ser configurados para lectura y escritura, mas no en Basic.

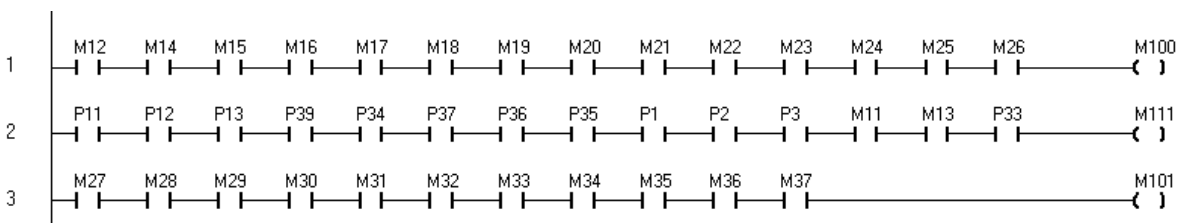


Figura 41.- Registros para lectura y escritura utilizados como contactores.

Creado por: Grupo diseñador

5.3.2. Programación del Software

Aquí se va mostrar la configuración del software para establecer comunicación con el PLC. Existen varios software los cuales se pueden utilizar para leer y escribir registros en la memoria del PLC. En este trabajo se muestra la configuración de la comunicación en dos programas distintos, los cuales son muy utilizados para el diseño de un SCADA. Aunque el protocolo de comunicación es el mismo, cada software requiere una configuración especial y propia. De esta manera lo que busca es dejar una guía para establecer la comunicación entre un PLC CUBLOC y cualquiera de estos dos programas.

5.3.2.1. Consideraciones Generales

En la programación del software se requerirán las direcciones Modbus de los registros utilizados, la Tabla 5 muestra estos registros junto con a su dirección MODBUS.

Registros P	Tag/Alias	Dirección Modbus	Megistros M	Tag/Alias	Dirección Modbus
P1	Azul	2	M11	VarEncendido	4108
P2	Rojo	3	M12	Completo1	4109
P3	Verde	4	M13	ResA	4110
P11	X0	12	M14	Completo2	4111
P12	X1	13	M15	RestB	4112
P13	X2	14	M16	Completo3	4113
P39	X3	40	M17	RestC	4114
P34	X4	35	M18	RecogerProd	4115
P37	Y1	38	M19	EntregarProd	4116
P33	Y2	34	M20	AdelanteX	4117
P35	Y3	36	M21	AtrasX	4118
			M22	Stop	4119
Registros C	Tag/Alias	Dirección Modbus	M23	OnBobina	4120
C0	Contador Azul	24577	M24	S11	4121
C1	Contador Rojo	24578	M25	S12	4122
C2	Contador Verde	24579	M26	S13	4123
			M27	S14	4124
			M28	S21	4125
			M29	S22	4126
			M30	S23	4127
			M31	S24	4128
			M32	S31	4129
			M33	S32	4130
			M34	S33	4131
			M35	S34	4132
			M36	Stepper1	4133
			M37	Stepper2	4134
			M50	VarRestA	4147
			M51	auxiliarA	4148
			M53	VarRestB	4150
			M54	auxiliarB	4151
			M56	VarRestC	4153
			M57	auxiliarC	4154

Tabla 5.- Dirección Modbus de los espacios de memoria utilizados en el PLC CUBLOC

Creado por: Grupo diseñador

5.3.2.2. Indusoft

Para la establecer comunicación entre el programa Indusoft y el PLC CUSB-22R, lo primero que debemos hacer es seleccionar el driver de comunicación adecuado. Para RS-232/Modbus el driver de Indusoft que maneja este tipo de comunicación es En la figura siguiente se muestra esta opción en el programa.

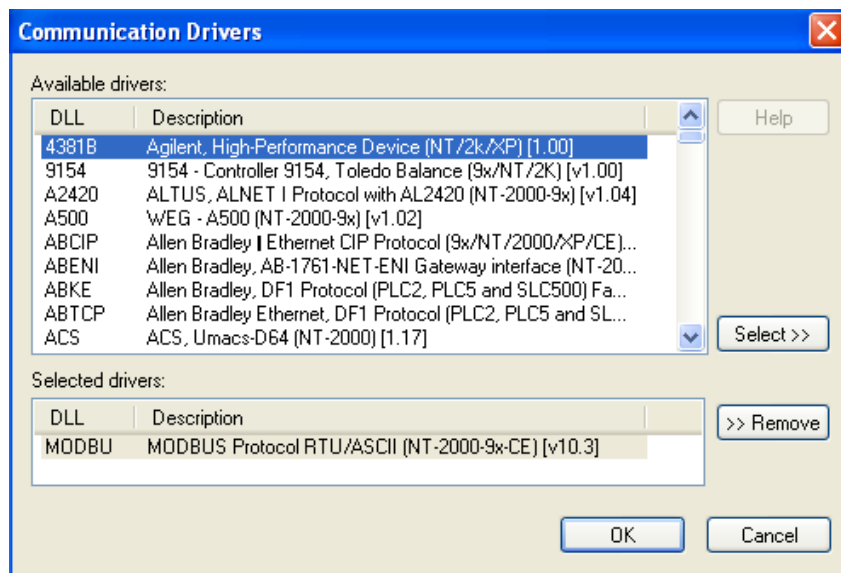


Figura 42.- Selección del driver de comunicación [16]

Una vez definido el driver el programa se deben configurar los parámetros de la comunicación. En la carpeta MODBU, que se crea como subcarpeta de Driver, se da clic derecho y se selecciona la opción *Settings*, para configurar los parámetros de la comunicación. Estos parámetros son los mismos que se establecieron al momento de programar el PLC y los cuales se describen en la sección 4.3.1. En figura a continuación se puede apreciar que los parámetros de comunicación corresponden a los establecidos en la sección 4.3.1. Se debe especificar también el puerto COM correspondiente. En este caso se está utilizado el COM3.

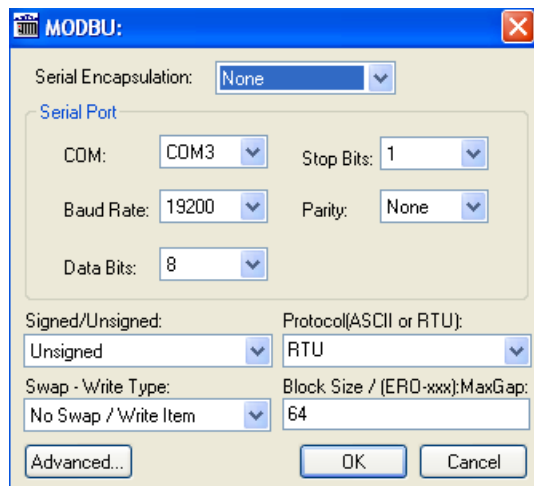


Figura 43. Propiedades de comunicación MODBUS [16]

Una vez establecidos las propiedades de la comunicación se crea automáticamente una interfaz llamada *Main Driver Sheet*, que es en dónde se administran las tareas de la comunicación. Lo primero que debemos realizar en esta hoja es la asignación de las etiquetas RC, RS, WC, WS a los campos *Read Completed*, *Read Status*, *Write Completed* y *Write Status* respectivamente así como se muestra en la Figura 44.

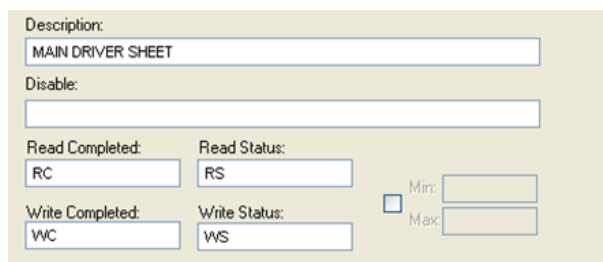


Figura 44. Configuración *Read Completed*, *Read Status*, *Write Completed* y *Write Status*

Creado por: Grupo diseñador

El siguiente paso que se debe realizar es la asignación de las etiquetas mediante el direccionamiento de los registros del PLC CUBLOC. Debido a que se está trabajando con el protocolo Modbus las direcciones de los espacios de memoria se deben escribir utilizando este estándar. La Tabla 5 muestra los espacios de memoria utilizados para este proyecto, y la dirección Modbus asociada. La asignación de etiquetas requiere configurar el campo *Station*, donde se coloca la dirección del esclavo asignada al PLC, la cual se estableció como 1 en el comando “Set Modbus” al momento de realizar la programación en el PLC (Ver sección 4.3.1).

En el campo *I/O Address* se escribe la dirección Modbus junto a un prefijo que hace referencia a las funciones Modbus asociadas. La Figura 45 muestra este formato.

Prefijo	Tipo de registro	Funciones Modbus Asociadas	Notación
0X	bit	1,5,15	0X: xxxxx
1X	bit	2	1X: xxxxx
3X	word	4	3X: xxxxx

Figura 45. Formato Direccionamiento MODBUS en Indusoft

Creado por: Grupo diseñador

Se debe también configurar el campo *Action*, para leer o escribir un los registros del PLC. En el campo *Scan*, se fija en “Always” para tener una actualización constante de los valores desde y hacia el PLC. A continuación se muestra el *Main Driver Sheet* donde están configuradas y asignadas todas las etiquetas que se necesitan en el HMI.

MODBU - MAIN DRIVER SHEET

Description:

Disable:

Read Completed: Read Status:

Write Completed: Write Status:

Min:
 Max:

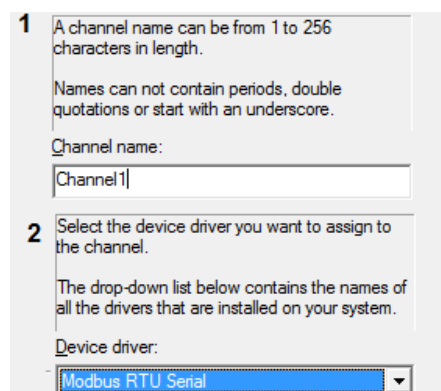
	Tag Name	Station	I/O Address	Action	Scan	Div	Add
1	Azul	1	1X:2	Read	Always		
2	Rojo	1	1X:3	Read	Always		
3	Verde	1	1X:4	Read	Always		
4	X0	1	1X:12	Read	Always		
5	X1	1	1X:13	Read	Always		
6	X2	1	1X:14	Read	Always		
7	X3	1	1X:40	Read	Always		
8	X4	1	1X:35	Read	Always		
9	Y1	1	1X:38	Read	Always		
10	Y2	1	1X:34	Read	Always		
11	Y3	1	1X:36	Read	Always		
12	VarEncendido	1	0X:4108	Read+Write	Always		
13	Completo1	1	1X:4109	Read	Always		
14	ResA	1	0X:4110	Read+Write	Always		
15	Completo2	1	1X:4111	Read	Always		
16	RestB	1	0X:4112	Read+Write	Always		
17	Completo3	1	1X:4113	Read	Always		
18	RestC	1	0X:4114	Read+Write	Always		
19	RecogerProd	1	1X:4115	Read	Always		
20	EntregarProd	1	1X:4116	Read	Always		
21	AdelanteX	1	1X:4117	Read	Always		
22	AtrasX	1	1X:4118	Read	Always		
23	Stop	1	1X:4119	Read	Always		
24	OnBobina	1	1X:4120	Read	Always		
25	S11	1	0X:4121	Read+Write	Always		
26	S12	1	0X:4122	Read+Write	Always		
27	S13	1	0X:4123	Read+Write	Always		
28	S14	1	0X:4124	Read+Write	Always		
29	S21	1	0X:4125	Read+Write	Always		
30	S22	1	0X:4126	Read+Write	Always		
31	S23	1	0X:4127	Read+Write	Always		
32	S24	1	0X:4128	Read+Write	Always		
33	S31	1	0X:4129	Read+Write	Always		
34	S32	1	0X:4130	Read+Write	Always		
35	S33	1	0X:4131	Read+Write	Always		
36	S34	1	0X:4132	Read+Write	Always		
37	Stepper1	1	1X:4133	Read	Always		
38	Stepper2	1	1X:4134	Read	Always		
39	VarRestA	1	0X:4147	Read+Write	Always		
40	auxiliarA	1	0X:4148	Read+Write	Always		
41	VarRestB	1	0X:4150	Read+Write	Always		
42	auxiliarB	1	0X:4151	Read+Write	Always		
43	VarRestC	1	0X:4153	Read+Write	Always		
44	auxiliarC	1	0X:4154	Read+Write	Always		

Figura 46. - Main Driver Sheet

Creado por: Grupo diseñador

5.3.2.3. NI OPC Servers

En la configuración del NI OPC Servers se debe crear primeramente un canal de comunicación y asignarle un nombre, posteriormente se debe seleccionar el driver que maneja la comunicación del canal. En esta aplicación el driver que maneja el protocolo de comunicación es “Modbus RTU Serial”.

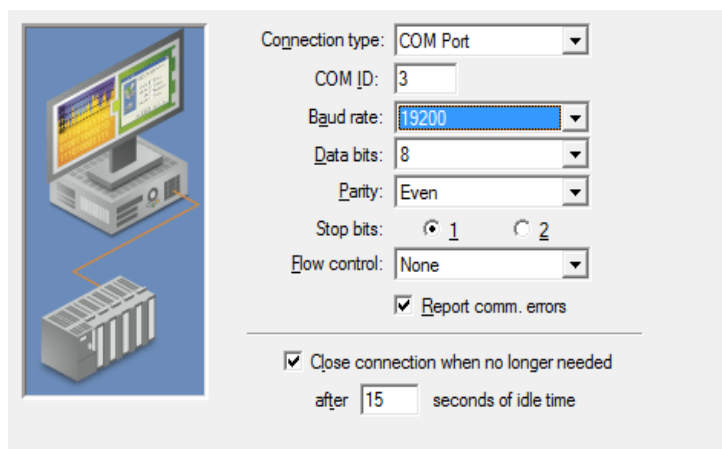


1 A channel name can be from 1 to 256 characters in length.
Names can not contain periods, double quotations or start with an underscore.
Channel name:
Channel1

2 Select the device driver you want to assign to the channel.
The drop-down list below contains the names of all the drivers that are installed on your system.
Device driver:
Modbus RTU Serial

Figura 47. Canal y Driver de la comunicación [16]

En la figura siguiente se muestra los parámetros de la comunicación que se deben definir el software, es necesario asegurarse que éstos coincidan con aquellos que se establecieron en la sección 4.3.1.



Connection type: COM Port
COM ID: 3
Baud rate: 19200
Data bits: 8
Parity: Even
Stop bits: 1 2
Flow control: None
 Report comm. errors
 Close connection when no longer needed
after 15 seconds of idle time

Figura 48. Parámetros de comunicación [16]

En el OPC Server se deben configurar algunos elementos adicionales aparte de las ya señaladas, pero estas no afectan la conexión. Para terminar de establecer conexión con el PLC se debe crear y asignar un dispositivo al canal. De esta manera se define el nombre, el protocolo de

comunicación que maneja el dispositivo con el que se quiere tener comunicación y por último se especifica la dirección de esclavo que se asignó al PLC.

The image shows a configuration window for a device. It is divided into three sections, numbered 1, 2, and 3. Section 1 is labeled 'Device name:' and contains a text input field with 'Device1' entered. Section 2 is labeled 'Device model:' and contains a dropdown menu with 'Modbus' selected. Section 3 is labeled 'Device ID:' and contains a numeric input field with '1' and a dropdown menu with 'Decimal' selected.

Figura 49. Configuración del Dispositivo

Creado por: Grupo diseñador

Una vez establecida la comunicación se crean los tags, y para ello se utilizan las direcciones Modbus de la Tabla 5. Para escribir una dirección en este software de comunicación se debe seguir el formato de que se muestra en la Figura 50. Como se puede apreciar en la grafica el primer número en este formato de direccionamiento indica el tipo de registro que se está utilizando, el cual también identifica a la función Modbus que se utiliza, cabe recalcar que este número no es el mismo que el de la función Modbus pero se observa que existe una relación.

Address	Range	Data Type	Access*
Output Coils [Function Codes (decimal): 01, 05, 15]	000001-065536	Boolean	Read/Write
Input Coils [Function Code (decimal): 02]	100001-165536	Boolean	Read Only
Internal Registers [Function Code (decimal): 04]	300001-365536 300001-365535 300001-365533 3xxxxx.0/1-3xxxxx.15/16***	Word, Short, BCD Float, DWord, Long, LBCD Double Boolean	Read Only

Figura 50. Formato direccionamiento MODBUS en NI OPC Servers [16]

5.3.3. Monitoreo y supervisión, HMI

Hasta hora se ha descrito el sistema electromecánico de la planta y también la programación y automatización de la misma, ahora se tratará la parte del sistema que corresponde al monitoreo y la supervisión, para ello se ha creado un HMI, que permite la interacción entre un operador y la planta.



Figura 51. HMI

Creado por: Grupo diseñador

La Figura 51 muestra el HMI que se utiliza en el SCADA, en donde se puede apreciar todas las funciones de monitoreo y control. Esta interfaz se la realizó con el software LABVIEW. Se tienen dos acciones de control, la primera acción de control que se tiene en el sistema es el encendido o el apagado del proceso, el switch de encendido se encuentra en la parte superior derecha, la segunda acción de control se detalla más adelante.

En cuanto al monitoreo se puede dividir en tres secciones, la primera tiene que ver con la dinámica del sistema, aquí se pueden apreciar las acciones que está realizando la planta, la

segunda es la revisión de la disponibilidad de las posiciones en el pallet, y la tercera está asociada a los contadores. En la Figura 51 se aprecia un cuadro titulado “Posición” en donde se observa la ubicación en la que se encuentra el actuador final, ahí se indica la posición en X, la posición en Y, y también se muestra la posición real en el espacio en la que se encuentra el electroimán. Alrededor de ese cuadro se encuentran varios leds que indican la acción que se está llevando a cabo, se puede conocer si el elemento móvil se está desplazando en X hacia adelante, hacia atrás o se encuentra parado, también se puede conocer si se están realizando movimientos en el eje Y o en Z dependiendo del estado de los leds nombrados respectivamente como Stepper Z y Stepper Y. En la parte superior derecha está un led que indica el estado del actuador final por el que se puede conocer si el electroimán está encendido y por ello sosteniendo un objeto o no. Además se indica cuando la planta está realizando la acción de entregar o recoger el producto. La combinación de todos estos elementos permite conocer todos los movimientos y estados de planta

Hasta el final del capítulo 3 se puede apreciar que el sistema de organización y clasificación está automatizado y puede funcionar bajo el mando del PLC sin intervención humana. En ese capítulo se describió la lógica que el sistema utiliza para diferenciar las fichas y ordenarlas en el pallet pero se dejó abierta una posible estrategia de control. Recordando la lógica de la clasificación vemos que la planta mira las fichas y las ubica en una columna respectiva (eje Y) de acuerdo al color, las fichas son ubicadas desde la posición en el eje X más lejana hasta que se llena la columna, es ahí cuando se puede establecer el control, haciendo que el sistema no reciba más fichas o por otra parte que se reinicie el conteo. El SCADA diseñado realiza esta acción de control, así la planta trabaja automáticamente hasta que se llena una columna, y es cuando se debe ejecutar la acción de mando, cabe recalcar que cada columna correspondiente a cada color es independiente, es decir si se llena la columna del color azul únicamente rechaza las piezas azules pero puede seguir ubicando y organizando las piezas de otros colores. El SCADA está configurado para mostrar una alerta cuando una columna del pallet está completa, es cuando el operador ya sea un hombre o una máquina toma la decisión de vaciar dicha columna. Si el pallet está lleno no se pueden seguir ubicando más objetos, y es lógico que el sistema se detenga y espere hasta que el pallet se vacíe, la acción de vaciar el pallet la puede realizar otro sistema, y el momento que se tienen las posiciones nuevamente disponibles se debe avisar al sistema que se

puede empezar un nuevo ciclo. Se cuenta además con contadores que llevan los registro de cuantas veces se han vaciado las columnas del pallet, con ello se puede conocer la cantidad de productos de cada tipo que se han recogido. En la parte izquierda de la Figura 51 se muestran las posiciones del pallet donde se puede conocer si se ha ubicado o no un objeto en base al sistema de memoria que esta implementado en la programación del PLC. En el medio de la interfaz podemos apreciar los contadores y los leds de alarma que se encienden cuando no hay más espacios disponibles, por último se tienen los botones de *Reset* los cuales deben ser accionados cuando se hayan liberado o recogido las piezas del pallet, para de esta manera limpiar la memoria y agregar una unidad al contador que registra el número de veces que se han vaciado las espacios ocupados por las fichas.

6. Conclusiones y Recomendaciones

6.1. Conclusiones

- La estructura base diseñada cumple con los requisitos mecánicos necesarios para brindar soporte a todos los elementos del sistema y para el correcto funcionamiento de la dinámica de la planta. La estructura móvil diseñada que sostiene al actuador final es capaz de realizar movimientos en el espacio en los tres ejes cartesianos para recoger y posicionar las fichas. El sistema de visión programado diferencia los objetos de acuerdo a su color. Los colores que se pudieron distinguir son los colores básicos, rojo, azul y verde. Las señales de control se transmiten al controlador lógico programable utilizando las salidas de una tarjeta de adquisición de datos. De esta manera la planta diseñada es capaz de ordenar y clasificar los productos según los criterios establecidos en el tercer y cuarto capítulo, y puede ser monitoreada y controlada a distancia mediante un SCADA.

- En cuanto a la programación del PLC CUBLOC se concluyeron varios aspectos relevantes que ayudan a la optimización del código y que son señalados a continuación. Primero, la declaración del tipo de variables que se utilizan en la programación debe ser del tipo *byte* en lugar de *integer*, con ello se utilizan menos recursos de memoria y el procesador ejecuta de manera más eficiente las acciones. Segundo, el uso de subrutinas brinda muchas ventajas puesto que se reducen las líneas de programación, el manejo y la lectura del código resulta más fácil, lo cual es importante si se pretende realizar alguna modificación a la lógica del programa. Por último, debido a que la comunicación entre el PLC CUBLOC y el ordenador se la realiza sobre Ladder se debe programar lo más que se pueda en dicho lenguaje, si esto es posible, y llenar los espacios utilizando lenguaje Basic. En este proyecto se realizó la programación en Basic, debido a la naturaleza del proyecto, por ello para implementar el SCADA se tuvieron que utilizar variables auxiliares tipo *M* que permitan una conexión entre los dos lenguajes dentro de la memoria del PLC.

- Como sabemos el color es una propiedad física que está relacionada con la cantidad de luz reflejada, y debido a que la respuesta del programa a la detección del color varía dependiendo de la cantidad de iluminación que se tenga, es importante tomar en cuenta este punto al momento de realizar la configuración del sistema de visión

- En esta aplicación, la manera más eficiente de ahorrar recursos fue evitando la necesidad de utilizar sensores adicionales para las posiciones de la grilla. Por este motivo se implementó un mecanismo de memoria que recuerda la ubicación y las posiciones ocupadas, por lo que además se ahorra conexiones, cableado y la necesidad de entradas digitales en el PLC.

- Para evitar problemas con el motor paso a paso, la señal de reloj que se obtienen de las salidas PWM del PLC debe ser configurada de tal manera que la frecuencia de los pulsos no sea tan elevada que llegue a causar vibraciones en el motor. El valor de esta frecuencia se consiguió con el método prueba-error y se la especifica en el capítulo 4 (Señal PWM) en la descripción del código del programa.

- Después de un conjunto exhaustivo de pruebas se determinó que para configurar directamente el PLC CUBLOC CUSB-22R y el software Indusoft mediante RS-232/Modbus se tiene que utilizar el driver MODBU MODBUS Protocol RTU/ASCII (NT-2000-9x-CE). En cuanto a la comunicación con LabView se puede utilizar el software NI OPC servers el cual funciona como una interfaz de conexión entre el HMI, creado en LabView, y el PLC.

6.2. Recomendaciones

- Se recomienda la utilización de guías y topes mecánicos para brindar seguridad a la dinámica del sistema evitando que la estructura móvil salga de su camino, y ahorrando la necesidad de utilizar más sensores que requerirían un mayor número de entradas digitales en el PLC, lo cual resulta inconveniente.

- Para el cableado es aconsejable que todas las conexiones estén etiquetadas, con ello se ahorra tiempo al momento de realizar las conexiones si se tuviera la necesidad de desconectar el PLC para tareas de mantenimiento o actualización.

- Con el fin de tener una mayor seguridad y fiabilidad en las conexiones y en la transmisión de las señales de control se recomienda el diseño y la construcción de una placa PCB para los circuitos electrónicos que incluya todos los circuitos integrados encargados de procesar las señales del PLC para el control de los motores, y en donde se realiza la integración de la parte de potencia y control.

- Para lograr una exitosa comunicación con el PLC Cubloc es aconsejable entender tres aspectos fundamentales. Primero, el protocolo Modbus, con esto me refiero a las funciones y al frame de datos, luego se debe conocer los tipos de registros que utiliza el CPU del PLC para poder relacionar la dirección de la memoria interna con las direcciones Modbus equivalentes; y por último se debe programar el PLC de tal manera que se establezca una conexión adecuada, revisando que las propiedades de la comunicación programadas coincidan con las que se configuran con el software de comunicación.

Es aconsejable realizar la configuración del programa de visión de tal manera que el objeto a observar esté acotado en una región específica en el espacio. Además se debe establecer un tamaño mínimo del objeto a reconocer y agregar iluminación artificial, con ello se evitan confusiones en el reconocimiento de color debido al brillo presente en superficie de la grilla.

BIBLIOGRAFÍA

- [1] Ogata Katsuhiko, Ingeniería de Control Moderna, Cuarta edición, Madrid, España, Impreso por Gráficas Rógar, 2007.
- [2] Oppenheim Alan V., Señales y sistemas, Segunda edición, México D.F, México, Impreso por Prentice Hall Hispanoamericana S.A, 1998.
- [3] Hassan Mohamed Houcine, Martínez Rubio Juan Miguel, y Dominguez Montagud carlos, Problemas de Microcontroladores de la familia MCS-51, Valencia, España, Impreso por: Reproval, 2006.
- [4] García Moreno Emilio, Automatización de procesos industriales: robótica y automática, Valencia, España, editorial REPROVAL, 1999
- [5] MatrikonOPC. *Que es un servidor OPC*. Extraído 1 de Enero de 2013
<<http://www.matrikonopc.es/opc-servidor/index.aspx>>
- [6] Comfile Technology Ing. “PLC Embedded computer CUBLOC User Manual”. Version 3.3. 2012-03-26.
- [7] Comfile Technology Ing. “CuSB-22R CUBLOC Integrated Single Body Board, User Manual”.
- [8] Sparkfun. *Micro Metal Gearmotor 100:1*. Extraído el 26 de Noviembre de 2012
< <https://www.sparkfun.com/products/8910>>
- [9] Escuela Politécnica Superior de Alcoy. *Control de motores paso a paso mediante micro controladores*. Extraído el 27 de Noviembre de 2012. <<http://server-die.alc.upv.es/asignaturas/lased/2002-03/MotoresPasoPaso/ Motorespasoapaso.pdf>>

-
- [10] Explainthatstuff. *Reed Switches*. Extraído el 27 de Noviembre de 2012.
<<http://www.explainthatstuff.com/howreedswitcheswork.html>>
- [11] Texas Instruments. L293 - QUADRUPLE HALF-H DRIVERS Datasheet. Dallas, Texas 2005.
- [12] Sprague. Seven Darlington array ULN2003 Datasheet. February 2002.
- [13] CGS-Thomson Microelectronics. L297, L297D Stepper Motor Controllers Datasheet. August 1996.
- [14] Sosa Lili, Señales TTL, 1 de enero del 2012 < www.unicrom.com >
- [15] Bill Drury, Control Techniques Drives and Controls Handbook (2nd Edition) . 2009, Institution of Engineering and Technology
- [16] National Instruments. “Help version 1.192 NI OPC Servers”. Version 5.5. 2012
- [16] National Instruments. Help Labview Version 2012. 2012-11-25.
- [17] Indusoft. Help Indusoft version 7.0. 2012-11-25.
- [19] G. Clarke, D. Reynders. Practical Industrial Data Networks - Design, Installation and Troubleshooting. Newnes 2004
- [20] G. Clarke, D. Reynders. Practical Modern SCADA Protocols. Newnes 2004
- [21] Henry Mendiburu Díaz. *Sistemas Scada*. Extraído el 26 de Noviembre de 2012
< <http://www.galeon.com/hamd/pdf/scada.pdf>>

ANEXOS

CÓDIGO LADDER

Desarrollado por Paul Frutos

