

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ



SECRETARÍA DE
EDUCACIÓN PÚBLICA

SEP

TRABAJO PROFESIONAL

COMO REQUISITO PARA OBTENER EL TITULO DE:

INGENIERO EN ELECTRÓNICA

QUE PRESENTA:

**ANDRÉS EDUARDO FRIAS ESCOBAR
JORGE IVÁN GÓMEZ ZEA**

CON EL TEMA:

**“SISTEMA DE POSICIONAMIENTO DE
OBJETOS POR MEDIO DE PLC’s Y UN BRAZO
DE ROBOT”**

MEDIANTE :

**OPCION X
(MEMORIA DE RESIDENCIA PROFESIONAL)**

TUXTLA GUTIERREZ, CHIAPAS

ENERO 2014

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ



SECRETARÍA DE
EDUCACIÓN PÚBLICA

SEP

TRABAJO PROFESIONAL

COMO REQUISITO PARA OBTENER EL TITULO DE:

INGENIERO EN ELECTRÓNICA

QUE PRESENTA:

**ANDRÉS EDUARDO FRIAS ESCOBAR
JORGE IVÁN GÓMEZ ZEA**

CON EL TEMA:

**“SISTEMA DE POSICIONAMIENTO DE
OBJETOS POR MEDIO DE PLC’s Y UN BRAZO
DE ROBOT”**

MEDIANTE :

**OPCION X
(MEMORIA DE RESIDENCIA PROFESIONAL)**

TUXTLA GUTIERREZ, CHIAPAS

ENERO 2014

Agradecimientos.

Gracias a Dios por darme fuerzas para seguir adelante, por guiarme en el sendero de lo sensato y darme sabiduría en las situaciones difíciles.

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento.

A mis maestros que en este andar por la vida, influyeron con sus lecciones y experiencias en formarme como una persona de bien y preparada para los retos que pone la vida, a todos y cada uno de ellos les dedico cada una de estas paginas de mi tesis.

INDICE GENERAL

CAPITULO I	4
1.1 Introducción	5
1.2 Justificación	6
1.3 Objetivos	7
1.3.1 Objetivo General:.....	7
1.3.2 Objetivos específicos:	7
1.4 Caracterización del área en que participó	8
1.5 Problemas a resolver	10
1.6 Alcances y limitaciones	11
CAPITULO II	12
2.1 Fundamento Teórico	13
2.1.1 Robot CRS.....	13
2.1.2 Programación básica en RAPL-3	16
2.1.3 Controladores Lógicos Programables (PLC) SIEMENS	20
2.1.4 Administrador Simatic STEP-7	26
CAPITULO III	30
3.1 Procedimiento y descripción de las actividades realizadas	31
3.1.1 Banda Transportadora y PLC 315-2DP.....	31
3.1.2 Brazo de Robot	49
3.1.3 Plano Cartesiano	58
Conclusión y Resultados	68
Bibliografía	69
Anexos	70

INDICE DE FIGURAS

FIG. 2.1 ILUSTRACIÓN FÍSICA DE TEACHPENDANT	14
FIG. 2.2 VENTANA DE JERARQUÍA DE PROYECTO.....	26
FIG. 2.3 VENTANA DE CONFIGURACIÓN	27
FIG. 2.4 BASTIDOR DE CONFIGURACIÓN.....	28
FIG. 2.5 ILUSTRACIÓN DE POSICIÓN DE COMPONENTES DEL BASTIDOR EN PLC FÍSICO	28
FIG. 3.1 VENTANA DE PRINCIPAL	33
FIG. 3.2 BASTIDOR	33
FIG. 3.3 VENTANA DE OBJETOS	35
FIG. 3.4 EJEMPLO DE BAHÍA	37
FIG. 3.5 CAMBIO DE BANDA PRINCIPAL A SECUNDARIA.....	37
FIG. 3.6 PISTÓN DE CAMBIO DE BANDA PRINCIPAL A SECUNDARIA	38
FIG. 3.7 ILUSTRACIÓN DE PLATAFORMA DE BAHÍA.....	38
FIG. 3.8 SENSORES INDUCTIVOS.....	38
FIG. 3.9 PALETS (VISTA SUPERIOR).....	39
FIG. 3.10 PALET 1 Y PALET 2 (VISTA INFERIOR)	39
FIG. 3.11 TIPO DE MOVIMIENTOS DEL BRAZO ROBOT.....	50
FIG. 3.12 ILUSTRACIÓN FÍSICA DEL BRAZO DE ROBOT	51
FIG. 3.13 CONFIGURACIÓN JOINT 6	51
FIG. 3.14 GRADOS DE MOVIMIENTO DEL BRAZO DE ROBOT	52
FIG. 3.15 VENTANA PRINCIPAL	53
FIG. 3.16 MENÚ APPLICATION	54
FIG. 3.17 VENTANA EMERGENTE DE CONFIGURACIÓN.....	54
FIG. 3.18 EDITOR DE PROGRAMAS	55
FIG. 3.19 ILUSTRACIÓN FÍSICA DEL PLANO CARTESIANO	58
FIG. 3.20 ILUSTRACIÓN FÍSICA DE SWITCH EN ENTRADA DEL PLANO CARTESIANO.....	59
FIG. 3.21 SISTEMA DE PISTONES RECOLECTOR DE PALETS	59
FIG. 3.22 MOTOR	60
FIG. 3.23 VENTANA PRINCIPAL	61
FIG. 3.24 MENÚ APPLICATION	61
FIG. 3.25 VENTANA EMERGENTE DE CONFIGURACIÓN.....	62
FIG. 3.26 EDITOR DE PROGRAMAS	63

INDICE DE TABLAS

TABLA 3.1 CONFIGURACIÓN DE SALIDAS PLC 315-2DP	34
TABLA 3.2 CONFIGURACIÓN DE ENTRADAS PLC 315-2DP.....	36
TABLA 3.3 CÓDIGO DE PALETS	43
TABLA 3.4 SALIDAS Y ENTRADAS EN RED DEL PLC 315-2DP CON EL 215-2DP	65
TABLA 3.5 SALIDAS Y ENTRADAS EN RED DEL PLC 315-2DP CON EL 226.....	66
TABLA 3.6 BYTES DE SALIDAS DEL PLC 315-2DP	66
TABLA 3.7 BYTES DE SALIDAS DEL PLC 215-2DP	66
TABLA 3.8 PROGRAMACIÓN DE RED PROFIBUS	67

CAPITULO I

1.1 Introducción

En la actualidad dentro de la industria tanto nacional como internacional se ha dado un creciente avance tecnológico en el proceso de automatización, ya que para lograr ser competitivos industrialmente se requiere, a más de un producto de buena calidad reducir los tiempos de producción sin defectos en el producto final y por sobre todo con seguridad para el operario. Es por ello que la implementación de equipos de última tecnología constituye una necesidad dentro de los procesos de producción.

Un concepto que es importante entender en estos momentos es el de automatización ya que de aquí en adelante se estará recurriendo constantemente a esta palabra por ello aquí una pequeña definición.

El término Automatización viene del griego antiguo: guiado por uno mismo, es decir automatización se puede definir como el uso de elementos o sistemas computarizados dedicados a controlar maquinarias y/o procesos industriales substituyendo a operadores humanos.

En el ámbito de la industrialización la automatización es un paso más allá de la mecanización. Considerando que la mecanización proporcionan a los operadores humanos con maquinaria para ayudarles con los requerimientos físicos del trabajo, la automatización reduce la necesidad de requisitos humanos sensorial y mental. Y los procesos y sistemas también pueden ser automatizados.

La automatización hoy en día juega un papel cada vez más importante en la economía mundial y en la experiencia cotidiana. Los ingenieros se esfuerzan por combinar dispositivos automatizados con herramientas matemáticas y de organización para crear sistemas complejos para una amplia rápida expansión de las aplicaciones y las actividades humanas.

Equipos especializados endurecido, conocido como controladores lógicos programables (PLC), se utilizan con frecuencia para sincronizar el flujo de las aportaciones de los sensores y eventos con el flujo de salidas a los actuadores y eventos. Esto nos lleva precisamente a las acciones de control que permiten un estricto control de casi cualquier proceso industrial.

1.2 Justificación

Se realiza este proyecto para desarrollar una secuencia de automatización conformada de tres partes y que tiene como propósito principal el posicionamiento de objetos utilizando PLC's de la familia SIEMENS y Robots CRS.

El proyecto terminado ayudará a los docentes que dan clases en el laboratorio de manufactura avanzada del ITESI, mostrando a los alumnos como se realiza un proceso industrial y como configurar los elementos que componen el laboratorio hasta implementar sus propias secuencias.

La información obtenida de cómo configurar, programar y manejar los componentes del laboratorio, servirá para mantener en buen estado los equipos ya que la mayoría de los estudiantes no tienen la preparación adecuada para manejarlos, y debido a estas limitantes pueden llegar a descomponerlos.

1.3 Objetivos

1.3.1 Objetivo General:

Realizar la programación para la automatización de un sistema el cual esta integrado por tres partes que trabajan en conjunto y así simular un trabajo de proceso industrial de recolección, transportación y clasificación de Palets. Este sistema consta con un plano cartesiano donde se almacenan y recolectan Palets, un Brazo de Robot que sujeta y transporta los palets y una banda transportadora en donde finalmente se clasifican y sacan los palets por la Bahía correspondiente,

1.3.2 Objetivos específicos:

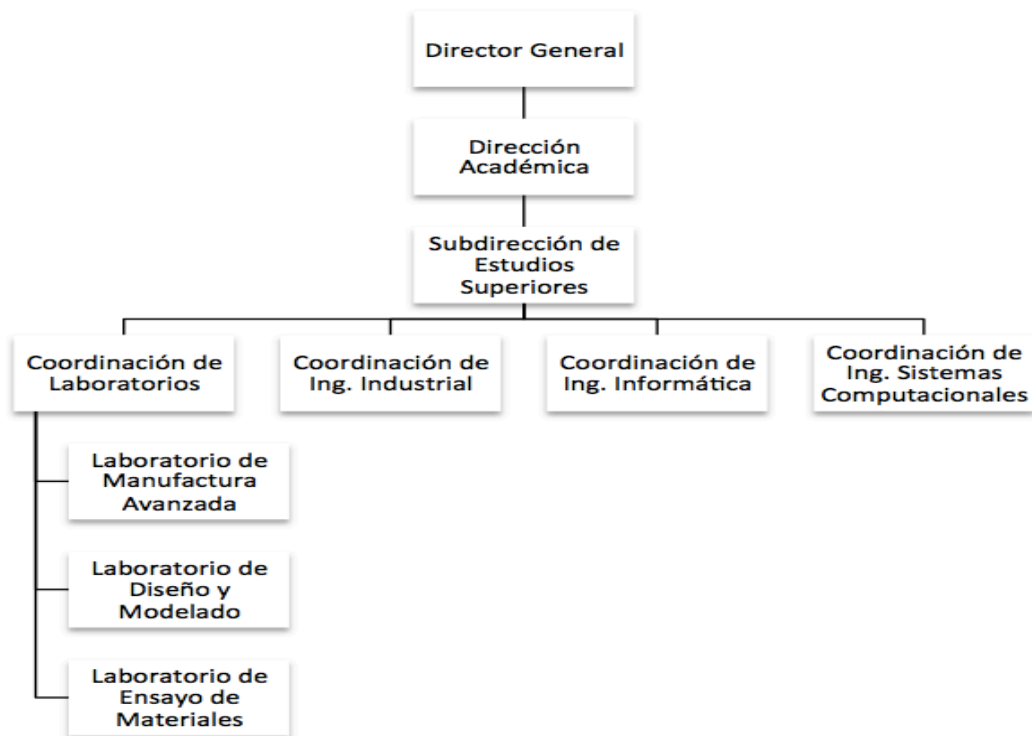
Programar el PLC 315-2DP para la comunicación vía Profibus y el control de una banda transportadora para la clasificación de los Palets.

Programar un Brazo de Robot para interactuar con la Banda Transportadora y un Plano Cartesiano, y realizar diferentes secuencias hasta obtener la mas apropiada para el proyecto.

Programar un Plano Cartesiano, para que proporcione de Palets al Brazo de Robot, realizando diferentes secuencias.

1.4 Caracterización del área en que participó

El Instituto Tecnológico Superior de Irapuato (ITESI) es una institución de educación superior de carácter público, cuyo propósito es servir a la sociedad formando profesionales con elevados conocimientos técnicos, científicos y humanísticos, que los habilite para: Generar riqueza en las cadenas de valor de la actividad económica y social. Convertirse en promotores y agentes de cambio, que mejoren la calidad de vida de la sociedad Fortalecer la democracia, solidaridad, cultura y medio ambiente.



El Departamento encargado del Laboratorio de Manufactura Avanzada es el de Coordinación de Laboratorios el cual se encarga de programar, coordinar, administrar y gestionar las actividades de los laboratorios, verificando que se proporcione tanto al personal docente como alumnos el material, equipo, atención y asesoría que se requiera para la realización de las diversas prácticas, así como vigilar por el buen estado y correcto funcionamiento de los equipos y el mantenimiento necesario del edificio que constituye la dependencia de los laboratorios.

El Laboratorio de Manufactura Avanzada del Instituto Tecnológico Superior de Irapuato, esta encargado de desarrollar proyectos de implementación y adecuación de nuevas tecnologías hacia la industria.

El Laboratorio cuenta con dos módulos de la compañía CRS Robotics y tres PLC's de la marca SIEMENS, los cuales se encargan del control de los centros de maquinado, un Plano Cartesiano, un Brazo de Robot y una Banda Transportadora. También se cuenta con PC's para la programación de los módulos de control al igual que las tuberías neumáticas para las maquinas que lo requieren.

1.5 Problemas a resolver

- Investigar sobre los programas de software Administrador Simatic y Robcom, su funcionamiento y su estructura de programación, y las herramientas con las que cuentan estos software.
- Conocer los componentes, sensores y actuadores con los que cuenta cada sistema del laboratorio.
- Hacer un listado de todas las entradas y salidas con las que cuentan los controladores.
- Realizar un mantenimiento preventivo y correctivo (eléctrico, mecánico y neumático) a las maquinas que lo requieren.

1.6 Alcances y limitaciones

Alcances

Con la ejecución de este proyecto, se desarrollará un sistema de automatización eficiente que cuenta con un proceso de tres fases la recolección, transportación y clasificación de palets.

Limitaciones

En el desarrollo del proyecto de residencia se presentaron las siguientes limitaciones:

- No estar familiarizado con el programa Administrador Simatic, encargado de la programación de PLC's de la familia S7-300.
- Falta de conocimiento de la tecnología de los Robots CRS, tanto configuración física como el software y lenguaje de programación.

CAPITULO II

2.1 Fundamento Teórico

2.1.1 Robot CRS

Los Robots CRS fijan un nuevo estándar para los Robots “humano-escala”, entregando una combinación de gran alcance de velocidad, de confiabilidad, de facilidad de empleo.

El cambio de herramientas dispositivo especializado y la facilidad de variar el movimiento a realizar permiten que, al incorporar al robot en el proceso productivo, sea posible y rentable la automatización en procesos que trabajan con series mas reducidas y gamas mas variadas de productos.

Sus aplicaciones son extensas y variadas, estas varían desde la alimentación de piezas en un proceso, la distribución de las misma, la inspección, el manejo material, el empaquetado, el Paletizado, el mando de calidad, la carga de la maquina, el rociado y la comprobación del producto hasta llegar a formar parte de una célula de enlatado, soldadura o pintura.

Estos Robots serán constituidos por un brazo articulado de seis grados de libertad y de un gran alcance, la unidad controladora (controlador C500C), un programador manual o Teach Pendant, cables con conectores normalizados y una computadora personal.

Controlador C500c

Los modelos de robots de la marca THERMO/CRS cuentan para su control con el controlador C500c el cual se configura y calibra de acuerdo al tipo de Robot que se va a manipular.

En el controlador se almacenan los datos de posiciones y programas, y a través de este se manipula el Robot usando los botones e indicadores de la parte frontal. De igual forma se tiene un módulo de entradas y salidas de propósito general (GPIO).

Con su nombre lo que indica es un puerto de entradas / salidas el cual cuenta con:

- 16 entradas opto aisladas, NPN
- 12 salidas opto aisladas, NPN
- 4 salidas a relevador, con salida normalmente abierta y salida normalmente cerrada, conectadas a línea común .

- Una entrada analógica de 0 a 5 volts.

Teach Pendant

El modo de operación de los Robots es por medio del Teach pendant (ver Fig. 2.1) que funciona como un control directo que permite mover el Robot, enseñar posiciones, crear variables y correr programas. Una vez que la aplicación ya esta indicada, por medio de las variables aprendidas es posible guardar posiciones, iniciar variables, etc.

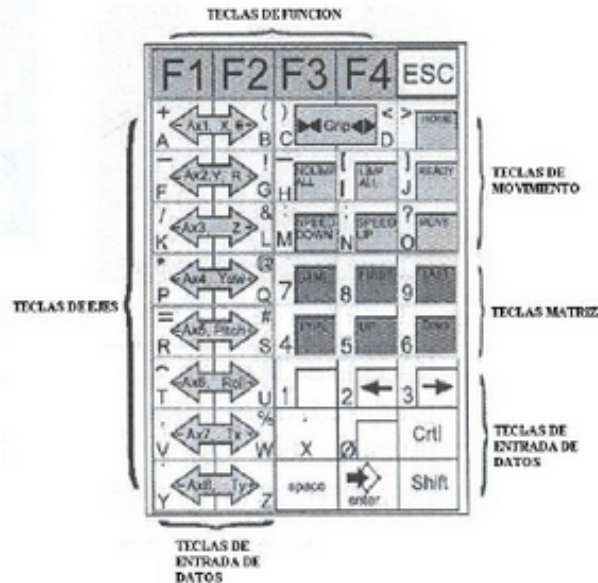


Fig. 2.1 Ilustración física de Teach Pendant

Características:

- Teclado
- Interruptor de seguridad (live-man)
- Paro de emergencia.
- Pantalla
- Sonidos

Teclado

- Teclas función. Se asignan las teclas (F1 a F4), una función correspondiente. Por medio de cada una de ellas se representa un menú diferente dependiendo de la aplicación.
- Teclas de ejes. Se emplean para mover el Robot.

- Teclas de movimiento. Se utilizan para colocar al robot en posición inicial, cambiar la velocidad, elegir punto al que se debe mover, energizar y desenergizar los ejes del motor y abrir y cerrar el Grip.
- Teclas matriz. Se utilizan para seleccionar los elementos de una matriz, así como incrementar o decrementar sus índices.
- Teclas de entrada de datos. Cada tecla de eje, de movimiento y de matriz también cuentan con letras o números. Las letras sirven para crear, nombrar y buscar variables o aplicaciones. Los números sirven para seleccionar un índice de una matriz, crear una variable y especificar su tamaño.

Interruptor de seguridad (Live-man-switch)

Este interruptor cuenta con tres posiciones. Cuando se oprime ligeramente, está habilitado el movimiento. Si se presiona fuertemente o si se suelta está deshabilitado el movimiento. Para manipular el brazo presionar ligeramente el interruptor de seguridad ya que si se quitara la condición de habilitar durante un movimiento provocaría la interrupción de la alimentación del Brazo y el Robot se parará súbitamente.

Paro de emergencia

Interrumpe la alimentación del Robot.

Pantalla

Los diferentes menús son mostrados por medio de la pantalla. Por medio de las teclas función se puede acceder. Por medio de la tecla ESC se va al menú anterior.

2.1.2 Programación básica en RAPL-3

Todos los CRS trabajan con el software ROBCOMM3; este utiliza el lenguaje RAPL-3 (Robotic Automation Programming Language).

La estructura de un programa en RAPL-3, muy aparte de las subrutinas, puede ser descompuesto en 3 partes: configuración, preparación, y el cuerpo del programa.

La configuración consiste, básicamente, en declarar las variables con las que se trabaja.

La preparación deja al Robot listo para trabajar. Esto es, llevar a una velocidad precautoria a READY, limpiar los puertos de salida, dar a las variables de trabajo el valor predeterminado, etc.

El cuerpo del programa es donde se aplican las variables y constantes de trabajo junto con la ruta de programa apropiada, a la par con los puntos enseñados.

A continuación se enlistan los principales comandos y estructuras utilizadas en RAPL-3. Posteriormente, un programa típico comentado.

Manejo de Variables:

Las variables en RAPL-3 pueden ser de varios tipos, y pueden declararse con letras o números. No deben usarse palabras reservadas (en rojo).

<i>int</i>	Entera
<i>float</i>	Flotante
<i>string</i>	Cadena de caracteres
<i>cloc</i>	Posición cartesiana
<i>ploc</i>	Posición de precisión

Anteponer la palabra *teachable* para variables que serán enseñadas desde el pendant.

COMANDO

main

ACCION

Declaración de inicio del programa

Main

<i>end main</i>	Declaración de fin de programa <i>end main</i>
<i>cloc</i>	Posición cartesiana. Se declara el tipo de posición a utilizar. Así mismo se declaran el numero de posiciones a utilizar y el nombre de la cadena. <i>cloc [20]a</i>
<i>ploc</i>	Posición de precisión. Declara el tipo de posición a utilizar, así mismo se declaran el número de posiciones a utilizar y el nombre de la cadena. <i>ploc [10]a</i>
<i>teachable</i>	Declaración de variable <i>teachable ploc [10]a</i>
<i>int</i>	Tipo de variable entera <i>int var</i>
<i>float</i>	Tipo de variable flotante <i>float var</i>
<i>string</i>	Tipo de variable string <i>string var</i>
<i>ready</i>	Posición inicial del robot o posición absoluta. <i>ready()</i>
<i>move</i>	Mueve el brazo a la posición especifica por la variable de posición. El movimiento es realizado en forma de interpolación lineal. <i>move(a[5])</i> mueve el brazo a la posición 5 en la cadena "a"
<i>finish</i>	Espera a que el robot deje de moverse para realizar el siguiente comando. <i>finish()</i>
<i>speed</i>	Declaración de la velocidad de operación del robot en % <i>speed(40)</i>

<i>delay</i>	<p>Pone el programa en estado de espera durante un tiempo determinado en milésimas de segundo.</p> <p><i>delay(2000)</i> espera 2 segundos</p>
<i>goto</i>	<p>Es una instrucción de salto incondicional hacia una etiqueta declarada.</p> <p><i>ciclo::</i> **</p> <p><i>goto ciclo</i></p>
<i>loop,</i> <i>end loop</i>	<p>Esta instrucción se utiliza para realizar una función repetitiva, este ciclo no se rompe a menos que dentro de este ocurra un salto fuera del <i>loop</i>. Para terminar el comando se escribe <i>end loop</i> y para terminar el ciclo hay que presionar en el teclado ctrl+k</p> <p><i>Loop</i> <i>Move(a)</i> <i>Move(b)</i> <i>End loop</i></p>
<i>if, end if</i>	<p>Esta instrucción es utilizada como una condición, aquí se compara una variable y si el resultado es verdadero, el robot lee las líneas entre el <i>if</i> y el <i>end if</i></p> <p><i>Int c</i> <i>If c==5</i> <i>Move(a)</i> <i>Move(b)</i> <i>End if</i></p>
<i>while</i>	<p>Esta instrucción también es para la comparación de una variable, la diferencia con el <i>if</i> es que mientras la condición sea verdadera, queda ciclada entre <i>while</i> y <i>end while</i></p> <p><i>Int c</i> <i>While (contador==5)</i> <i>Move(a)</i> <i>Move(b)</i> <i>C=1+1</i> <i>End while</i></p>
<i>do</i>	<p>Funciona prácticamente igual que <i>while</i>, ciclada mientras su</p>

función sea verdadera. La diferencia es que *do* primero realiza y después compara, lo contrario que *while*.

```
int c
do
move(a)
move(b)
c = 1+1
until contador > 5
```

<i>print</i>	El comando nos permite desplegar mensajes en la pantalla <i>Printf(inicio)</i>
<i>gripper open</i>	Suelta la herramienta Grip_open()
<i>gripper close</i>	Sujeta la herramienta Grip_close()
<i>pendant</i>	Estado en la terminal, se envía el control al teach pendant
<i>ash</i>	Muestra directorio de programas
<i>refresh</i>	Este comando ejecuta las modificaciones hechas al programa.
<i>run</i>	Ejecución del programa.
<i>exit</i>	Sale de la aplicación en curso
<i>output</i>	Habilita el estado de una salida
<i>input</i>	Verificar la señal de las entradas

2.1.3 Controladores Lógicos Programables (PLC) SIEMENS

Un PLC se puede definir como un sistema basado en un microprocesador. Sus partes fundamentales son la Unidad Central de Proceso (CPU), la Memoria y el Sistema de Entradas y Salidas (E/S). La CPU se encarga de todo el control interno y externo del PLC y de la interpretación de las instrucciones del programa. En base a las instrucciones almacenadas en la memoria y en los datos que lee de las entradas, genera las señales de las salidas. La memoria se divide en dos, la memoria de solo lectura o ROM y la memoria de lectura y escritura o RAM.

La memoria ROM almacena programas para el buen funcionamiento del sistema. La memoria RAM está conformada por la memoria de datos, en la que se almacena la información de las entradas y salidas y de variables internas y por la memoria de usuario, en la que se almacena el programa que maneja la lógica del PLC.

El sistema de Entradas y Salidas recopila la información del proceso (Entradas) y genera las acciones de control del mismo (salidas). Los dispositivos conectados a las entradas pueden ser Pulsadores, interruptores, finales de carrera, termostatos, presostatos, detectores de nivel, detectores de proximidad, contactos auxiliares, etc. Al igual, los dispositivos de salida son también muy variados: Pilotos, relés, contactores, Drives o variadores de frecuencia, válvulas, etc.

Entradas y salidas (E/S) - Inputs and Outputs (I/O)

Las entradas y salidas (E/S) de un PLC son digitales, analógicas o especiales. Las E/S digitales se identifican por presentar dos estados diferentes: on u off, presencia o ausencia de tensión, contacto abierto o cerrado, etc. Los niveles de tensión de las entradas más comunes son 5 VDC, 24 VDC, 48 VDC y 220 VAC. Los dispositivos de salida más frecuentes son los relés.

Las E/S análogas se encargan de convertir una magnitud analógica (tensión o corriente) equivalente a una magnitud física (temperatura, flujo, presión, etc.) en una expresión binaria. Esto se realiza mediante conversores analógico-digitales (ADC's). Por último, las E/S especiales se utilizan en procesos en los que con las anteriores E/S vistas son poco efectivas, porque es necesario un gran número de elementos adicionales, ya que el programa necesita de muchas instrucciones o por protocolos especiales de comunicación que se necesitan para poder obtener el

dato requerido por el PLC (HART, Salidas de trenes de impulso, motores paso a paso)

Programa y lenguaje de programación

Se puede definir un programa como un conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada. El Lenguaje de Programación en cambio, permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida.

Al igual como los PLC's se han desarrollado y expandido, los lenguajes de programación también se han desarrollado con ellos. Los lenguajes de hoy en día tienen nuevas y más versátiles instrucciones y con mayor poder de computación. Por ejemplo, los PLC's pueden transferir bloques de datos de una localización de memoria a otra, mientras al mismo tiempo llevan cabo operaciones lógicas y matemáticas en otro bloque. Como resultado de estas nuevas y expandidas instrucciones, los programas de control pueden ahora manejar datos más fácilmente.

Adicionalmente a las nuevas instrucciones de programación, el desarrollo de nuevos módulos de entradas y salidas también ha obligado a cambiar las instrucciones existentes.

Programas de aplicación y del sistema

Los programas de aplicación que crean los usuarios están orientados a ejecutar, a través del controlador, tareas de automatización y control. Para ello, el usuario escribe el programa en el lenguaje de programación que mejor se adapte a su trabajo y con el que sienta poseer un mejor dominio. En este punto es importante señalar, que algunos fabricantes no ofrecen todas las formas de representación de lenguajes de programación, por lo que el usuario deberá adaptarse a la representación disponible

Por otro lado, el conjunto de programas que realizan funciones operativas internas del controlador, incluyendo los traductores de lenguaje, reciben la denominación de programas del sistema o software del sistema. Un elemento importante de éste, es el sistema operativo, cuyos servicios incluyen el manejo de los dispositivos de entrada y salida del PLC, el almacenamiento de la información durante largos períodos, el procesamiento de los programas del usuario, etc. Estos programas ya vienen escritos y están almacenados en una memoria No volátil dentro de la CPU,

por lo tanto no se pierden ni alteran en caso de pérdida de alimentación al equipo. (el usuario No tiene acceso a ellos).

Tipos de lenguajes de programación de PLCs

En la actualidad cada fabricante diseña su propio software de programación, lo que significa que existe una gran variedad comparable con la cantidad de PLC's que hay en el mercado. No obstante, actualmente existen tres tipos de lenguajes de programación de PLC's como los más difundidos a nivel mundial; estos son:

- Lenguaje KOP o de contactos (Escalera)
- Lenguaje AWL (Lista de instrucciones)
- Lenguaje FUB (Diagrama de funciones)

Es obvio, que la gran diversidad de lenguajes de programación da lugar a que cada fabricante tenga su propia representación, originando cierta incomodidad al usuario cuando programa más de un PLC.

Lenguaje KOP

El LADDER, también denominado lenguaje de contactos o de escalera, es un lenguaje de programación gráfico muy popular dentro de los Controladores Lógicos Programables (PLC), debido a que está basado en los esquemas eléctricos de control clásicos. De este modo, con los conocimientos que todo técnico eléctrico posee, es muy fácil adaptarse a la programación en este tipo de lenguaje. Su principal ventaja es que los símbolos básicos están normalizados según normas NEMA y son empleados por todos los fabricantes.

Lenguaje AWL

El lenguaje Booleano utiliza la sintaxis del Álgebra de Boole para ingresar y explicar la lógica de control. Consiste en elaborar una lista de instrucciones o nemónicos, haciendo uso de operadores Booleanos (AND, OR, NOT, etc.) y otras instrucciones nemónicas, para implementar el circuito de control. El lenguaje "Lista de Instrucciones" (IL) de la Norma IEC 1131-3, es una forma de lenguaje Booleano.

Ejemplo de programación Booleana:

A	I	2.3
A	I	4.1
O	I	3.2
=	Q	1.6

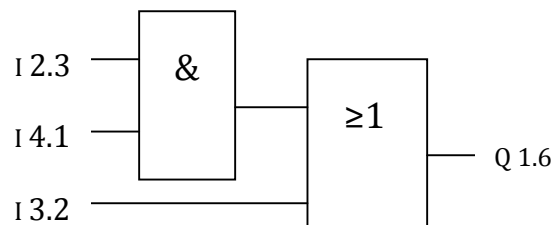
Lenguaje FUB

Es un lenguaje gráfico que permite al usuario programar elementos (bloque de funciones del PLC) en tal forma que ellos aparecen interconectados al igual que un circuito eléctrico. Generalmente utilizan símbolos lógicos para representar al bloque de función. Las salidas lógicas no requieren incorporar una bobina de salida, porque la salida es representada por una variable asignada a la salida del bloque.

El diagrama de funciones lógicas, resulta especialmente cómodo de utilizar, a técnicos habituados a trabajar con circuitos de puertas lógicas, ya que la simbología usada en ambos es equivalente.

Adicionalmente a las funciones lógicas estándares y específicas del vendedor, el lenguaje FBD de la Norma IEC 1131-3 permite al usuario construir sus propios bloques de funciones, de acuerdo a los requerimientos del programa de control.

Ejemplo de programación mediante diagrama de funciones:



Profibus

Profibus es actualmente uno de los líderes de los sistemas basados en buses de campo en Europa y goza de una aceptación mundial. Sus áreas de aplicación incluyen manufacturación, automatización y generación de procesos. Profibus es un bus de campo normalizado internacionalmente que fue estandarizado bajo la norma EN50170. Los componentes de distintos fabricantes pueden comunicarse sin necesidad de ajustes especiales de interfaces.

Profibus puede ser usado para transmisión crítica en el tiempo de datos a alta velocidad y para tareas de comunicación extensas y complejas. Esta versatilidad viene dada por las tres versiones compatibles que componen la familia Profibus:

- Profibus PA:
 - o Diseñado para automatización de procesos
 - o Permite la conexión de sensores y actuadores a una línea de bus común

incluso en áreas especialmente protegidas.

o Permite la comunicación de datos y energía en el mismo bus (norma IEC 1158-2)

- Profibus DP:
 - o Optimizado para alta velocidad
 - o Conexiones sencillas y baratas
 - o Diseñada especialmente para la comunicación entre los sistemas de control de automatismos y de E/S distribuidas

- Profibus FMS:
 - o Solución general para tareas de comunicación a nivel de célula
 - o Gran rango de aplicaciones y flexibilidad
 - o Posibilidad de uso en tareas de comunicación complejas y extensas

Puede decirse que Profibus ha conseguido definir toda una red de comunicación industrial, desde el nivel físico hasta el de aplicación.

Comunicación

Desde el punto de vista del control de las comunicaciones, el protocolo Profibus es maestro-esclavo:

Aplicaciones mono maestro. Un sólo maestro está activo en el bus, usualmente un PLC. Los demás dispositivos son esclavos. Este esquema es el que permite los ciclos de lectura más cortos

Aplicaciones multimaestro. Permite más de un maestro. Pueden ser aplicaciones de sistemas independientes, en que cada maestro tenga sus propios esclavos u otro tipo de configuraciones con dispositivos de diagnóstico y otros.

En un ambiente multimaestro, pueden haber dos tipos de maestros:

- DPM1. DP Master Class 1. Es un controlador central que intercambia información con sus esclavos en forma cíclica. Típicamente un PLC.

- DPM2. DP Master Class 2. Son estaciones de operación, configuración o ingeniería. Tienen acceso activo al bus, pero su conexión no es necesariamente permanente.

Junto con las especificaciones de otros buses de campo se recoge en las normas internacionales IEC61158 e IEC61784.

Características:

- Velocidades de transmisión:
9.6, 19.2, 93.75, 187.5, 500, 1500, 3000, 6000 y 12000 Kbps.
- Número máximo de estaciones: 127 (32 sin utilizar repetidores).
- Distancias máximas alcanzables (cable de 0.22 mm de diámetro):
Hasta 93.75 KBaudios: 1200 metros 187.5 KBaudios: 600 metros 500 KBaudios:
200 metros
- Estaciones pueden ser activas (maestros) o pasivas (esclavos).
Conexiones de tipo bidireccionales, multicast o broadcast.

Terminadores de bus

La instalación de un cableado para una Red Profibus DP requiere de un terminador en cada extremo del bus. Según la norma este terminador debe ser activo. Esto es, no basta una resistencia terminal, sino que se trata de un arreglo de resistencias que están energizadas.

El objetivo de estos terminadores es garantizar un voltaje de referencia en estado inactivo del bus, es decir, sin mensajes, y minimizar las reflexiones de línea. Generalmente uno de los extremos de la red es el maestro Profibus DP, y en ese caso proporcionará uno de los terminadores activos.

El otro extremo del bus tiene dos alternativas. Que el terminador activo sea provisto por el último nodo de la red, o que sea provisto por un terminador externo. La primera opción es muy simple, pero tiene el inconveniente de que si se requiere desenergizar el nodo, se perderá la función del terminador activo, comprometiendo con ello la integridad de toda la red. Por ello, puede ser preferible tener la resistencia activa en forma separada, de tal modo de mantenerla siempre energizada, generalmente desde una UPS.

En el caso de Profibus PA los terminadores de bus son pasivos, es decir, sólo resistencia.

2.1.4 Administrador Simatic STEP-7

Es el software estándar para configurar y programar los sistemas de automatización SIMATIC.

Funciones de STEP 7

- **Crear y gestionar proyectos de automatización:**
- **Configurar el hardware**
 - Asignar módulos hardware a un rack
 - Asignar direccionamientos
 - Parametrización de los módulos
- **Configurar las comunicaciones**
 - Definir maestros y esclavos
 - Fijar las propiedades de conexión
- **Administrar el programa de usuario**
 - Escribir el programa (diferentes lenguajes) y depurarlo
 - Cargar programas en sistemas de destino
 - Evaluar el programa en ejecución
 - Diagnosticar fallos

La jerarquía de objetos para proyectos y librerías del Administrador SIMATIC está estructurada (ver Fig. 2.2).



Fig. 2.2 Ventana de Jerarquía de Proyecto

Configurar

- Consiste en la disposición de bastidores y módulos: PS, CPU, E/S, DP, etc. en la ventana del equipo del proyecto (ver Fig. 2.3).

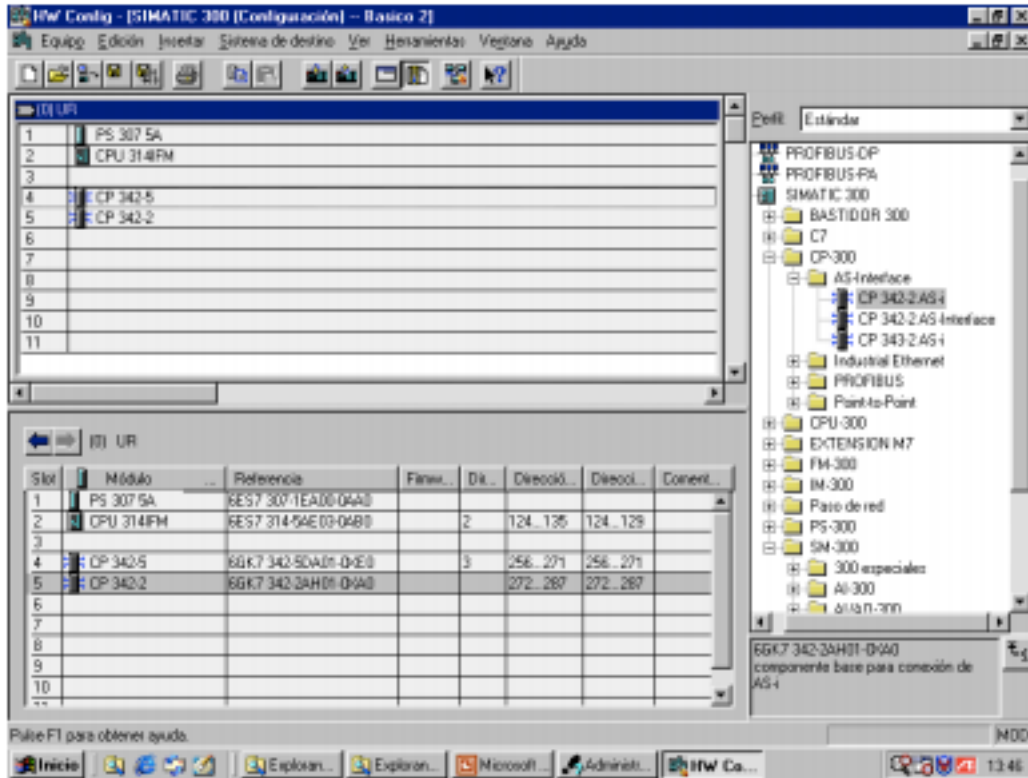


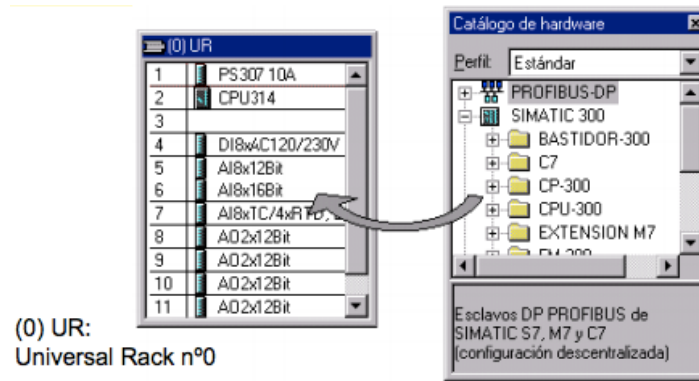
Fig. 2.3 Ventana de Configuración

- En la Tabla de Configuración, Step-7 asigna automáticamente una dirección a cada módulo.
- La configuración se puede copiar cuantas veces se desee a otros proyectos de Step 7.
- Durante el arranque, la CPU compara la configuración teórica (creada con Step-7) con la configuración física (real de la instalación) y detecta posibles errores.

Parametrizar

- Consiste en ajustar las propiedades de los diferentes módulos y del bus de comunicaciones (si ha lugar) .

El lugar donde se agrega la configuración física del PLC dentro del programa es el bastidor de configuración (ver Fig. 2.4).



(0) UR:
Universal Rack nº0

Fig. 2.4 Bastidor de Configuración

Para tener una idea más clara de cómo está la relación de la configuración física del PLC con el bastidor en el programa ver la fig. 2.5

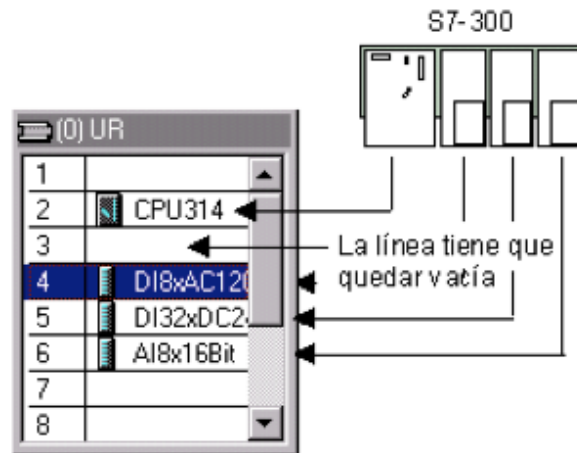


Fig. 2.5 Ilustración de posición de componentes del bastidor en PLC físico

Reglas de ocupación de los slots (S7-300)

- Slot 0: Bastidor 0:
- Slot 1: sólo fuentes de alimentación (p. ej. 6ES7 307-...) o dejar vacío
- Slot 2: sólo módulos centrales CPU (p. ej., 6ES7 314-...)
- Slot 3: sólo módulos interface (p. ej. 6ES7 360-.../361-...) o dejar vacío
- Slots 4 a 11: módulos de señal, módulos de función, procesadores de comunicación (CP) o dejar vacío **NO DEJAR ESPACIOS LIBRES** (excepto IM)

Los cambios hechos en el ordenador NO serán efectivos hasta que sean transferidos al PLC:

- Primero, asegurarse de que el conmutador de la CPU está en la posición RUN-P o STOP, entonces:
- “Click” en el menú “Sistema de destino” y elegir “Cargar en módulo”.
- El sistema hará varias preguntas, a las cuales se responden con “Sí”, “Aceptar” o “Cerrar”, según el caso.
- Una vez que se configura el hardware y transferido al PLC, cerrar la ventana HW Config Tool, continuando el trabajo en la ventana del Administrador Simatic.

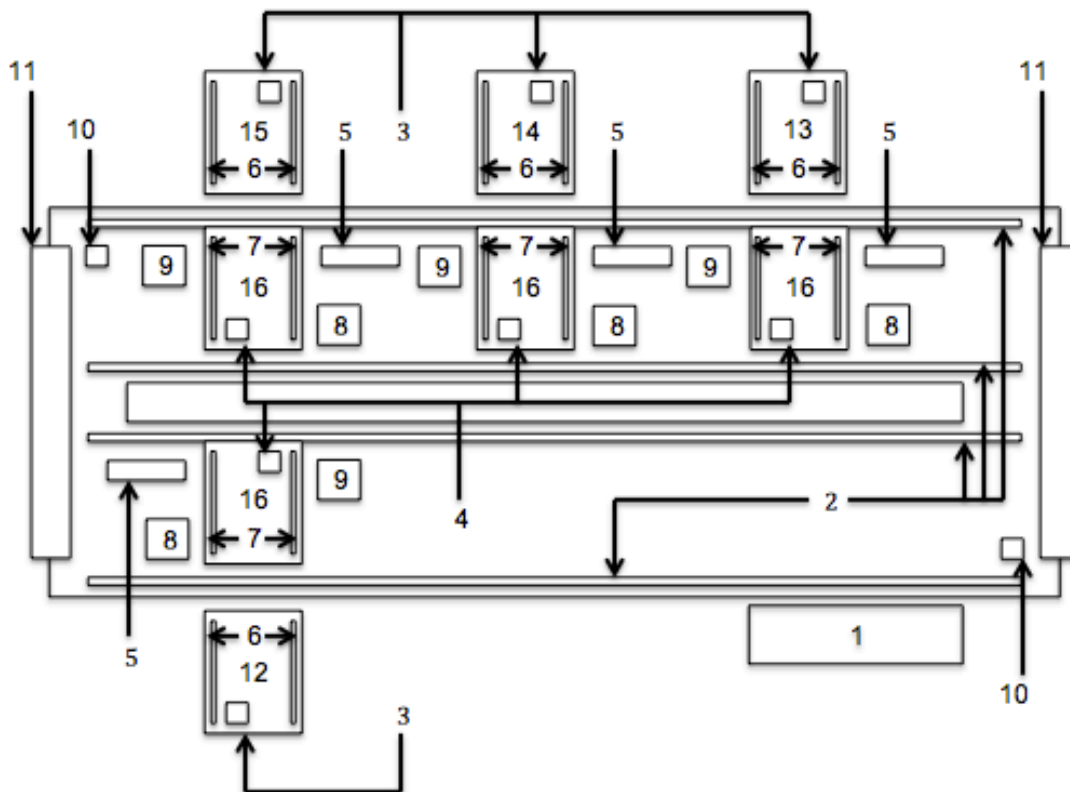
CAPITULO III

3.1 Procedimiento y descripción de las actividades realizadas

3.1.1 Banda Transportadora y PLC 315-2DP

La Banda Transportadora es un sistema de transporte formados principalmente por dos cadenas que son las encargadas del recorrido de los Palets, en cada extremo de la Banda se encuentra un pistón que realiza el movimiento para cambio de cadena, cuenta con cuatro Bahías las cuales son de entrada y salida, Bahía 1, Bahía 2 y Bahía 3 cada una con sus respectivas plataformas que mueven los Palets desde las cadenas principales a las Bahía y viceversa. Cada Bahía y plataforma de Bahía cuenta con sensores capacitivos y sensores inductivos para la detección y clasificación de Palets, así como de pistones para la sujeción de palets y clasificación de Bahía, también cuenta con dos sensores capacitivos para detectar un Palet y hacer el cambio de Banda principal.

Diagrama a Bloques de la Banda Transportadora.



1. PLC 315-2DP
2. Cadenas principales de Banda
3. Sensores Capacitivos de Bahías
4. Sensores Capacitivos de Plataformas de Bahías
5. Sensores Inductivos detectores de Código de Palet para Bahía
6. Cadenas de Bahías
7. Cadenas de Plataformas de Bahías
8. Pistones de Configuración de Bahías
9. Pistones de Sujeción de Palet
10. Sensores Capacitivos cercano a Pistón de Cambio de Banda
11. Pistones de Cambio de Banda
12. Bahía de Entrada y Salida
13. Bahía 1
14. Bahía 2
15. Bahía 3
16. Plataformas de Bahías

Se investigaron diferentes comandos y su funcionamiento en el PLC S7-315-2DP de SIEMENS, para la programación KOP en el Administrador SIMATIC también de la compañía SIEMENS.

Este PLC cuenta con dos idiomas para su programación Ingles y Alemán. En este programa se pueden crear diferentes tipos de bloques, entre estos bloques se encuentran las funciones que se cargan en un bloque maestro llamado Objeto, el Objeto será el que finalmente se cargue en el PLC se pueden hacer tantas funciones como sean necesarias y mandarlas a llamar en el Objeto esto con la finalidad de agrupar las diferentes secuencias que se encuentren en las funciones, las funciones se ejecutaran en el orden que sean llamadas en la programación del KOP o si bien se desea ejecutar simultáneamente dos funciones a la vez.

Para entender mejor el funcionamiento de Administrador SIMATIC que es el programa especializado para los PLC's S7-300, se asistió a un curso básico que fue impartido en las instalaciones del ITESI en el edificio I por uno de los ex-alumnos que en la actualidad trabaja en la industria.

En este curso se aprendió a utilizar el programa desde el aspecto mas simple hasta poder manejar un motor por comunicación Profibus utilizando un MicroMaster de Siemens.

En el curso se ve como configurar el programa para reconocer los PLC´s S7-300, aquí se configura todo lo relacionado con las características técnicas que vienen el PLC físico; la fuente, el CPU, las entradas, las salidas y módulos especiales. En la Fig. 3.1 que aparece a continuación se puede observar el aspecto de la pagina principal de Administrador SIMATIC.

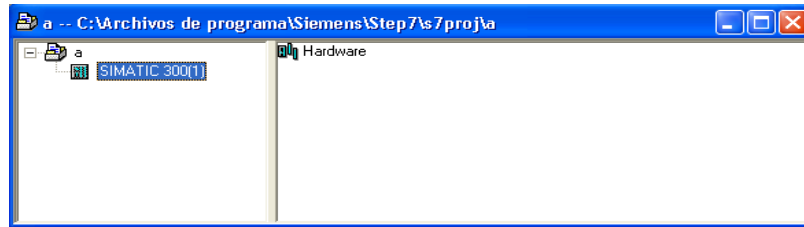


Fig. 3.1 Ventana de Principal

Para realizar este procedimiento se tienen que seguir ciertas reglas en la ventana del programa ya que esta distribuido en forma de tabla que se conoce con el nombre de bastidor (véase Fig. 3.2) y las filas de este bastidor están destinadas para cada componente del PLC. Al seleccionar una fila se elige de un menú el objeto que concuerde con las características que vienen escritas en el PLC físico.

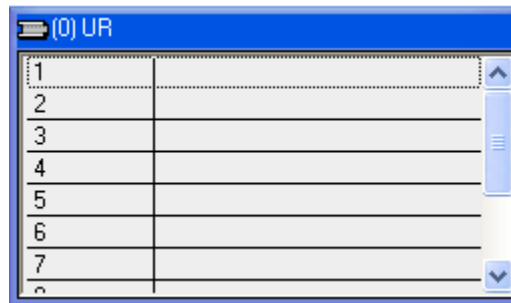


Fig. 3.2 Bastidor

En la primer fila se coloca el tipo de fuente que esta utilizando el PLC.

En la segunda se coloca el modelo de CPU.

En la tercer fila se coloca los módulos especiales.

Y a partir de la cuarta fila se colocan lo que son los módulos de entradas y salidas.

En lo que respecta a la fuente y CPU sólo se selecciona uno de los objetos que vienen en el menú. Pero en cuanto a las entradas y salidas (ver Tabla 3.1) viene la opción de configurar las direcciones que ya vienen por default en el programa con

esto facilitar el reconocimiento entre diferentes módulos de entradas o de salidas si se cuenta con ellos.

Una vez terminada la configuración de las características del CPU, dirigirse a la ventana donde se programará ya sea en lenguaje KOP, AWL, FUB. En esta ventana se divide la programación en objetos (OB) y funciones (FN). (ver Fig. 3.3).

Tabla 3.1 Configuración de Salidas PLC 315-2DP

Salidas (Alemán)	Descripción
A16.0	Banda Principal (Activa banda principal)
A16.1	Banda Principal (Desactiva banda principal)
A16.2	Bahía 1 (activa cadenas mete a bahía)
A16.3	Bahía 1 (activa cadenas saca de bahía)
A16.4	Bahía 2 (activa cadenas mete a bahía)
A16.5	Bahía 2 (activa cadenas saca de bahía)
A16.6	Bahía 3 (activa cadenas mete a bahía)
A16.7	Bahía 3 (activa cadenas saca de bahía)
A17.0	Bahía Entrada/Salida (activa cadenas mete a bahía)
A17.1	Bahía Entrada/Salida (activa cadenas saca de bahía)
A17.7	Levanta Plataforma Entrada/Salida
A20.0	Levanta Pistón de Plataforma 3
A20.1	Levanta Plataforma de Bahía 3
A20.2	Levanta Pistón antes de entrar a Bahía 3
A20.4	Levanta Pistón de Bahía Entrada/Salida
A20.5	Activa Pistón lateral de cambio de banda Cercano a Bahía 3
A20.6	Levanta Pistón Antes de entrar a Bahía Entrada/Salida
A20.7	Levanta Pistón antes de entrar a Bahía 2
A21.0	Levanta Plataforma de Bahía 2
A21.1	Levanta Pistón de Plataforma 2
A21.3	Levanta Pistón antes de entrar a Bahía 1
A21.4	Levanta Plataforma de Bahía 1
A21.5	Levanta Pistón de Plataforma 1
A21.7	Activa Pistón lateral de cambio de banda Cercano a Bahía 1

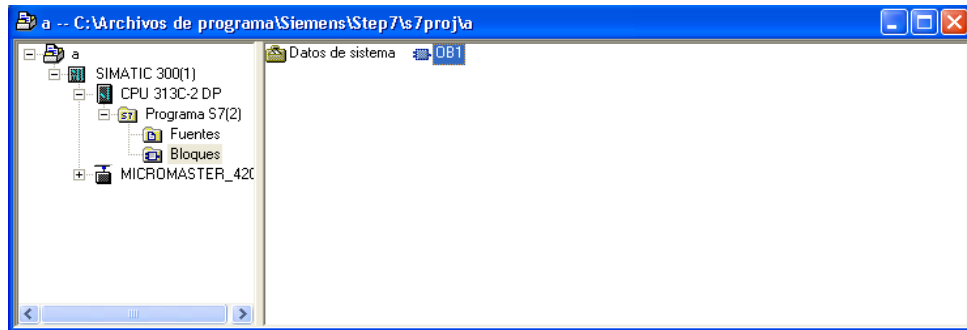


Fig. 3.3 Ventana de Objetos

Después de recibir el curso ya se cuenta con los conocimientos para programar la Banda Transportadora, pero antes de ello se realizó la clasificación de las entradas y salidas del PLC S7-315-2DP probando cada una de las salidas y entradas (ver Tabla 3.2) de los módulos, así como la clasificación de los componentes que constituyen la Banda Transportadora.

La banda transportadora consta de:

- Cuatro Bahías por donde pueden entrar o salir los Palets, Bahía de entrada y salida, Bahía 1, Bahía 2 y Bahía 3, las Bahías cuentan con dos cadenas paralelas para el deslizamiento de los Palets las cuales son movidas por un motor, cada Bahía cuenta con un sensor capacitivo para la detección de estos (ver Fig. 3.4).
- Dos Bandas para el traslado de los Palets a las Bahías, la Banda principal (en esta se encuentra la Bahía de entrada y salida) y la Banda secundaria (en esta se encuentran las Bahías 1, 2 y 3) están movidas por dos motores (ver Fig. 3.5).
- Dos pistones, el pistón A que cambia de la Banda principal a la secundaria y el pistón B que cambia de la banda secundaria a la principal a los Palets, los pistones son activados por unos sensores capacitivos (ver Fig. 3.6).
- Cuatro Plataformas ubicadas una en cada Bahía para el ascenso o descenso de los Palets, estas Plataformas cuentan con un pistón cada una los cuales sirven para la retención de los Palets, igual que las Bahías cuentan con un sensor capacitivo y dos cadenas paralelas también movidas por un motor (ver Fig. 3.7).
- Cuatro pistones ubicados antes de las Plataformas para la retención de los palets y facilitar la clasificación de estos ya que hay palets que salen por la

misma bahía y de igual forma evitar que estos choquen con la Plataforma cuando esta se encuentre elevada (ver Fig. 3.8).

- Cuatro series conformadas por seis sensores inductivos cada serie, esto con el fin de reconocer la configuración que tiene cada Palet para saber que Bahía se activara (ver Fig. 3.8).
- Nueve Palets con diferentes configuraciones cada uno, para realizar diferentes clasificaciones con los sensores inductivos, ejemplo de un Palet en la Fig. 3.9.

Tabla 3.2 Configuración de Entradas PLC 315-2DP

Entradas (Alemán)	Descripción
E0.2	Sensor de Pistón de cambio de Banda cercano a Bahía 1
E0.4	Sensor de Pistón antes de Plataforma 1 OK
E0.5	Sensor de Plataforma 1 Levantada OK
E0.6	Sensor de Pistón Plataforma 1 OK
E0.7	Sensor de Bahía 1
E1.0	Sensor de Plataforma 1
E1.3	Sensor de Pistón antes de Plataforma 2 OK
E1.4	Sensor de Plataforma 2 Levantada OK
E1.5	Sensor de Pistón Plataforma 2 OK
E1.6	Sensor de Bahía 2
E1.7	Sensor de Plataforma 2
E4.2	Sensor de Pistón antes de Plataforma 3 OK
E4.3	Sensor de Plataforma 3 Levantada OK
E4.4	Sensor de Pistón Plataforma 3 OK
E4.6	Sensor de Bahía 3
E4.7	Sensor de Plataforma 3
E5.3	Sensor de Pistón antes de Plataforma Entrada/Salida OK
E5.4	Sensor de Plataforma Entrada/Salida Levantada OK
E5.5	Sensor de Pistón Plataforma Entrada/Salida OK
E5.6	Sensor de Bahía Entrada/Salida
E5.7	Sensor de Plataforma Entrada/Salida
E8.0	Sensor de Pistón de cambio de Banda entre Bahía 3 y Entrada/Salida
E9.0	1er Sensor antes de entrar a Plataforma Entrada/Salida
E9.1	2do Sensor antes de entrar a Plataforma Entrada/Salida
E9.2	3er Sensor antes de entrar a Plataforma Entrada/Salida
E9.3	4to Sensor antes de entrar a Plataforma Entrada/Salida
E9.4	5to Sensor antes de entrar a Plataforma Entrada/Salida
E9.5	6to Sensor antes de entrar a Plataforma Entrada/Salida
E9.6	1er Sensor antes de entrar a Bahía 1
E9.7	2do Sensor antes de entrar a Bahía 1

E12.0	3er Sensor antes de entrar a Bahía 1
E12.1	4to Sensor antes de entrar a Bahía 1
E12.2	5to Sensor antes de entrar a Bahía 1
E12.3	6to Sensor antes de entrar a Bahía 1
E12.4	1er Sensor antes de entrar a Bahía 2
E12.5	2do Sensor antes de entrar a Bahía 2
E12.6	3er Sensor antes de entrar a Bahía 2
E12.7	4to Sensor antes de entrar a Bahía 2
E13.0	5to Sensor antes de entrar a Bahía 2
E13.1	6to Sensor antes de entrar a Bahía 2
E13.2	1er Sensor antes de entrar a Bahía 3
E13.3	2do Sensor antes de entrar a Bahía 3
E13.4	3er Sensor antes de entrar a Bahía 3
E13.5	4to Sensor antes de entrar a Bahía 3
E13.6	5to Sensor antes de entrar a Bahía 3

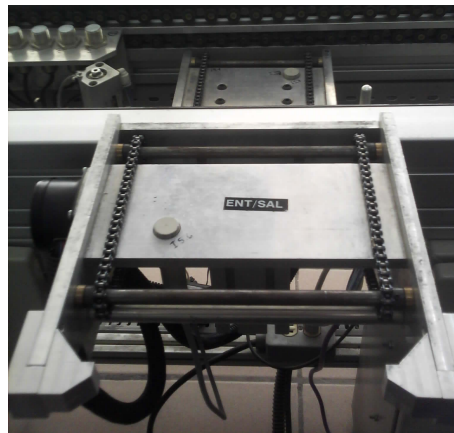


Fig. 3.4 Ejemplo de Bahía

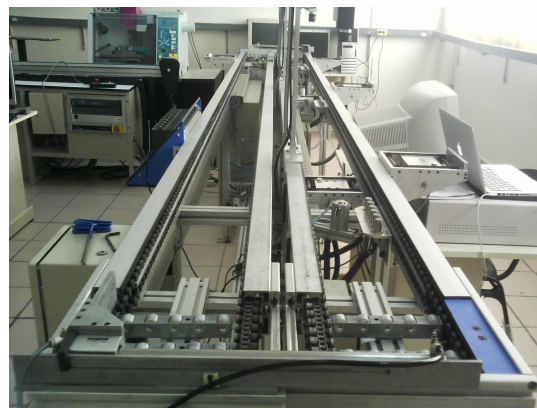


Fig. 3.5 Cambio de Banda principal a secundaria



Fig. 3.6 Pistón de cambio de banda principal a secundaria

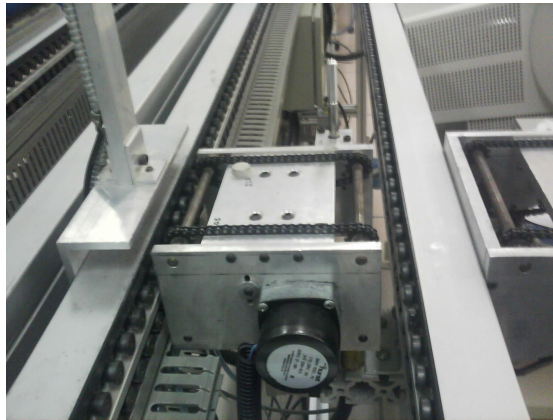


Fig. 3.7 Ilustración de Plataforma de Bahía

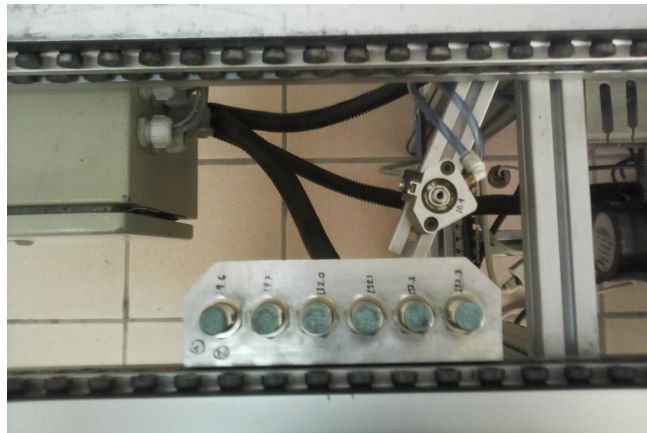


Fig. 3.8 Sensores inductivos

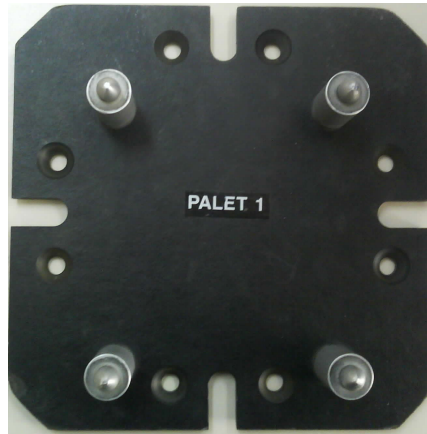


Fig. 3.9 Palets (vista superior)

Se procedió a hacer diferentes secuencias para la Banda Transportadora, para encontrar la mas adecuada y la que brinde un mejor funcionamiento.

Se eligió la siguiente secuencia:

La Bahía de entrada y salida será por donde se administren los Palets a la Banda principal, dependiendo de la configuración que tenga el Palet (ver Fig. 3.10) que sea ingresado a la banda será retirado por la Bahía 1, 2 o 3, dicha configuración será detectado con los seis sensores inductivos ubicados antes de cada Plataforma, todos los Palets cuentan con una configuración diferente en la parte de abajo, y con esto los sensores detectan un único código por Palet. La configuración (Código Físico) se hace con tornillos distribuidos en la parte inferior de cada Palet, configuración distinta para cada uno.

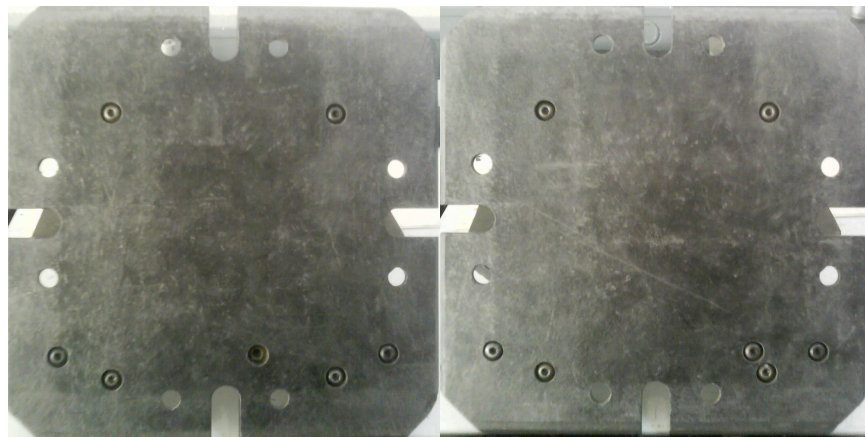


Fig. 3.10 Palet 1 y Palet 2 (vista inferior)

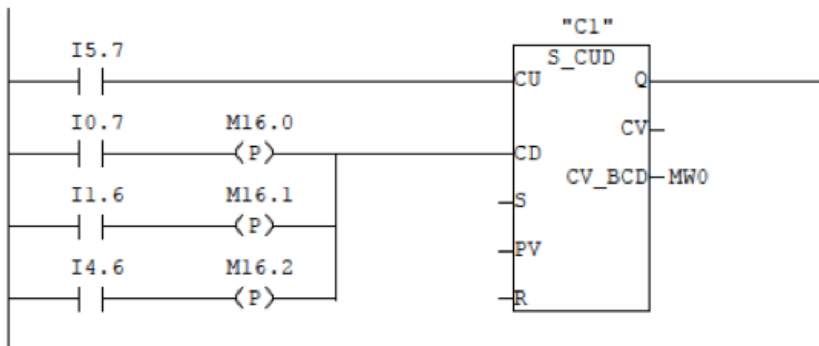
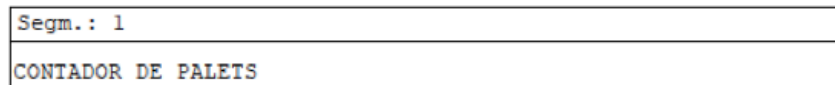
A continuación se presentan los diagramas de tiempo y la programación en lenguaje KOP utilizado en la Banda Transportadora.

Diagramas de Tiempo.

Activación y desactivación de cadenas principales de la Banda principal.

Las cadenas principales se activan cuando el sensor de la Bahía de entrada (I5.7) detecte un Palet entrando y se desactivan cuando la cantidad de palets ingresados salga por cualquiera de las tres Bahías de salida y sea detectado por el sensor capacitivo correspondiente de la Bahía (I0.7, I1.6, I4.6). Para poder realizar este proceso se programó un contador el cual incrementa su cuenta por cada palet que se detecte con el sensor de la Bahía de entrada y decrementa por cada Palet detectado en cualquiera de los sensores de las Bahías de salida.

Programación KOP contador de Palets entrantes y salientes de la Banda.



Información del símbolo
C1 C1

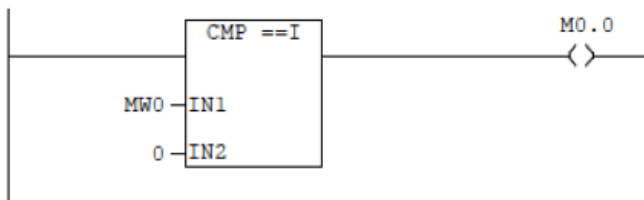
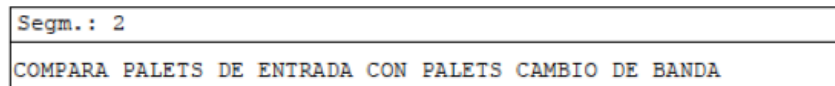
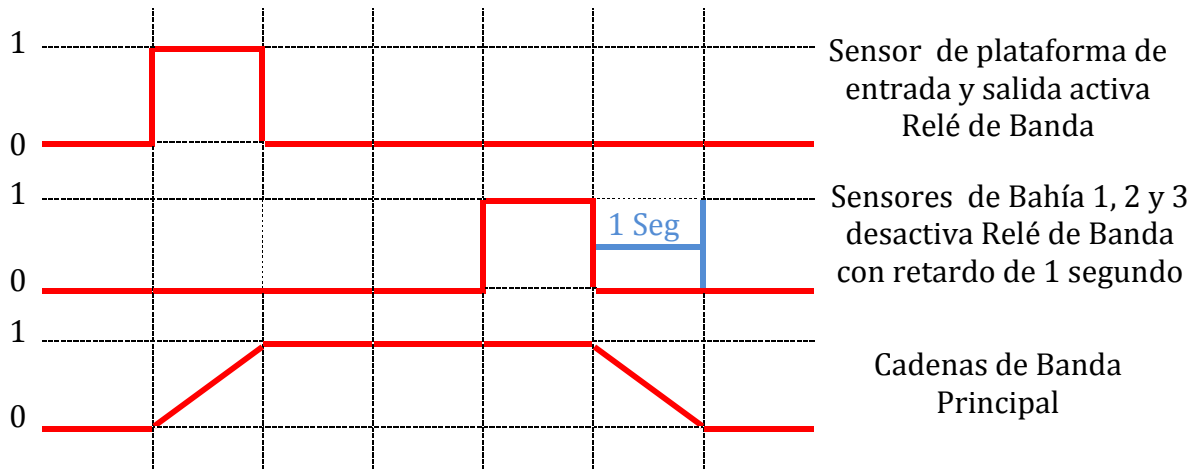
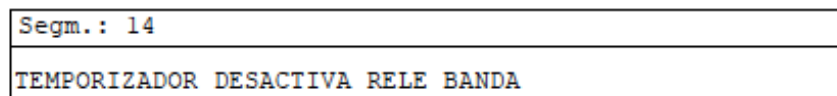
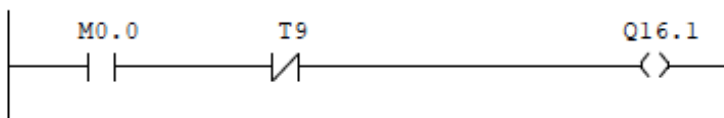
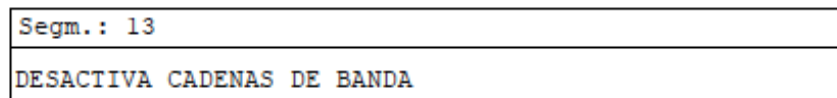
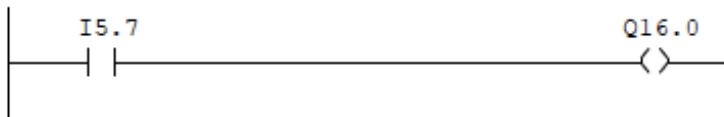
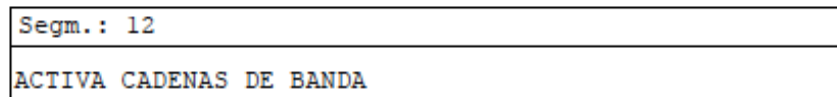


Diagrama de tiempo activación y desactivación de Cadenas Principales.



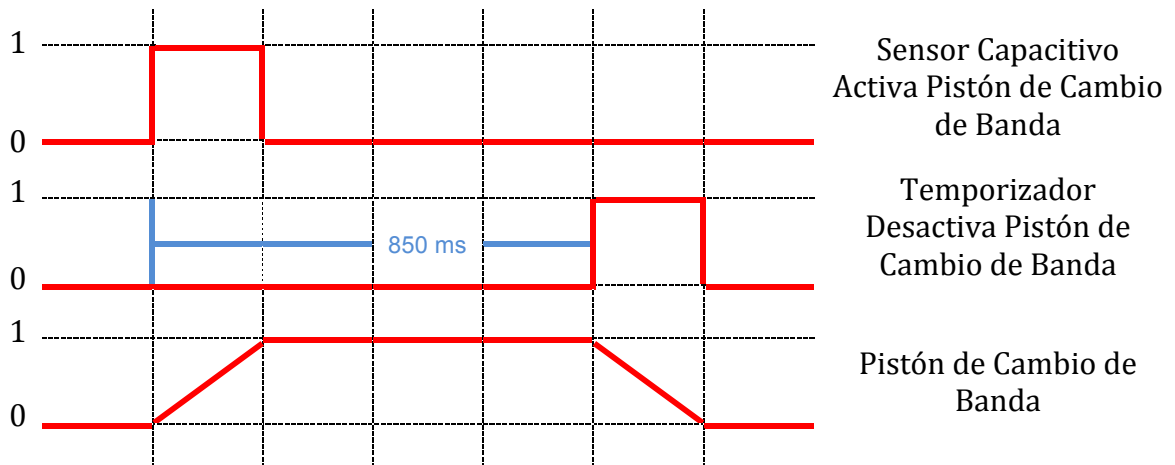
Programación KOP activación y desactivación de cadenas principales de la Banda.



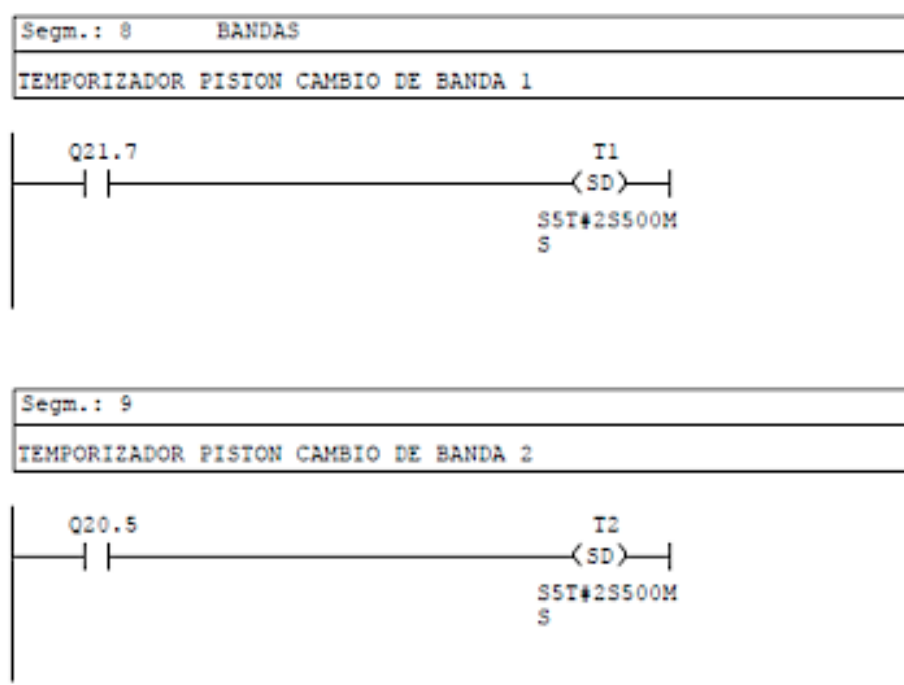
Activación y desactivación de pistones de Cambio de Banda

Los pistones de cambio de Banda se activan con un sensor cercano a los pistones (I0.2 para Pistón cercano a Bahía 1 e I8.0 para Pistón cercano a Bahía 3) y se desactivan con un temporizador el cual activa una bandera a los 850 ms y desenchava al pistón de cambio de Banda.

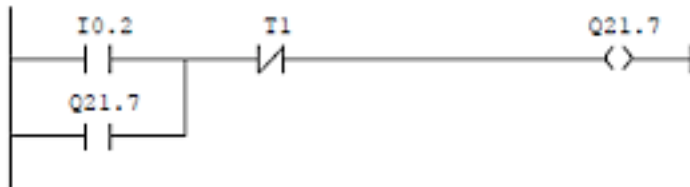
Diagrama de tiempo de activación y desactivación de Pistones de Cambio de Banda.



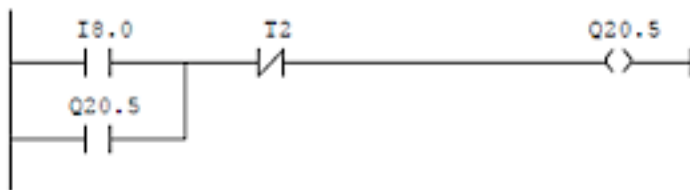
Programación KOP de pistones de Cambio de Banda.



Segn.: 10
 ACTIVA PISTON CAMBIO DE BANDA 1



Segn.: 11
 ACTIVA PISTON CAMBIO DE BANDA 2



Activación y desactivación de pistón de Plataforma de Bahía y de Sujeción de Palets.

Cada Palet tiene un “Código Físico” (Ver Anexos) que detectan los sensores inductivos (ver Tabla 3.3), el cual activa el pistón de Plataforma de Bahía y el pistón de sujeción de Palet correspondiente a la Bahía asignada a dicho Palet y se desactiva cuando el sensor capacitivo de la Bahía detecte al palet salir.

Tabla 3.3 Código de Palets

CODIGO DE PALETS							
PALET	S1	S2	S3	S4	S5	S6	ACTIVA
1	1	0	0	0	0	1	BAHIA 1
2	1	1	0	0	0	1	BAHIA 2
3	0	0	0	1	0	1	BAHIA 3
A	1	0	1	0	1	0	BAHIA 3
C	1	0	0	0	0	1	BAHIA 1
E	0	1	1	1	0	0	BAHIA 3
F	1	0	0	1	0	0	BAHIA 2
G	0	1	0	0	1	0	BAHIA 2
H	1	0	0	1	1	0	BAHIA 1

Diagrama de Tiempo de activación y desactivación de pistones de Plataforma.

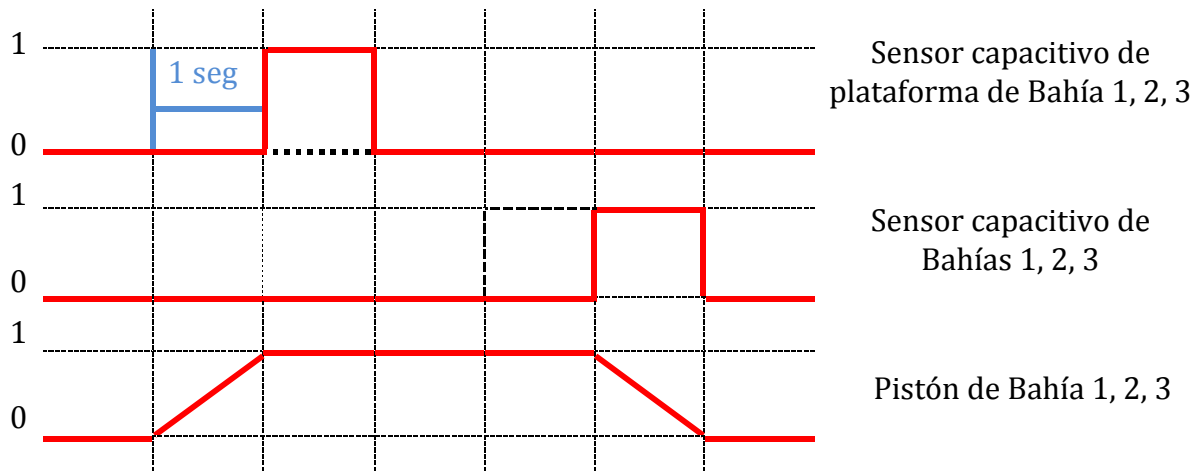
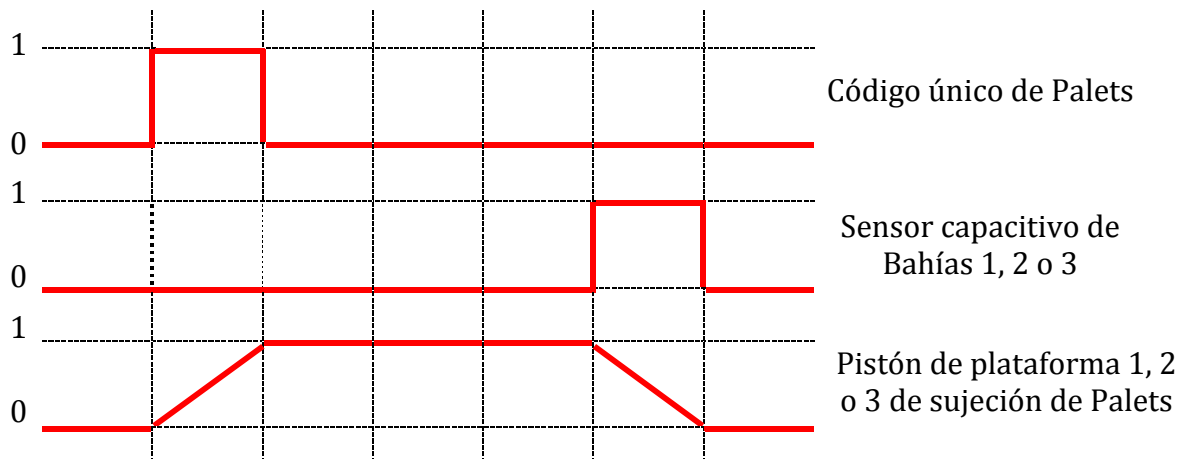
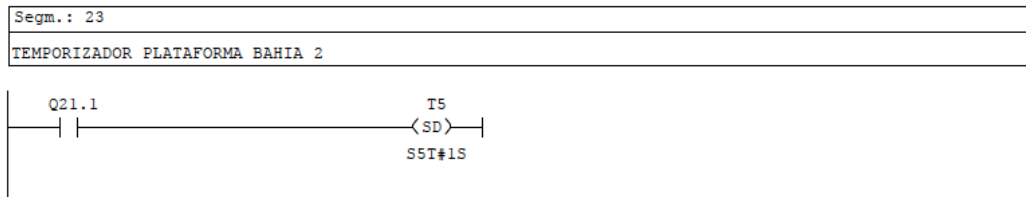
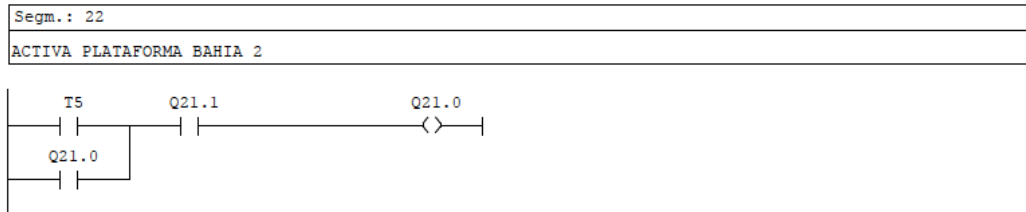
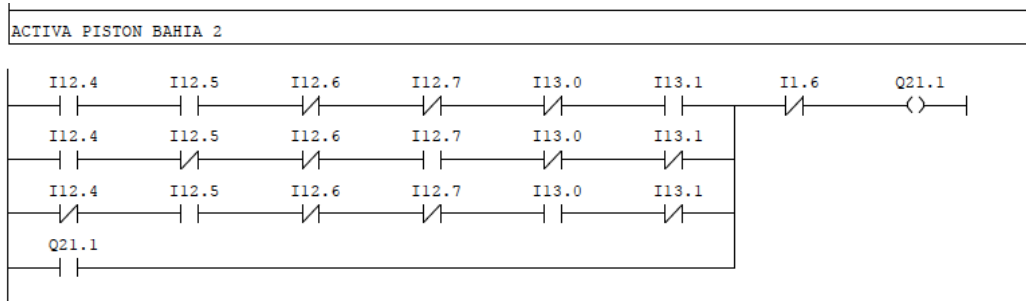


Diagrama de Tiempo de activación y desactivación de pistones de sujeción de Palet.



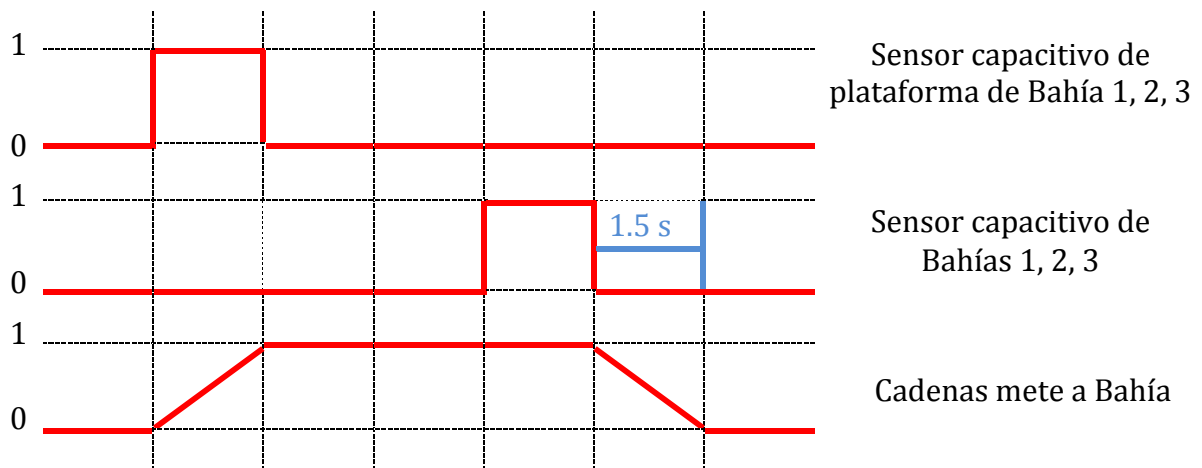
La programación KOP asignada a las Bahías para detectar los códigos de cada Palet, consta en cerrar los contactos NA en serie y así activar el pistón que le corresponde y con ese mismo se enclava para que el pistón permanezca activado y se desactiva con un contacto NC el cual hace referencia al sensor de una de las Bahías.

Activación de Pistón de Plataforma de Bahía y de Sujeción de Palets por código de Palet y desactivación por sensor de Bahía (Bahía 2).



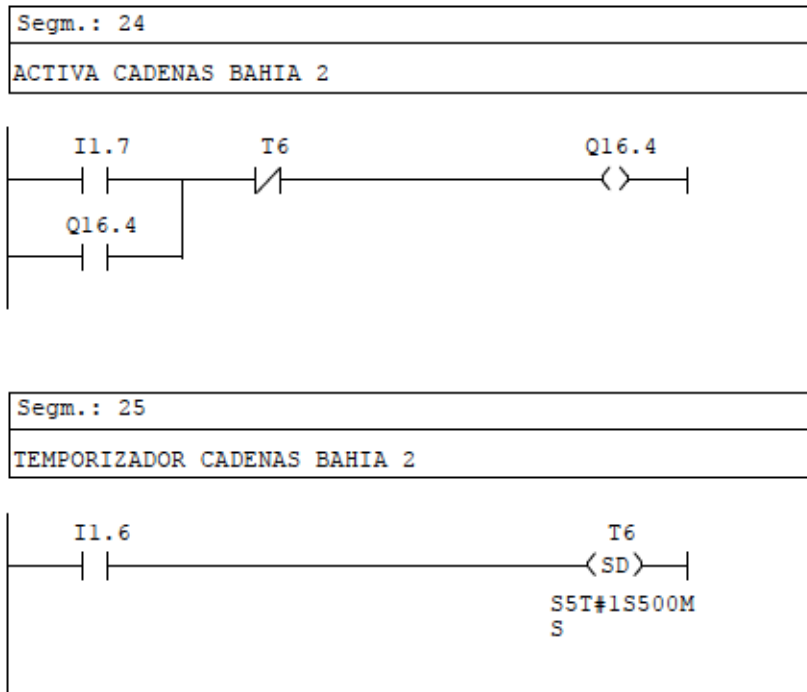
Activación y desactivación de cadenas mete a Bahía.

Diagrama de Tiempo de Activación y desactivación de cadenas de Bahía



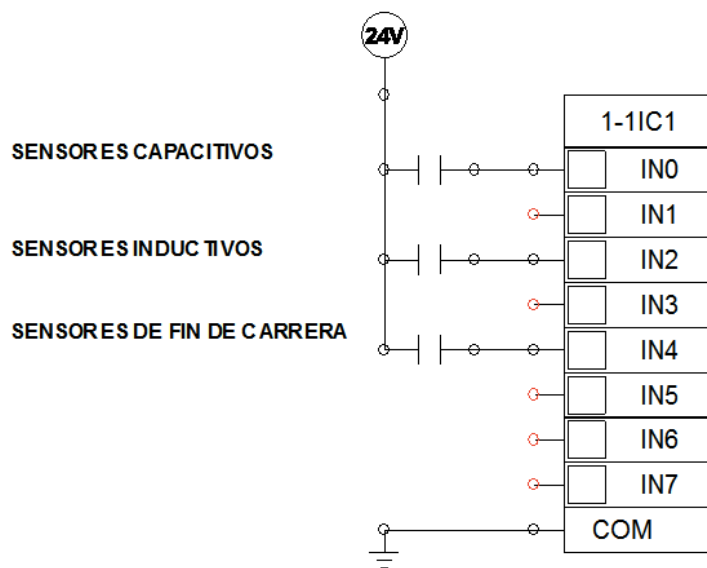
Las cadenas que meten a Bahía se activan cuando el sensor capacitivo de plataforma de Bahía detecte un Palet y se desactiva después de un segundo y medio cuando el sensor capacitivo de Bahía detecte salir el Palet, esto para cada Bahía correspondiente a cada Palet.

Cadenas mete a Bahía.



Diagramas eléctricos de las entradas y salidas del PLC

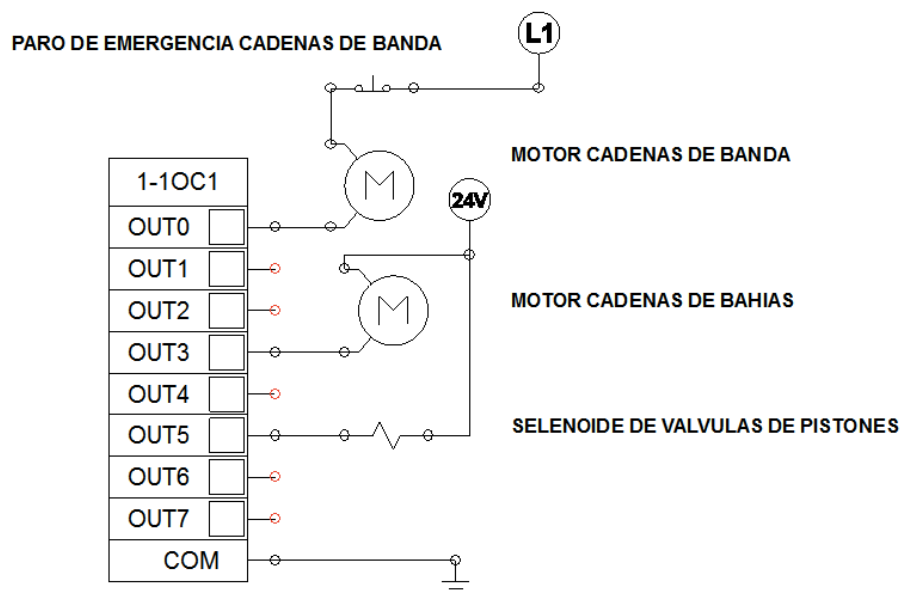
Diagrama eléctrico de las entradas del PLC 315-2DP.



El PLC cuenta en sus entradas con:

- 10 Sensores Capacitivos de 3 líneas 24 VDC, Tierra Común y una de Señal.
- 24 Sensores Inductivos de 3 líneas 24 VDC, Tierra Común y una de Señal.
- 12 Sensores de Fin de Carrera de 3 líneas 24 VDC, Tierra Común y una de Señal.

Diagrama eléctrico de las salidas del PLC 315-2DP.



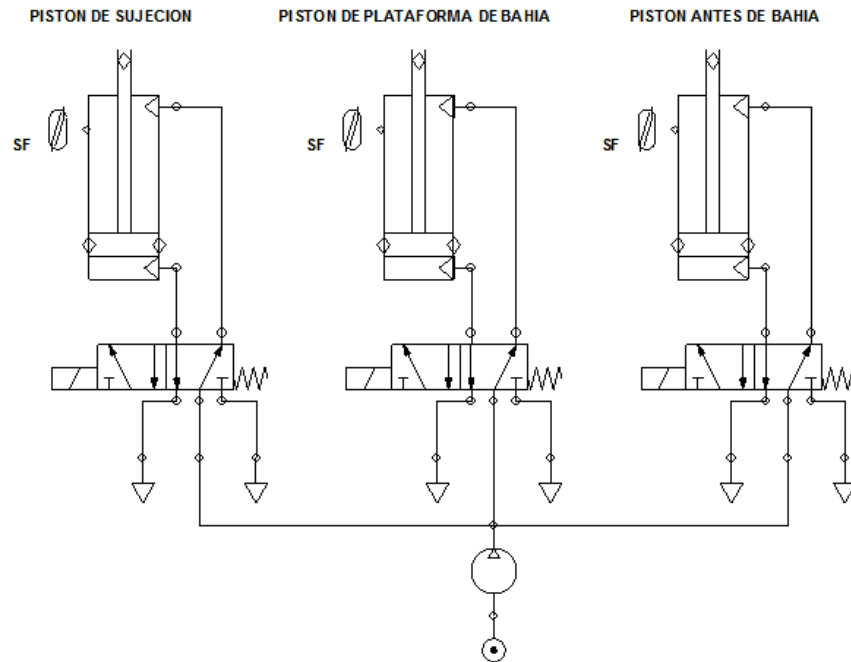
El PLC cuenta en sus salidas con:

- 2 Motores Monofásicos para las cadenas Principales de la Banda.
- 8 Motores de CD alimentados con 24 V para las cadenas de las Bahías.
- 14 Válvulas alimentadas con 24 VDC para cada pistón.

Los Motores Monofásicos tienen un interruptor de Paro de Emergencia el cual desactiva el funcionamiento de las cadenas principales de la Banda.

Diagramas neumáticos de los pistones que componen la Banda Transportadora.

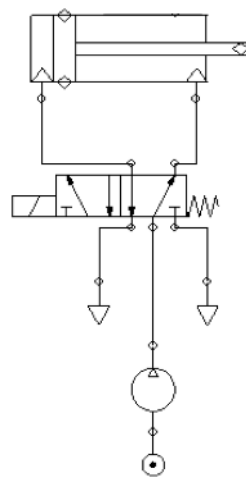
Diagrama Neumático de pistones de Bahías.



Con este diagrama de pistones se ejemplifica a una de las Bahías ya que las tres están compuestas por la misma cantidad de pistones y en el mismo orden, los pistones cuentan con un sensor de fin de carrera.

Diagrama Neumático de pistones de cambio de Banda.

PISTON DE CAMBIO DE BANDA



Este diagrama representa la conexión de los pistones de Cambio de Banda (cercano a Bahía 1 y cercano a Bahía 3).

3.1.2 Brazo de Robot

El Brazo de Robot cuenta con un mecanismo de sujeción llamado Grip o Gripper y 6 ejes. Cada parte del Brazo de Robot es similar a la de un brazo humano incluyendo los movimientos que este puede realizar. a continuación la información detallada de cada uno de los componentes del brazo (de Robot).

- Eje o Joint 1 “Waist”, su movimiento es de rotación izquierda-derecha en base al eje X (ver Fig. 3.11). Grados de libertad permitidos de hasta +/- 350 y una velocidad máxima de 210 grados/seg.
- Eje o Joint 2 “Shoulder Arm”, realiza movimientos rotatorios abajo-arriba sobre el eje Z (ver Fig. 3.11). Grados de libertad permitidos de hasta +/- 110 y una velocidad máxima de 210 grados/seg.
- Eje o Joint 3 “Elbow”, al igual que el “Shoulder Arm se mueve de la misma forma sobre el eje Z (ver Fig. 3.11). Grados de libertad permitidos de hasta +/- 220 y una velocidad máxima de 125 grados/seg.
- Eje o Joint 4 “Pitch”, este tiene movimientos rotarios en base al eje Z (ver Fig. 3.11). Grados de libertad permitidos de hasta +/- 220 y una velocidad máxima de 675 grados/seg.
- Eje o Joint 5 “Roll”, al igual que el “Waist” su movimiento es de rotación izquierda-derecha en base al eje X (ver Fig. 3.11), en esta parte se encuentra el “Grip”. Grados de libertad permitidos de hasta +/- 360 y una velocidad máxima de 1360 grados/seg.
- Eje o Joint 6 (Motor de Eje), este eje permite al brazo hacer movimientos de izquierda a derecha y viceversa en una distancia total de 4 metros aproximadamente, permitiendo al brazo hacer puntos intermedios en el eje si así lo desea en la programación. Grados de libertad permitidos puede moverse por todo el eje abarcando toda la distancia total de este, velocidad máxima desconocida.
- “Grip o Tool”, no es mas que una tenaza activada por un pistón que abre o cierra dos placas para sujetar objetos (los usados en este proyecto serán los “Palets”).
- Programador Manual “Teach Pendant”, El programador manual es una herramienta para mover el robot, enseñar posiciones, crear variables y correr programas. Una vez que la aplicación ya esta iniciada, por medio de

variables aprendidas (teachables) es posible guardar posiciones, iniciar variables, etc.

Nota.- Cuando alguno de los ejes llega a sobrepasar los grados de libertad permitidos se desactivará el Brazo (modo de protección), para no causar daños en este.

Los 360 grados de libertad de el "Roll" no pueden llevarse acabo ya que el pistón de el "Grip" esta conectado a dos mangueras por donde fluye el aire para dar la presión al pistón, a causa de esto no se hace el giro completo ya que si se hiciera las mangueras se reventarían, provocando daños en el Brazo.

"Home" son los valores iniciales del Brazo de Robot, es el estado en el que el Brazo debe estar para poder ser programado ya sea de la computadora o desde el "Teach".

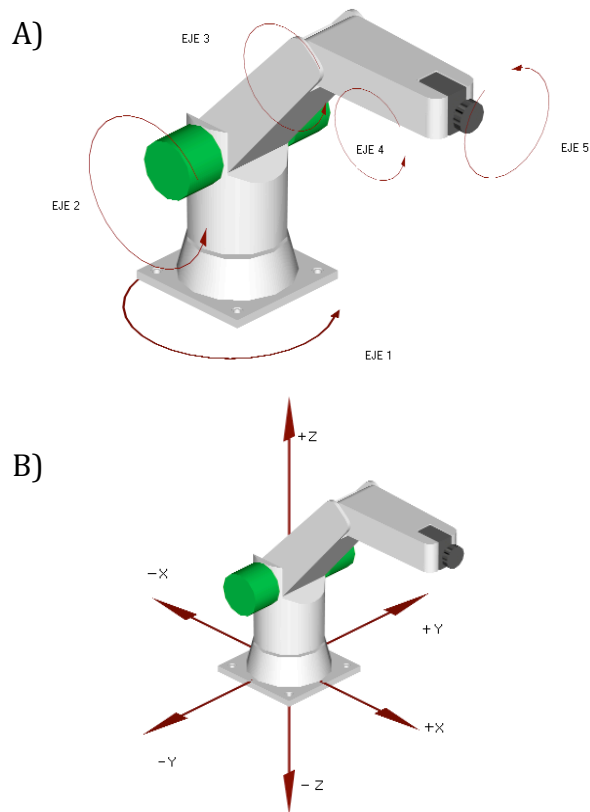


Fig. 3.11 Tipo de movimientos del Brazo Robot
A) *Movimiento de Coordenadas por Eje*
B) *Movimiento Cartesiano*

En la Fig. 3.12 se puede apreciar el Brazo de Robot en 2 ángulos diferentes y sus partes que lo componen.

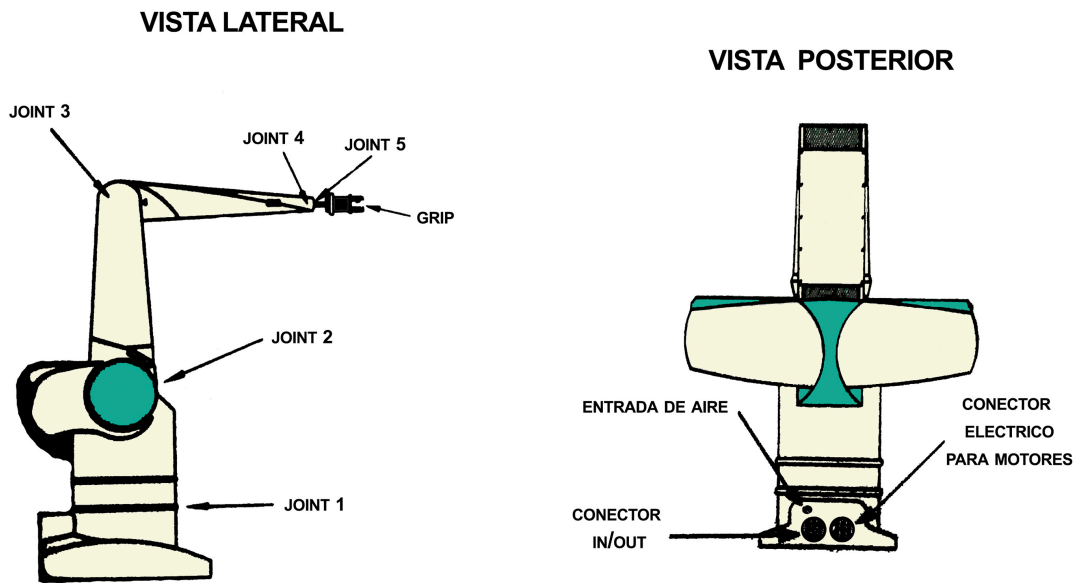


Fig. 3.12 Ilustración física del Brazo de Robot

En la Fig. 3.13 se muestra las partes que componen el Joint 6.

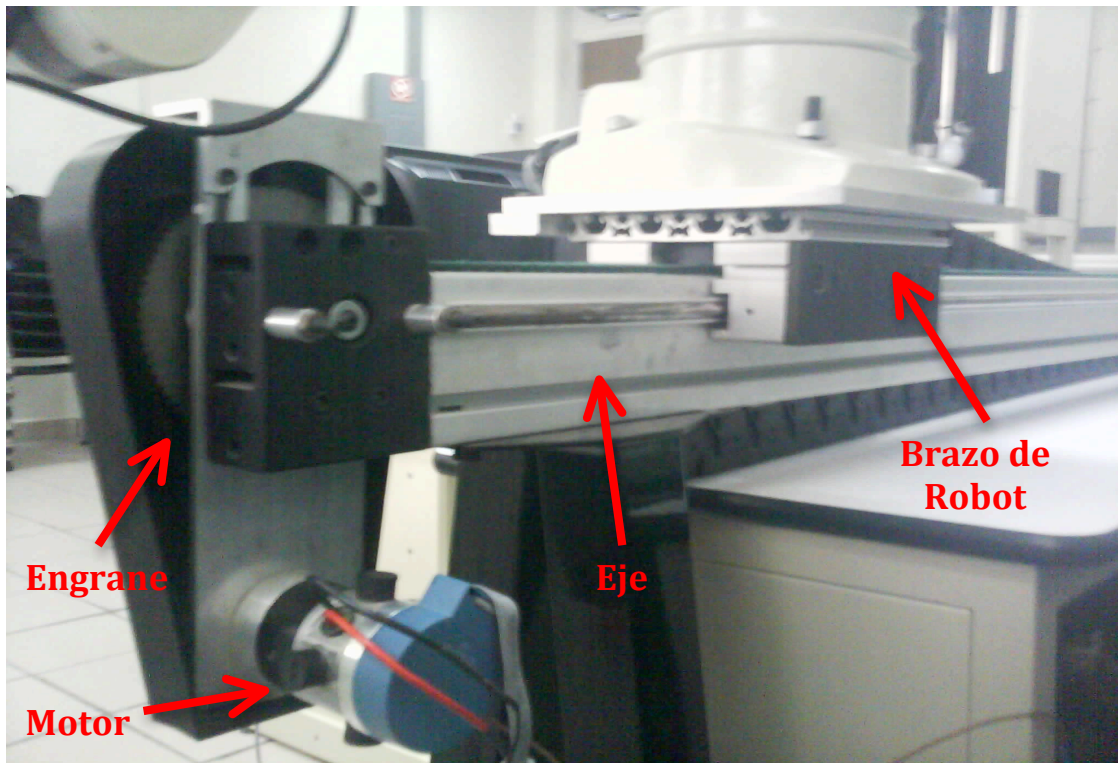


Fig. 3.13 Configuración Joint 6

Todos los Joint a excepción de el Joint 6 vienen definidos por default en el modulo de programación.

Si por alguna razón el Joint 6 estuviera desconfigurado del modulo de programación el Brazo de Robot nunca se podría ir a “Home”, y de esta forma el Brazo no podría ser programado para realizar alguna acción. Cuando este sea deshabilitado tendrá que ser configurado con los valores de fabrica establecidos, en los cuales se define si el motor esta sobre un eje o no, así como la relación de engrane-motor y la velocidad de el motor.

El motor puede funcionar con valores diferentes a los de fabricación, pero no es recomendable, ya que si usara valores mayores a los de fabrica podría ocasionar daños al Brazo.

En la Fig. 3.14 se puede apreciar el sentido de los movimientos permitidos al Brazo de Robot así como los grados de libertad antes mencionados.

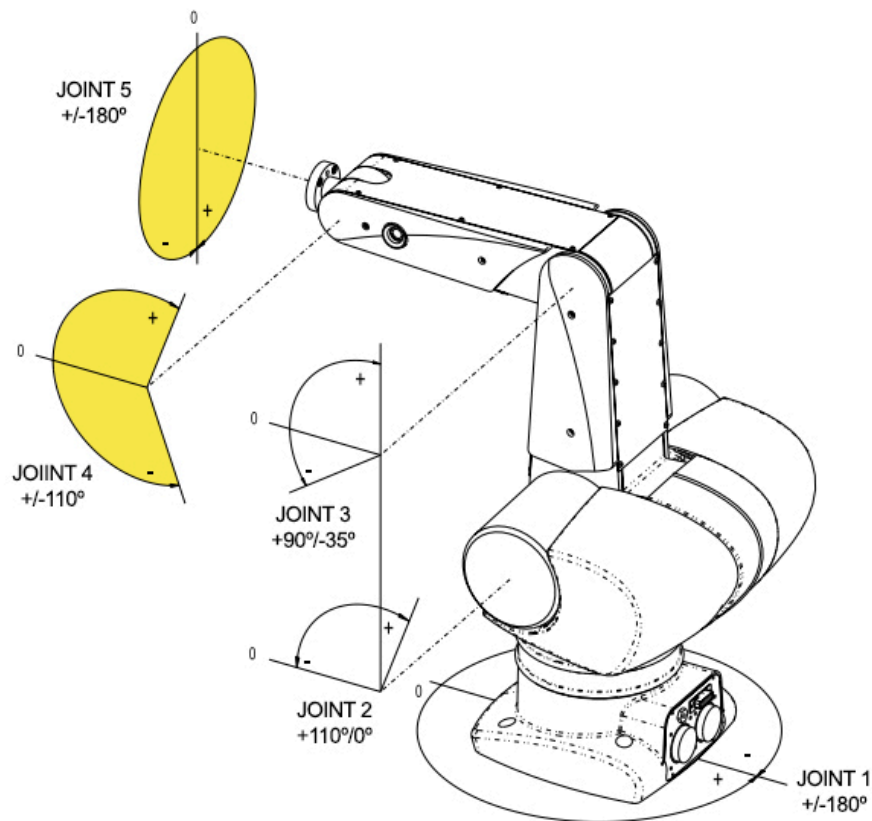


Fig. 3.14 Grados de Movimiento del Brazo de robot

Se le enseñó al Robot diferentes posiciones en tres dimensiones las cuales se utilizarán para la transportación de los Palets desde el Plano Cartesiano hasta la Banda Transportadora.

Teniendo estas posiciones se procedió a realizar la programación para realizar el recorrido del Brazo de Robot. Dicha programación fue hecha de diferentes formas para conocer distintas secuencias de movimiento del Brazo de Robot y poder probar cual es la adecuada para el proyecto.

Explicación de la programación del Brazo de Robot

La programación del Brazo de Robot se empezó creando una aplicación nueva dentro del programa Robcomm3 y enseñar los puntos de trabajo.

- Para crear un nueva aplicación donde se programa la secuencia del Robot es necesario seguir los pasos que a continuación se enumeran:

- 1) Abrir el Robcomm3 (ver Fig. 3.15).

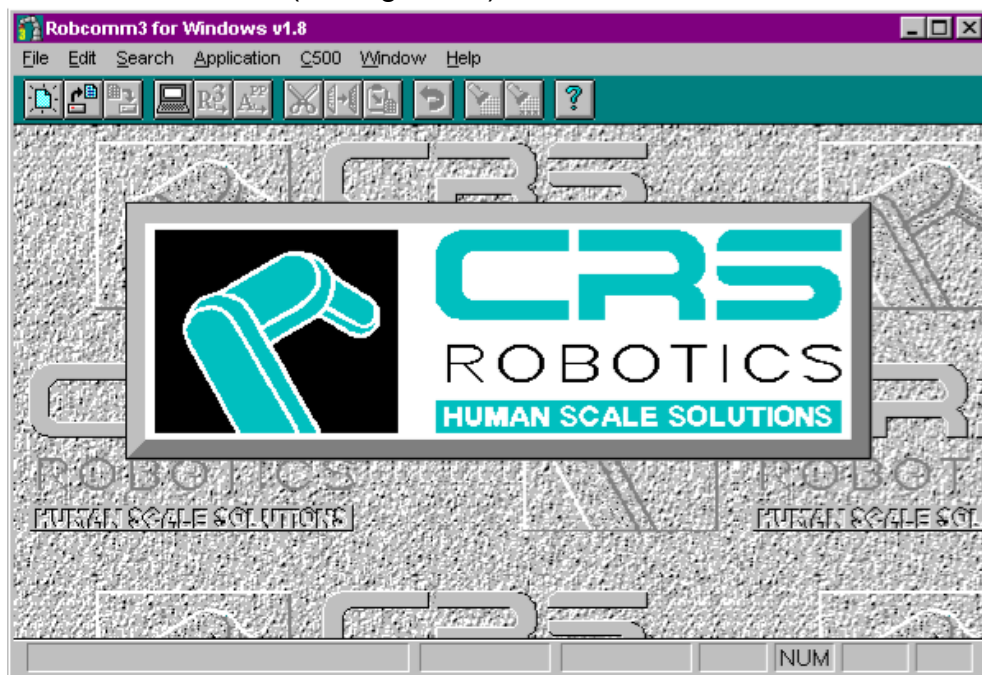


Fig. 3.15 Ventana Principal

- 2) Crear una aplicación (ver Fig. 3.16)
En el menú Application, seleccionar New Application

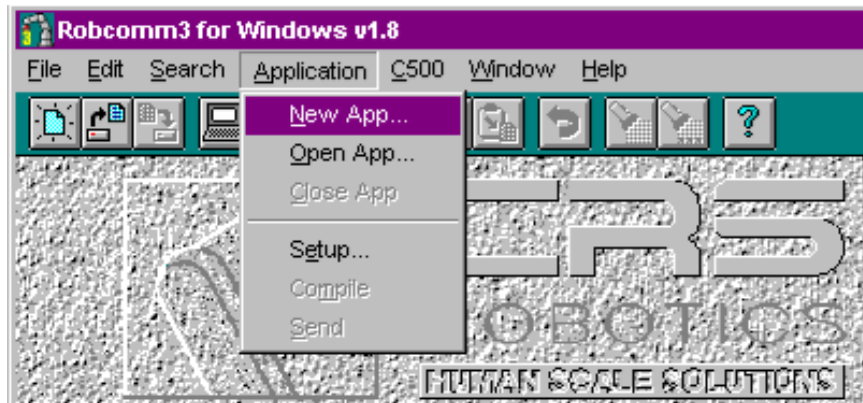


Fig. 3.16 Menú Application

En ese momento aparecerá un cuadro de mensaje en donde pedirá el nombre de la aplicación que se desea generar (ver Fig. 3.17), evitar borrar la extensión app.

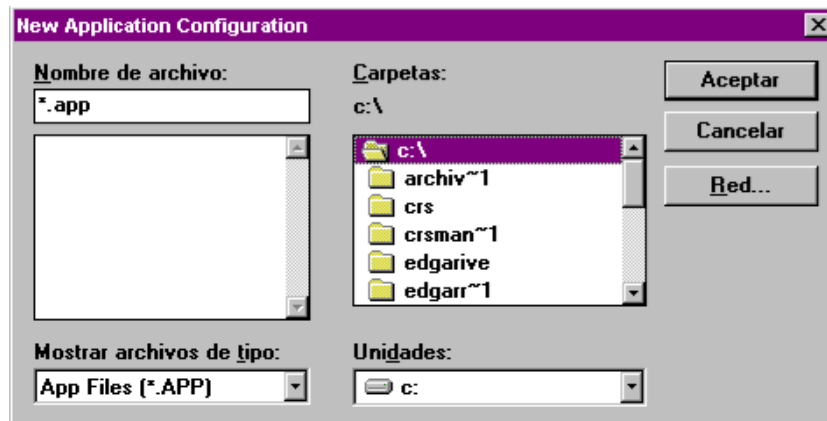


Fig. 3.17 Ventana emergente de Configuración

- Para enseñar los puntos los cuales se utilizan para que el Brazo de Robot se mueva a lo largo del eje 6 desde el Plano Cartesiano donde recolecta el Palet hasta la Banda Transportadora donde los deposita se realiza lo siguiente:

- Se debe crear una variable donde se guarde la posición.
- Se debe colocar el Robot en la posición que se desea enseñar

Existen dos métodos para crear las variables y colocar el robot en la posición que se quiere enseñar, que se explican a continuación:

Desde el teach.

1. Comando tch
2. F1 (var)
3. Nombre de la variable
4. Especificar el tipo y la dimensión de la variable
5. Ir a posición a guardar
6. Oprimir F1(tch)
7. Si se desean más variables repetir los pasos

Desde la terminal.

Es necesario estar en una aplicación para poder lograrlo. Dentro de la aplicación se mueve el Brazo y se guarda la posición. Con el comando "here *nom*" donde *nom* es el nombre de la variable.

Una vez enseñado los puntos de trabajo se comenzó a editar la programación de la secuencia que se realizó. Para hacer esto se ingresó al editor de programas de RAPL-3 en el menú Archivo seleccionamos New y en la pantalla que aparece será donde vamos a capturar el código (ver Fig. 3.18).

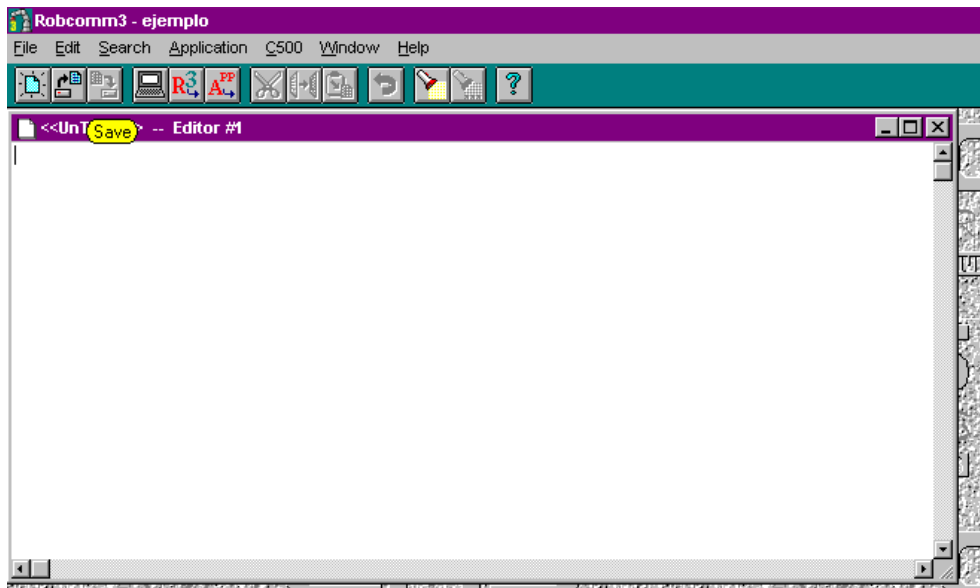
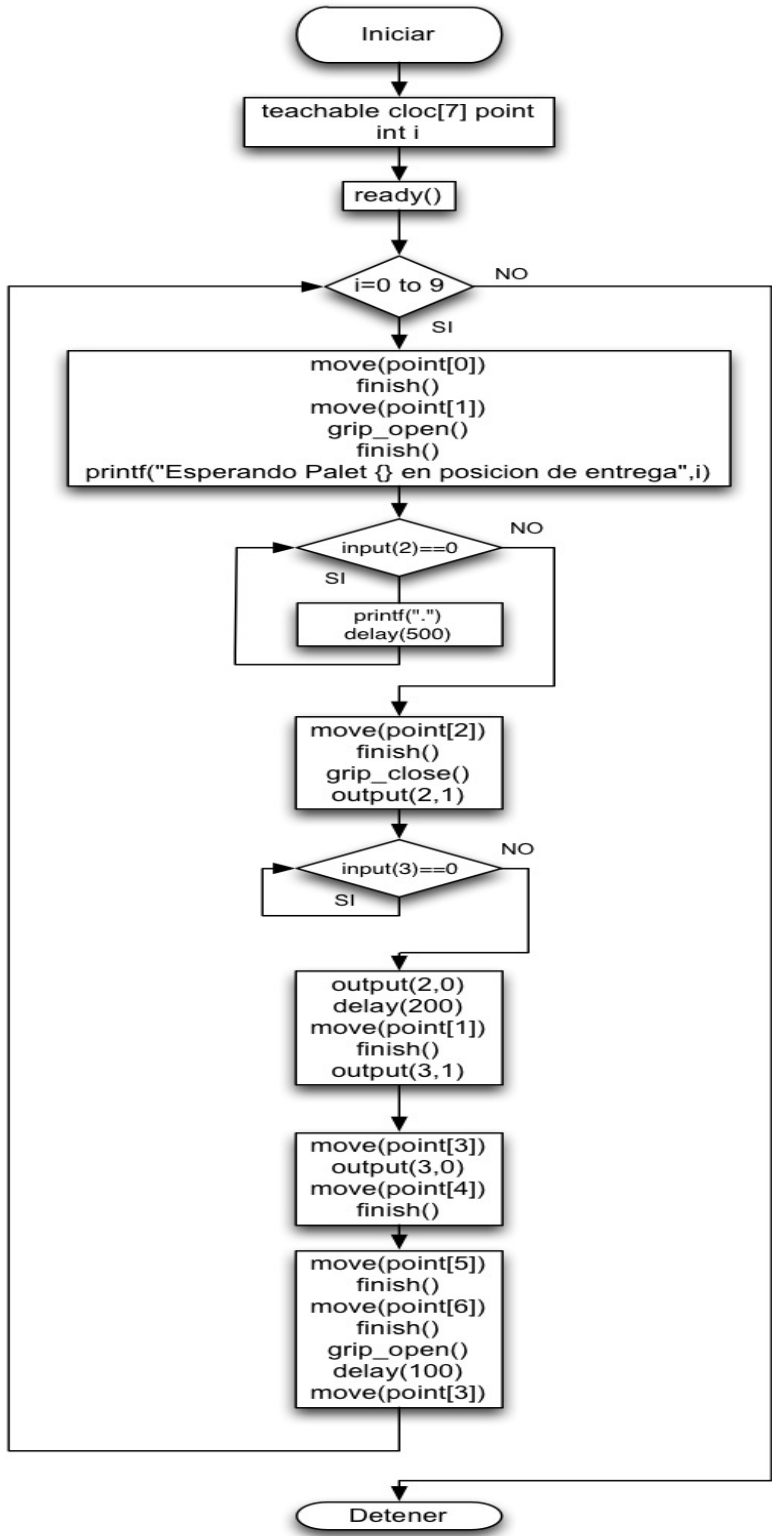


Fig. 3.18 Editor de Programas

La programación que se realizó se representa con el diagrama de flujo y se explica a continuación.

La secuencia programada que se utilizó en el Brazo Robot comienza mandándolo a su posición *ready()* lo que indica que el Robot esta preparado, seguidamente se pasa a una condición de seguridad para saber que la secuencia se realiza nueve veces que es la cantidad de Palets que se manipulan, dentro de esta condición se encuentra la programación de los movimientos que hace el Brazo de Robot, los dos primeros movimientos *move(point[0])* y *move(point[1])* colocan al Robot en posición de espera y abren el Grip, una vez que se llega a la posición de espera entra a un ciclo y ahí permanece hasta que la entrada *input(2)* cambie de 0 a 1, una vez que se activada la entrada el Robot se mueve a la posición de recolección *move(point[2])*, al posicionarse correctamente para recolectar el Palet cierra el Grip y se activa la salida *output(2,1)* entonces entra a un ciclo donde espera a que se active la entrada *input(3)* que significa que el Grip del sistema de pistones del Plano Cartesiano ya soltó el Palet, hasta entonces se desactiva la salida *output(2,0)* y se mueve el Brazo de Robot para dejar el Palet a la Banda Transportadora el primer movimiento es *move(point[1])* cuando llega a este punto se activa la salida *output(3,1)* que le indica al Plano que ya se alejo el Brazo y se puede mover para que no hayan colisiones, después se mueve el Brazo al punto *move(point[3])*, se desactiva la salida *output(3,0)*, continua con el movimiento *move(point[4])* el cual acerca el Brazo a la Banda Transportadora, los siguientes dos movimientos *move(point[5])* y *move(point[6])* sirven para posicionar el Palet en la Bahía de entrada y salida de la Banda Transportadora, cuando se coloca el Palet en la Bahía entonces se abre el *Grip* se espera un tiempo y se retira el Brazo con el movimiento *move(point[3])*, cuando llega a este punto se regresa a la condición de seguridad para ver si se repite o se termina la secuencia.



3.1.3 Plano Cartesiano

El plano cartesiano al ser un sistema “Custom” este se configura de manera diferente al Brazo de Robot, para realizar la debida configuración de los “Joints” se definen las distancias en el eje “X” y en el eje “Y”, esto se hace después de haber instalado el firmware correspondiente, mediante un programa realizado por el usuario en el cual se bloquean los “Joints” sobrantes y se establecen los nuevos limites para los ejes “X” y “Y” reescribiendo los que vienen por defecto en el firmware. Este procedimiento se tiene que hacer cada vez que se enciende el controlador.

Características físicas del Plano Cartesiano

El sistema del Plano Cartesiano consta de una matriz de 6 filas por 7 columnas que es donde se depositan los Palets a utilizar (ver Fig. 3.19).



Fig. 3.19 Ilustración física del Plano Cartesiano

Cada entrada de la matriz consta con un “Switch” que indica la presencia de un Palet (ver Fig. 3.20), a su vez este “Switches” están conectados a diferentes entradas del PLC 215-2DP de la marca SIEMENS.



Fig. 3.20 Ilustración física de Switch en entrada del Plano Cartesiano

Para tomar algún Palet de la matriz se utiliza un sistema de pistones (ver Fig. 3.21) donde el pistón 1 extiende un Grip, el pistón 2 rota el Grip para colocarlo en la posición de tomar un Palet o posición de entregar un Palet y por ultimo el Pistón 3 que abre o cierra el Grip para la sujeción del Palet. Todo el sistema de pistones es impulsado por dos motores los cuales lo mueven tanto horizontal (Eje "X") como verticalmente (Eje "Y") en toda la superficie de la matriz (ver Fig. 3.22).

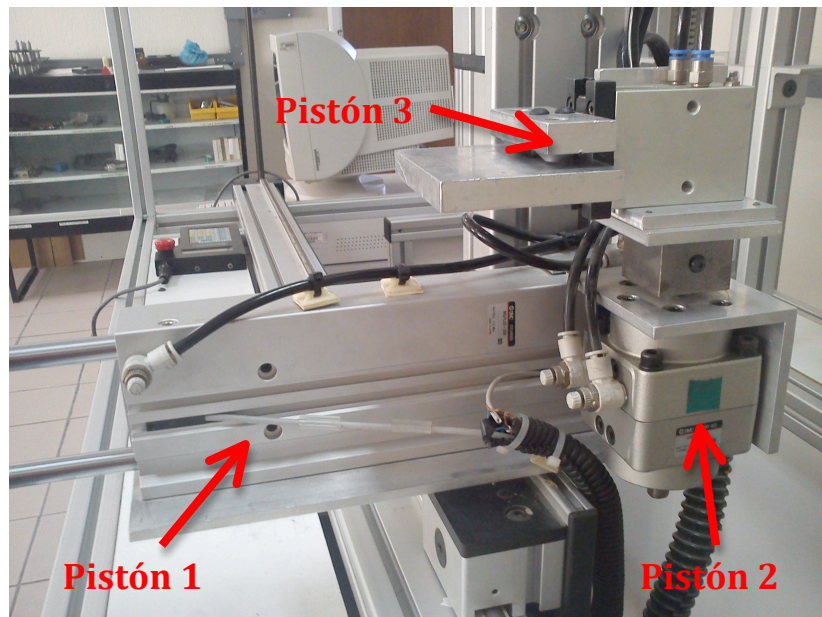


Fig. 3.21 Sistema de pistones recolector de Palets



Fig. 3.22 Motor

El Plano Cartesiano cuenta con 16 entradas y 16 salidas del PLC 215-2DP la cuales pueden ser activadas o desactivadas en el programa del Plano según sea requerido, estas entradas y salidas son utilizadas para la comunicación Profibus entre PLC's.

Las características en cuanto a los módulos entre el Brazo de Robot y el Plano Cartesiano son muy similares entres sí.

En esta etapa del proyecto debido a que también se contaba con un controlador CRS se procedió a enseñar al Robot del Plano Cartesiano las posiciones en dos dimensiones donde se colocarían los Palets.

Explicación de la programación del Plano Cartesiano

La programación del Plano Cartesiano se empezó creando una aplicación nueva dentro del programa Robcomm3 y enseñar los puntos de trabajo.

- Para crear un nueva aplicación donde se programa la secuencia del Robot es necesario seguir los pasos que a continuación se enumeran:

3) Abrir el Robcomm3 (ver Fig. 3.23).

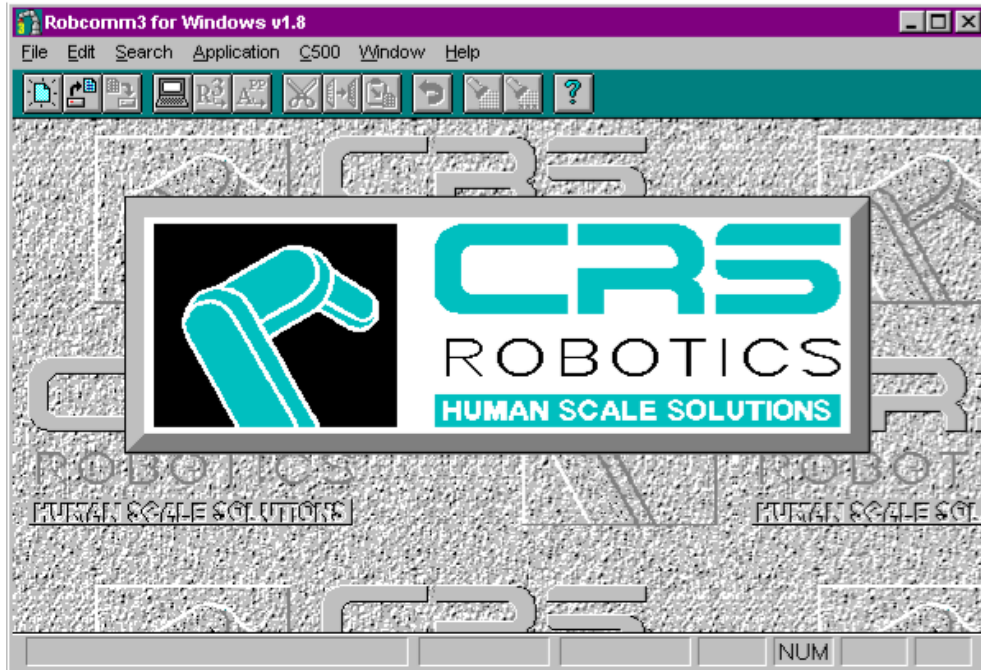


Fig. 3.23 Ventana Principal

- 4) Crear una aplicación (ver Fig. 3.24)
En el menú Application, seleccionar New Application

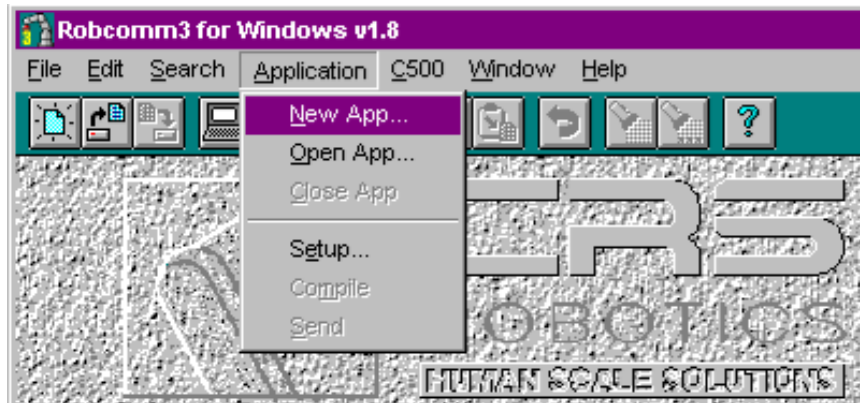


Fig. 3.24 Menú Application

En ese momento aparecerá un cuadro de mensaje en donde pedirá el nombre de la aplicación que se desea generar (ver Fig. 3.25), evitar borrar la extensión app.

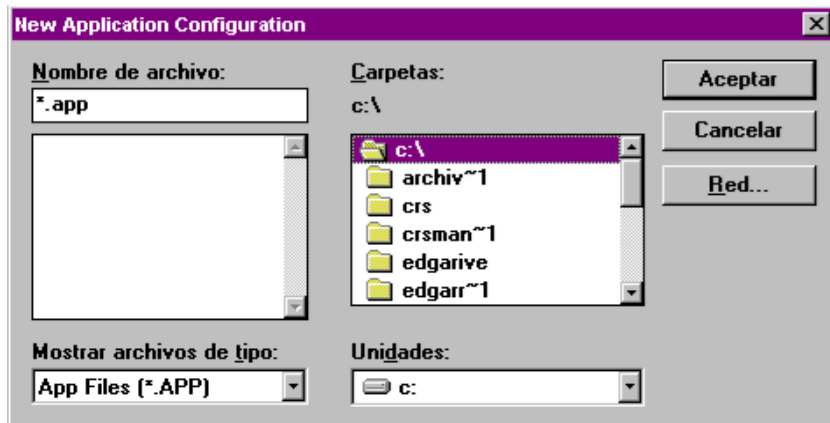


Fig. 3.25 Ventana emergente de Configuración

- Para enseñar los puntos donde se colocaron los Palets a lo largo del Plano Cartesiano y el punto de entrega donde se interactúa con el Brazo de Robot se realiza lo siguiente:

- Se debe crear una variable donde se guarde la posición.
- Se debe colocar el robot en la posición que se desea enseñar

Existen dos métodos para crear las variables y colocar el Robot en la posición que se quiere enseñar, que se explican a continuación:

Desde el Teach.

8. Comando tch
9. F1 (var)
10. Nombre de la variable
11. Especificar el tipo y la dimensión de la variable
12. Ir a posición a guardar
13. Oprimir F1(tch)
14. Si se desean más variables repetir los pasos

Desde la terminal.

Es necesario estar en una aplicación para poder lograrlo. Dentro de la aplicación se mueve el Brazo y se guarda la posición. Con el comando "here *nom*" donde *nom* es el nombre de la variable.

Una vez enseñado los puntos de trabajo se comenzó a editar la programación de la secuencia que se realizó. Para hacer esto se ingresó al editor de programas de RAPL-3 en el menú Archivo seleccionamos New y en la pantalla que aparece será donde vamos a capturar el código (ver Fig. 3.26).

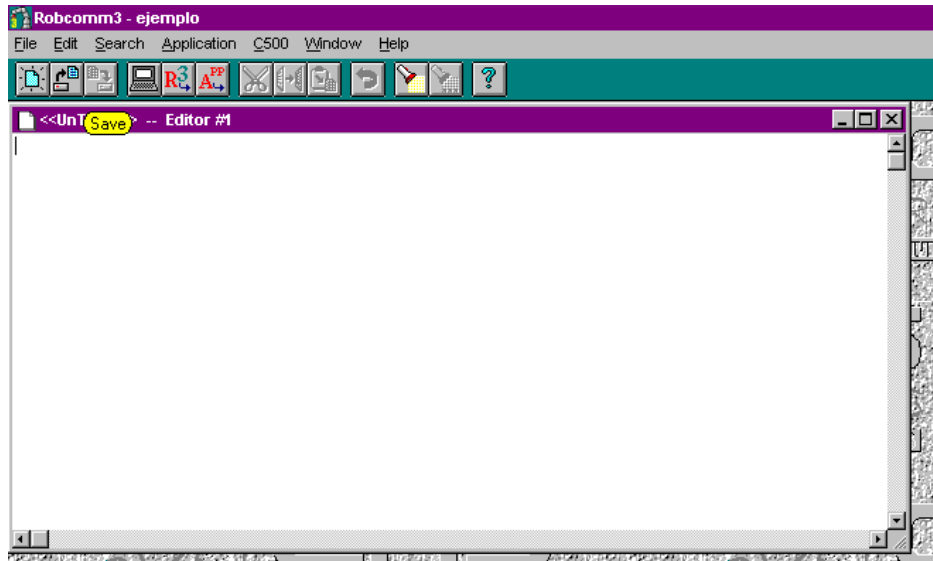
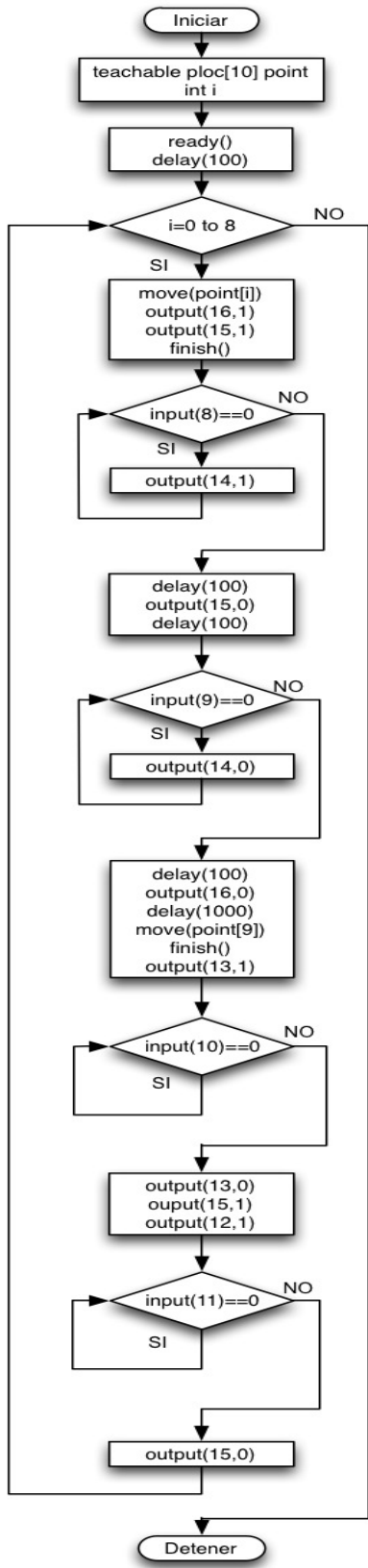


Fig. 3.26 Editor de Programas

A continuación se explica el programa final que se utilizó en el Plano Cartesiano y se muestra su diagrama de flujo.

La programación usada en el Plano Cartesiano empieza mandando el Robot a suposición *ready()* que significa que el Robot esta preparado para comenzar, espera un determinado tiempo y entra a una condición que esta encargada de repetir el proceso nueve veces que es la cantidad de palets que se manipulan, dentro de esta condición el Robot se mueve a los diferentes puntos *move(point[i])* donde se encuentran colocados los Palets iniciando con el que se encuentra en el *point[0]* hasta el *point[8]*, una vez estando en el en posición de recolección el Robot se gira *output(16,1)* y abre el Grip *output(15,1)*, después se entra un ciclo donde se extiende un pistón para alcanzar los Palets hasta que lo detecta un sensor de fin de carrera, una vez extendido el pistón espera cierto tiempo y cierra el Grip *output(15,0)*, cerrado el Grip y sujeto el Palet se retrae el pistón hasta que un sensor de inicio de carrera se activa, entonces se gira el Grip a su posición inicial *output(16,0)*, espera un tiempo hasta que gira por completo el Grip, de ahí se mueve al punto de entrega *move(point[9])*, ya colocado en la posición de entrega activa la salida *output(13,1)* para avisar al Brazo de Robot que puede ir a traer el Palet, entra en un ciclo donde espera hasta que reciba la señal de que el Brazo de Robot ya esta en posición de entrega, recibido la señal desactiva la salida *output(13,0)*, abre el Grip *output(15,1)* y activa la señal *output(12,1)* indicando que ya soltó el Palet, seguidamente entra a otro ciclo mientras espera la señal de que el Brazo se retiro de la posición de entrega y poder moverse para que no hayan colisiones, saliendo del ciclo cierra el Grip y regresa a la condición principal para repetir el procedimiento.



3.1.4 Comunicación Profibus

Comunicación Profibus

La comunicación Profibus es usada para que un numero determinado de PLC's puedan trabajar en conjunto, en este caso los PLC's usados para la comunicación fueron los siguientes:

- PLC 315-2DP, este hace la función de Maestro en la Red Profibus, y es el controlador de la Banda Transportadora.
- PLC 226, a diferencia del 315-2DP y el 215-2DP, este CPU cuenta con un modulo de expansión para comunicación Profibus EM27, ya que de fabrica no cuenta con el puerto de comunicación, este CPU esta conectado con el modulo del Brazo de Robot y funciona como Esclavo en la Red.
- PLC 215-2DP, esta conectado con el modulo del Plano Cartesiano y al igual que el CPU 226 funciona como Esclavo en la Red.

Para la comunicación Profibus se asignaron tanto para el Brazo de Robot como para el Plano Cartesiano posiciones especificas donde envían y reciben los datos del PLC Maestro, ya que estos dos son los que deben de estar comunicados entre si, pero la comunicación de estos dos no se puede hacer sin antes mandar los datos al PLC Maestro.

En las siguientes tablas se mostrarán las entradas y salidas correspondientes a cada PLC para enviar datos en la Red Profibus.

Tabla 3.4 Salidas y entradas en Red del PLC 315-2DP con el 215-2DP

	Salida "Q"	Virtual	
Salidas del 315-2DP	Q4	V0	Entradas del 215-2DP en red
	Q5	V1	
	Q6	V2	
	Q7	V3	
	Q8	V4	
	Q9	V5	
	Q10	V6	
	Q11	V7	
Salidas del 215-2DP en red	V8	Q19	Entradas del 315-2DP en red
	V9	Q20	
	V10	Q21	
	V11	Q22	
	V12	Q23	
	V13	Q24	
	V14	Q25	
	V15	Q26	

Salidas y entradas en Red del PLC 315-2DP con el 226 (ver Tabla 3.5)

Tabla 3.5 Salidas y entradas en Red del PLC 315-2DP con el 226

	Salida "Q"	Virtual	
Salidas del 315-2DP	Q0	V0	Entradas del 226 en red
	Q1	V1	
	Q2	V2	
	Q3	V3	
Salidas del 226 en red	V4	Q14	Entradas del 315-2DP en red
	V5	Q15	
	V6	Q16	
	V7	Q17	
	V8	Q18	

Bytes de salidas del PLC 315-2DP (ver Tabla 3.6) y bytes de entradas del PLC 215-2DP (ver Tabla 3.7)

Tabla 3.6 Bytes de salidas del PLC 315-2DP

byte 1_300	byte 1_200	byte 2_300	byte 2_200	byte 3_300	byte 3_200	byte 4_300	byte 4_200
4.0	0.0	5.0	1.0	6.0	2.0	7.0	3.0
4.1	0.1	5.1	1.1	6.1	2.1	7.1	3.1
4.2	0.2	5.2	1.2	6.2	2.2	7.2	3.2
4.3	0.3	5.3	1.3	6.3	2.3	7.3	3.3
4.4	0.4	5.4	1.4	6.4	2.4	7.4	3.4
4.5	0.5	5.5	1.5	6.5	2.5	7.5	3.5
4.6	0.6	5.6	1.6	6.6	2.6	7.6	3.6
4.7	0.7	5.7	1.7	6.7	2.7	7.7	3.7

Tabla 3.7 Bytes de salidas del PLC 215-2DP

byte 5_300	byte 5_200	byte 6_300	byte 6_200	byte 7_300	byte 7_200	byte 8_300	byte 8_200
8.0	4.0	9.0	5.0	10.0	6.0	11.0	7.0
8.1	4.1	9.1	5.1	10.1	6.1	11.1	7.1
8.2	4.2	9.1	5.2	10.2	6.2	11.2	7.2
8.3	4.3	10.1	5.3	10.3	6.3	11.3	7.3
8.4	4.4	9.2	5.4	10.4	6.4	11.4	7.4
8.5	4.5	11.1	5.5	10.5	6.5	11.5	7.5
8.6	4.6	9.3	5.6	10.6	6.6	11.6	7.6
8.7	4.7	12.1	5.7	10.7	6.7	11.7	7.7

A continuación se muestra la programación usada en los PLC's para su comunicación en Red Profibus (ver Tabla 3.8).

Tabla 3.8 Programación de Red Profibus

Programación para el CPU 215-2DP		Programación para el CPU 315-2DP	Programación para el CPU 226	
<i>Plano Cartesiano en posición para entregar Palet activa entrada I5.1</i>		<i>Envío de dato de CPU 215-2DP a CPU 226</i>	<i>Brazo de Robot activa salida Q2.6 para ir por el Palet</i>	
---- I5.1 ----(V8.0)	→	---- I19.0 ----(Q0.0)	→	---- V0.0 ----(Q2.6)
<i>Plano Cartesiano activa salida Q0.0 para abrir el Grip</i>		<i>Envío de dato de CPU 226 a CPU 215-2DP</i>	<i>Brazo de Robot en posición para recibir Palet activa entrada I2.6</i>	
---- V0.0 ----(Q0.0)	←	---- I14.0 ----(Q4.0)	←	---- I2.6 ----(V4.0)
<i>Plano Cartesiano activa entrada I5.2 para indicar que abrió el Grip</i>		<i>Envío de dato de CPU 215-2DP a CPU 226</i>	<i>Brazo de Robot activa salida Q2.7 para cerrar Grip</i>	
---- I5.2 ----(V9.0)	→	---- I20.0 ----(Q1.0)	→	---- V1.0 ----(Q2.7)
<i>Plano Cartesiano activa salida Q0.1 para continuar con su rutina</i>		<i>Envío de dato de CPU 226 a CPU 215-2DP</i>	<i>Brazo de Robot activa entrada I2.7, indica su movimiento hacia la Banda Transportadora</i>	
---- V1.0 ----(Q0.1)	←	---- I15.0 ----(Q5.0)	←	---- I2.7 ----(V5.0)

Conclusión y Resultados

Integrar un sistema de tres partes que trabajan en conjunto debido a una configuración de comunicación Profibus que intercambian señales de acuerdo a la posición que se encuentran e interactuar entre si, favoreció a la simulación de proceso industrial que se llevó acabo a lo largo del proyecto de residencia profesional.

Los resultados obtenidos al termino del proyecto fueron el obtener un sistema de automatización capaz de realizar un proceso industrial de forma automática, desarrollándose algoritmos de programación en diferentes tipos de lenguajes y Plataformas, como son los tres tipos de PLC's utilizados para el manejo de la Banda Transportadora y comunicación Profibus y los módulos controladores de CRS que manejan los Robots tanto en el plano cartesiano como el Brazo de Robot.

Bibliografía

- http://www.rocatek.com/forum_plc2.php
- Lenguajes de Programación de PLC's
- Manual de Prog. Robots CRS
- *C500C Controller User Guide*
- RAPL-3 Language Reference Guide
- A465 Robot Arm User Guide
- Pilar Mengual, STEP 7: UNA MANERA FACIL DE PROGRAMAR PLC DE SIEMENS. Ed. Alfaomega
- C.T. Jones, STEP 7 in 7 Steps. Ed. Brilliant Training

Anexos

Código de Programación de Plano Cartesiano.

```
Main //Inicia código principal

teachable ploc[10] point
int i //Se establecen las variables

ready() //Prepara el robot para comenzar
delay (100) //con un retraso de 100 ms

for i=1 to 9 //Inicia un ciclo "for" que se repetirá
// nueve veces

    move(point[i]) //Mueve el robot al punto "i"
    output(16,1) //Gira el grip
    output(15,1) //Abre el grip
    finish() //Termina el primer movimiento

    while(input(8)==0) //Extiende el pistón del grip
    output(14,1) //acercándolo al palet para sujetarlo,
    end while //hasta que se active el sensor de fin
//carrera

    delay(100)
    output(15,0) //Cierra el grip y se sujeta el palet
    delay(100)

    while(input(9)==0) //Contrae el pistón del grip hasta
    output(14,0) //que se active el sensor de inicio de
    end while //carrera
    delay(100)

    output(16,0) //Gira el grip
    delay(1000)

    move(point[9]) //Mueve el robot al punto de entrega
    finish()

    output(13,1) //Activa entrada PLC I5.1 indicando que
//se encuentra en posición de entrega

    while (input(10)==0) //El robot espera a que se active la salida
```

```

end while // del PLC Q0.0 indicando que el brazo
//robot llevo al punto de entrega

output(15,1) //Abre grip
output(12,1) //Activa entrada del PLC I5.2 indicando
//que se soltó el palet

while(input (11)==0) //Espera que se active la salida del PLC
end while // Q0.1 indicando que es seguro continuar
//con la secuencia

output(15,0) //Cierra grip
end for //Termina ciclo "for" y se repite el proceso

end main //Termina código principal

```

Código de Programación de Brazo de Robot

```
Main //Inicia código principal

teachable cloc[7] point //Se establecen las variables
int i

ready() //Prepara el robot para comenzar
delay (100) //con un retraso de 100 ms

for i=1 to 9 //Inicia un ciclo "for" que se repetirá
// nueve veces

    move(point[0]) //Mueve el brazo robot a una posición
    move(point[1]) //de espera

    grip_open() //Abre el grip
    finish()

    printf("Esperando Palet {} en posición de entrega", i) //Muestra en pantalla

    while input(2)==0 //El brazo robot espera que la salida del
    printf(".") // PLC Q2.6 se active para moverse
    delay(500) //a posición donde recibe el palet
    end while

    move(point[2]) //Mueve el brazo robot al punto donde
    finish() //recibe los palets

    output(2,1) //Activa entrada del PLC I2.6 indicando
//que se encuentra en posición

    while input(3)==0 //Espera mientras salida del PLC Q2.7 se
    end while //se activa para sujetar palet

    output(2,0) //Desactiva entrada del PLC I2.6
    grip_close() //Cierra grip sujetando el palet
    delay(200) //Espera 200 ms
    move(point(1)) //Empieza movimiento a posición de
    finish() //entrega

    output(3,1) //Activa entrada del PLC I2.7 para saber
//que el plano cartesiano puede seguir
//con la secuencia
```

```

move(point[3])           //Mueve brazo robot al punto 3
output(3,0)             //Desactiva entrada del PLC i2.7
move(point[4])          //Posiciona al brazo robot en punto
finish()                //de entrega

move(point[5])          //Mueve y acomoda el brazo
finish()                //robot para entregar palet a la
move(point[6])          //banda transportadora
finish()
grip_open()             //Abre grip para soltar palet
delay(100)              //espera 100 ms para moverse al
move(point[3])          //punto de espera y repetir la
                        //la secuencia

end for                 //Termina ciclo "for" y se repite el proceso

end main                //Termina código principal

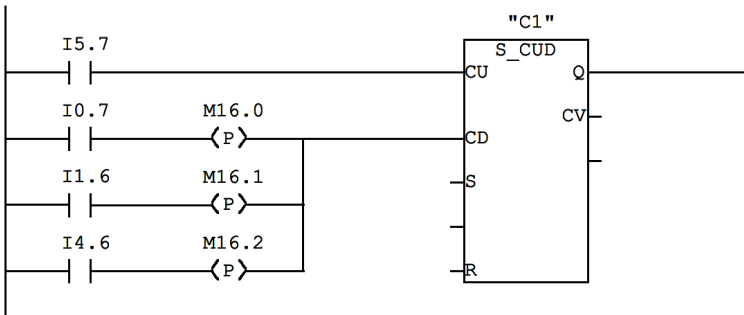
```

Código de Programación KOP para la Banda Transportadora.

Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
	In				
	out				
	in_out				
	temp				

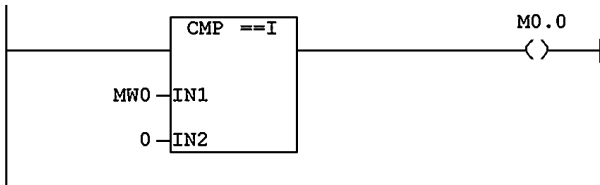
Bloque: FC5

Segm.: 1
 CONTADOR DE PALETS



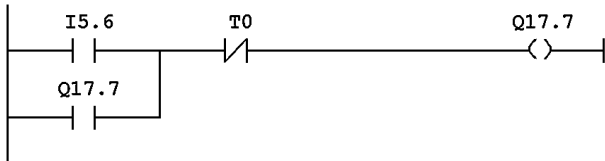
Información del símbolo
 C1 C1

Segm.: 2
 COMPARA PALETS DE ENTRADA CON PALETS CAMBIO DE BANDA



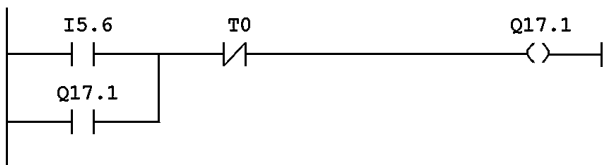
Segm.: 3 BAHIA ENTRADA Y SALIDA

ACTIVA PLATAFORMA BAHIA ENTRADA Y SALIDA



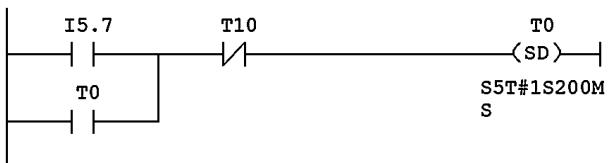
Segm.: 4

ACTIVA CADENAS BAHIA ENTRADA Y SALIDA



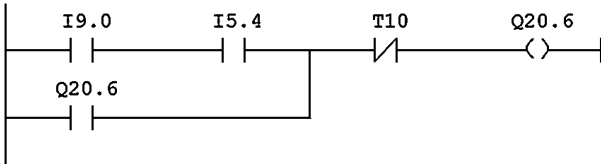
Segm.: 5

TEMPORIZADOR BAJA PLATAFORMA ENTRADA Y SALIDA



Segm.: 6

ACTIVA PISTON ANTES DE BAHIA ENTRADA Y SALIDA



Segm.: 7

TEMPORIZADOR BAJA PLATAFORMA ENTRADA Y SALIDA



Segm.: 8 BANDAS

TEMPORIZADOR PISTON CAMBIO DE BANDA 1



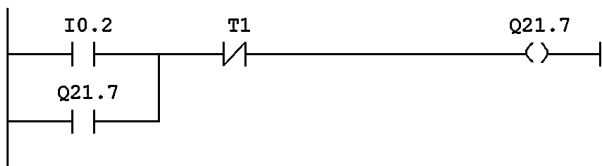
Segm.: 9

TEMPORIZADOR PISTON CAMBIO DE BANDA 2



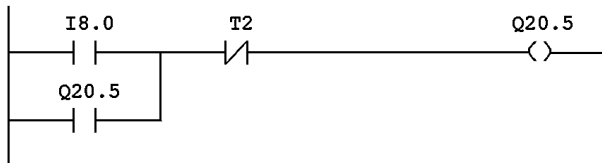
Segm.: 10

ACTIVA PISTON CAMBIO DE BANDA 1



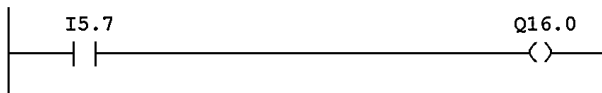
Segm.: 11

ACTIVA PISTON CAMBIO DE BANDA 2



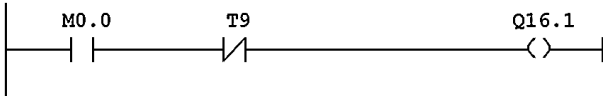
Segm.: 12

ACTIVA CADENAS DE BANDA



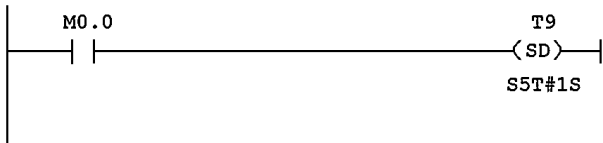
Segm.: 13

DESACTIVA CADENAS DE BANDA



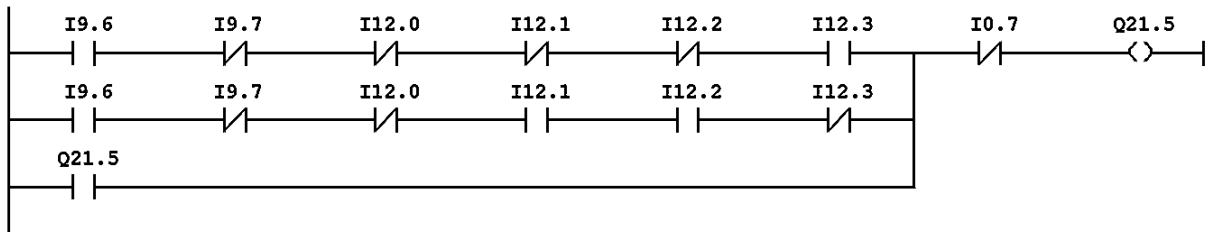
Segm.: 14

TEMPORIZADOR DESACTIVA RELE BANDA



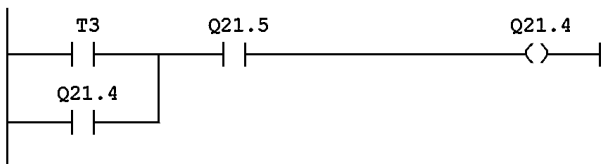
Segm.: 15 BAHIA 1

ACTIVA PISTON BAHIA 1



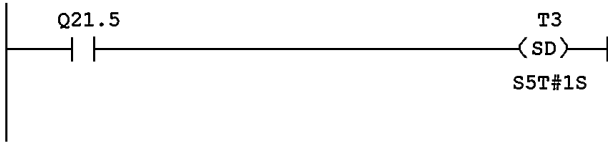
Segm.: 16

ACTIVA PLATAFORMA BAHIA 1



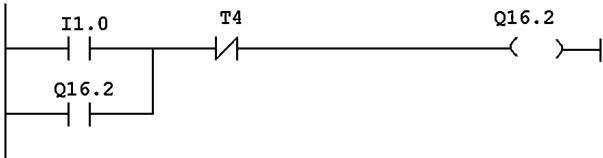
Segm.: 17

TEMPORIZADOR PLATAFORMA BAHIA 1



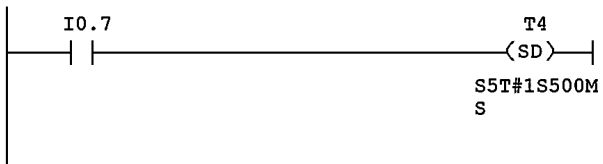
Segm.: 18

ACTIVA CADENAS BAHIA 1



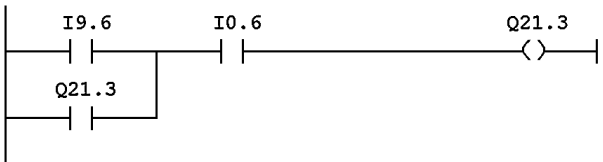
Segm.: 19

TEMPORIZADOR CADENAS BAHIA 1



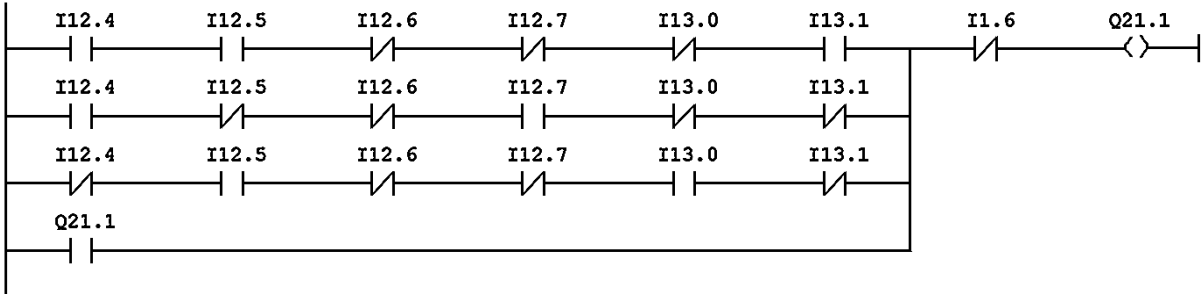
Segm.: 20

ACTIVA PISTON ANTES DE BAHIA 1



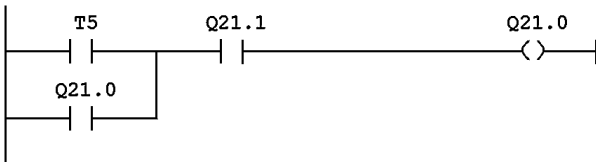
Segm.: 21 BAHIA 2

ACTIVA PISTON BAHIA 2



Segm.: 22

ACTIVA PLATAFORMA BAHIA 2



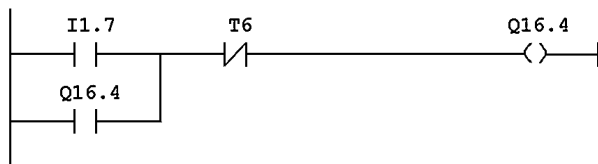
Segm.: 23

TEMPORIZADOR PLATAFORMA BAHIA 2



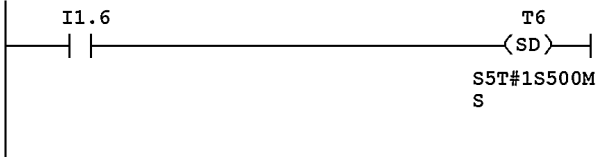
Segm.: 24

ACTIVA CADENAS BAHIA 2



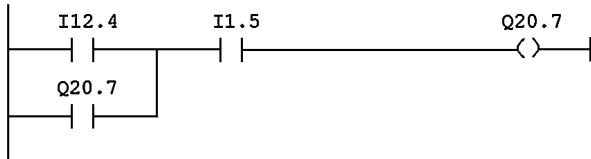
Segm.: 25

TEMPORIZADOR CADENAS BAHIA 2



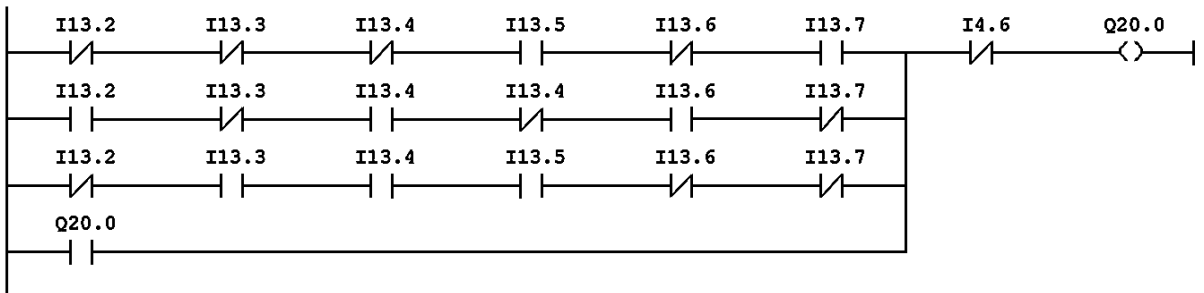
Segm.: 26

ACTIVA PISTON ANTES DE BAHIA 2



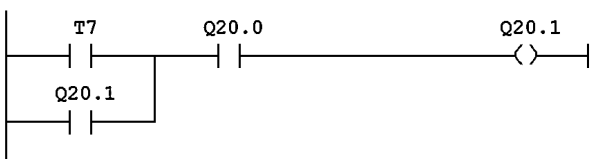
Segm.: 27 BAHIA 3

ACTIVA PISTON BAHIA 3



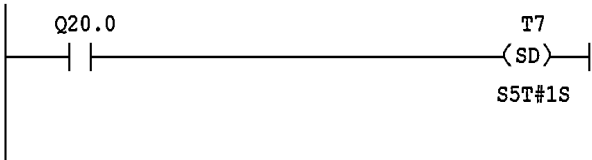
Segm.: 28

ACTIVA PLATAFORMA BAHIA 3



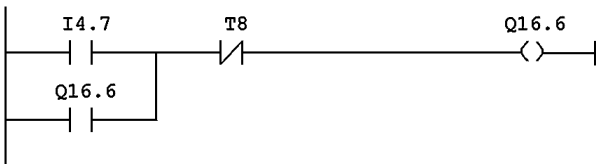
Segm.: 29

TEMPORIZADOR PLATAFORMA BAHIA 3



Segm.: 30

ACTIVA CADENAS BAHIA 3



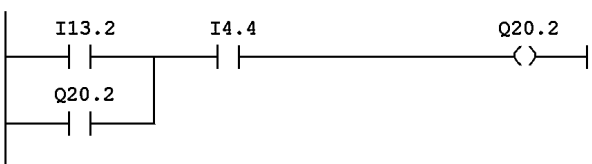
Segm.: 31

TEMPORIZADOR CADENAS BAHIA 3



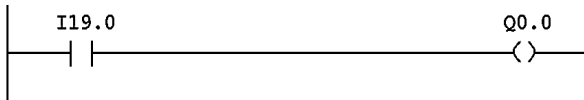
Segm.: 32

ACTIVA PISTON ANTES DE BAHIA 3



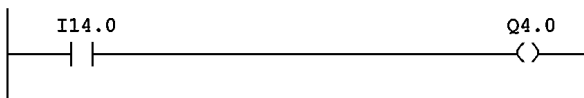
Segm.: 33

PLC 215 EN ESPERA PARA ENTREGAR PALET A PLC 226



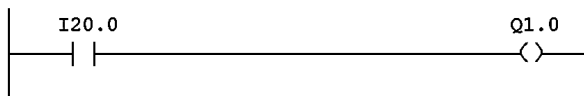
Segm.: 34

PLC 226 EN POSICION PARA QUE PLC 215 ABRA GRIP



Segm.: 35

PLC 215 ABRIO GRIP PLC 226 PUEDE IR A DEJAR PALET



Segm.: 36

PLC 226 MANDA SEÑAL PARA QUE PLC 215 PUEDA CONTINUAR

