



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE CONCEPCIÓN – REY BALDUINO DE BELGICA

PROGRAMACIÓN EFICIENTE E INTELIGENTE DE UN PROCESO DE MANUFACTURA FLEXIBLE

Trabajo de Titulación para optar al
Título de Ingeniero de Ejecución en
Control e Instrumentación Industrial

Alumno:

Christopher Alexander Morales Herrera

Profesor Guía:

Sr. Rodrigo Méndez

2018

DEDICATORIA

Este trabajo va dedicado especialmente a mi madre Rosa Herrera y mi padre Emiliano Morales, quienes desde pequeño me cuidaron, entregaron su cariño y apoyo incondicional, me inculcaron el espíritu de superación y esfuerzo para lograr las metas que nos proponemos. Además, quiero dedicar este trabajo a mi novia Viviana, quien me ha apoyado en cada paso durante la carrera en la universidad. Finalmente quiero agradecer a mi profesor guía, por el gran apoyo que me ha brindado durante este trabajo y por su disposición a ayudar en todo momento.

RESUMEN

Hoy en día, es común observar que diversos procesos de manufactura industriales son automatizados, con la finalidad de aumentar la productividad, reducir costos de operación, disminuir el sobre esfuerzo del personal, evitando enfermedades profesionales.

La Universidad Técnica Federico Santa María sede Concepción, cuenta con un laboratorio de celda flexible, el cual está compuesto por siete estaciones de procesos de manufactura, que representan etapas fundamentales en la industria, como lo son la producción, procesos robotizados, adquisición, transporte, inspección, almacenamiento y despacho de productos.

El siguiente trabajo muestra la integración de estas estaciones, generando una lógica de control sincronizada entre las distintas estaciones. Basándose en una estación principal encargada de comandar y sintonizar los distintos procesos.

En cuanto a programación, este trabajo trata de explicar de forma sencilla las ventajas de utilizar un lenguaje adecuado para programar distintos procesos. Obteniendo como resultado una celda de manufactura flexible trabajando de forma sincronizada en cada una de las estaciones, comandas por bloques de instrucciones acotados y fácil de entender, lo que a su vez trae enormes beneficios al momento de buscar una falla.

ÍNDICE DE CONTENIDOS

RESUMEN	I
ÍNDICE DE CONTENIDOS	II
ÍNDICE DE FIGURAS	III
ÍNDICE DE TABLAS	IV
INTRODUCCIÓN	1
OBJETIVO GENERAL	2
OBJETIVOS ESPECÍFICOS	2
CAPÍTULO 1: DEFINICIÓN Y CONCEPTOS GENERALES	3
1.1 DESCRIPCIÓN Y NOCIONES BÁSICAS	3
1.1.1 Producto	3
1.1.2 Proceso industrial	3
1.1.3 Sistema	3
1.1.4 Manufactura	3
1.1.5 Sistema de Manufactura Flexible	4
1.2 PARTES DE UNA CELDA DE MANUFACTURA.....	4
1.2.1 Máquina de Control Numérico por Computador (CNC)	5
1.2.2 Brazo Robótico.....	6
1.2.3 Adquisición	6
1.2.4 Distribución	7
1.2.5 Transporte.....	8
1.2.6 Inspección y Control de Calidad.....	8
1.2.7 Almacenamiento	9
1.3 LENGUAJES DE PROGRAMACIÓN	10
1.3.1 Normativa	11
1.3.2 Tipos de Lenguajes.....	11
1.4 BUSES DE CAMPO	14
1.4.1 Conceptos Generales	14
1.4.2 Tipos de Buses.....	14
1.4.3 Bus de Campo AS-interface	15
CAPITULO 2: PROGRAMACIÓN DE ESTACIONES DE DISTRIBUCIÓN Y TRANSPORTE DE PIEZAS.....	18
2.1 ESTACIÓN DE ADQUISICIÓN	19
2.1.1 Partes de la estación.....	19
2.1.2 Lógica de operación	22

2.1.3	Levantamiento	23
2.1.4	Programación en TIA Portal	23
2.2	ESTACIÓN DE DISTRIBUCIÓN	26
2.2.1	Partes de la estación.....	26
2.2.2	Lógica de Operación	30
2.2.3	Levantamiento	31
2.2.4	Programación en TIA Portal	32
2.3	ESTACIÓN DE TRANSPORTE.....	35
2.3.1	Partes de la estación.....	36
2.3.2	Lógica de Operación	42
2.3.3	Levantamiento	44
2.3.4	Programación en TIA Portal V14 SP1.....	44
2.4	ESTACIÓN DE INSPECCIÓN	53
2.4.1	Módulo de visión de FMS	53
2.4.2	Programación de cámara de Visión	55
2.5	ESTACIÓN DE ALMACENAMIENTO.....	63
2.5.1	Estación de almacenamiento AS-RS.....	64
2.5.2	Partes de la estación AS-RS.....	64
2.5.3	Lógica de operación	73
2.5.4	Levantamiento	74
2.5.5	Programación en TIA Portal	75
CAPITULO 3: ROBOTS INDUSTRIALES Y SU APLICACIÓN EN CELDAS FLEXIBLES DE MANUFACTURA.....		84
3.1	ROBOTS INDUSTRIALES.....	85
3.2	BRAZO ROBÓTICO MITSUBISHI RV-2SDB.....	86
3.2.1	Unidad procesadora	88
3.2.2	Teaching Pendant.....	89
3.3	SOFTWARE CIROS STUDIO.....	90
3.4	PROGRAMACIÓN DE BRAZO ROBÓTICO.....	91
3.4.1	Creación de un proyecto.....	92
3.4.2	Interfaz de programación	93
3.4.3	Programación y resultados	93
CAPITULO 4: CONCLUSIONES Y REFERENCIAS.....		95
4.1	RESULTADOS.....	96
4.2	CONCLUSIONES	96
4.3	BIBLIOGRAFÍA	97

ÍNDICE DE FIGURAS

Figura 1-1. Secuencia de proceso FMS USM Concepción	4
Figura 1-2. Fresadora CNC.....	5
Figura 1-3. Brazo robótico de la FMS	6
Figura 1-4. Estación de adquisición de la FMS	7
Figura 1-5. Estación de distribución de la FMS	7
Figura 1-6. Estación de transporte de la FMS	8
Figura 1-7. Estación de control de calidad de la FMS	9
Figura 1-8. Estación de almacenamiento de la FMS	9
Figura 1-9. Extracto de código de programación orientada a computador.....	10
Figura 1-10. Extracto de programación orientada a PLC.....	10
Figura 1-11. Extracto de programación SFC	11
Figura 1-12. Extracto de programación por bloques	12
Figura 1-13. Extracto de programación ladder	13
Figura 1-14. Extracto de programación texto estructurado	13
Figura 1-15. Programación lista de instrucciones	14
Figura 1-16. Pirámide de CIM	15
Figura 1-17. Cable AS-i	16
Figura 2-1. Estación de adquisición	19
Figura 2-2 Contenedor de entrada	20
Figura 2-3 Cilindro neumático	21
Figura 2-4 Brazo neumático	21
Figura 2-5 Diagrama de flujo estación de adquisición	22
Figura 2-6 Imagen de configuración de hardware estación adquisición	24
Figura 2-7 Variables estación de adquisición	24
Figura 2-8 Bloque de función.....	25
Figura 2-9 Programación estación de adquisición	25
Figura 2-10. Estación de distribución.....	26
Figura 2-11. Plataforma de entrada.....	27
Figura 2-12. Cilindro neumático.....	27
Figura 2-13. Cinta de rechazo	28
Figura 2-14. Elevador.....	28
Figura 2-15. Comparador analógico	29
Figura 2-16. Cinta transportadora por aire	30
Figura 2-17. Diagrama de flujo estación de distribución	31
Figura 2-18. Imagen de configuración de hardware estación distribución	32
Figura 2-19. Variables estación de distribución	33
Figura 2-20. Agregar bloque de función FB	33
Figura 2-21. Programación estación de distribución.....	34
Figura 2-22. Estación de transporte	36
Figura 2-23. Tablero de control	36
Figura 2-24. Esclavo AS-i SYS link	38
Figura 2-25. Esclavo AS-i Sensores.....	38
Figura 2-26. Esclavo AS-i para pistón neumático	39

Figura 2-27. Electroválvula neumática.....	39
Figura 2-28. Cilindro neumático.....	40
Figura 2-29. Sensores de proximidad	40
Figura 2-30. Sensor láser	41
Figura 2-31. Diagrama de flujo estación de transporte desde ST1 a ST3	43
Figura 2-32. Diagrama de flujo estación de transporte desde ST4 ST6.....	43
Figura 2-33. Hardware estación Transporte.....	45
Figura 2-34. Ciclo de trabajo de una CPU Simatic S7-300	46
Figura 2-35. Direcciones esclavos AS-i	47
Figura 2-36. Lectura y escritura de esclavos AS-i.....	48
Figura 2-37 Programa estación 1	49
Figura 2-38. Estación número 2.....	49
Figura 2-39 Programa estación 2	50
Figura 2-40. Bloque de datos	51
Figura 2-41. Etiqueta de carros	51
Figura 2-42 Lectura etiqueta de carros	52
Figura 2-43. Estación de visión USM.....	53
Figura 2-44 Conexiones eléctricas.....	54
Figura 2-45. Conexión mediante interfaz SYS Link	55
Figura 2-46. Software SBO-DeviceManager, dispositivos disponibles en la red	56
Figura 2-47. Interfaz de configuración de red	57
Figura 2-48. Ventana "Welcome to CheckOpti"	58
Figura 2-49. Ventana de propiedades del proyecto.....	58
Figura 2-50. Barra de herramientas CheckOpti	59
Figura 2-51. Ventana "Part List"	59
Figura 2-52. Ventana "Communication Manager"	60
Figura 2-53. Barra de funciones de Part Contour.....	60
Figura 2-54. Ventana de selección de filtro RGB.....	60
Figura 2-55. Herramienta de propiedades I/O Access	61
Figura 2-56. Interfaz "Check program manager".....	62
Figura 2-57. Ventana "System parameter"	62
Figura 2-58. Almacén automático	63
Figura 2-59. Estación de almacenamiento.....	64
Figura 2-60. Rack de almacenamiento.....	65
Figura 2-61. Brazo móvil	66
Figura 2-62. Motor eléctrico 24V.....	67
Figura 2-63. Encoder.....	67
Figura 2-64. Límite de carrera	68
Figura 2-65. Driver de motores	69
Figura 2-66. Llave de seguridad.....	69
Figura 2-67. Panel de control AS-RS	70
Figura 2-68. Interfaz SYS Link.....	70
Figura 2-69. Relés de seguridad	71
Figura 2-70. Interfaz de conexiones	71
Figura 2-71. PLC estación AS-RS	72
Figura 2-72. Lógica de operación de almacén	73
Figura 2-73. Mapa del almacén.....	76

Figura 2-74. Tabla de coordenadas en TIA Portal	78
Figura 2-75. Configuración contadores	79
Figura 2-76. Dirección de contadores	79
Figura 2-77. Contador rápido	81
Figura 2-78. Subrutina de disponibilidad	81
Figura 2-79. Subrutina de posicionamiento	82
Figura 2-80. Subrutina de almacenamiento	83
Figura 2-81. Subrutina de calibración inicial	83
Figura 3-1. Analogía brazo robótico - brazo humano	85
Figura 3-2. Robots en la industria automotriz	86
Figura 3-3. Brazo robótico	86
Figura 3-4. Grados de libertad brazo robótico	87
Figura 3-5. Controlador CR1DA-711	88
Figura 3-6. Teaching pendant	89
Figura 3-7. Software CIROS	91
Figura 3-8. Diagrama de flujo de brazo robótico	91
Figura 3-9. Nuevo proyecto	92
Figura 3-10. Modelo de robot	92
Figura 3-11. Interfaz de programación	93
Figura 3-12. Extracto de programación MELFA BASIC	94
Figura 3-13. Referenciación de puntos	94

ÍNDICE DE TABLAS

Tabla 1-1. Detalles Bus AS-i.....	17
Tabla 2-1 Sensores y actuadores estación de adquisición.....	23
Tabla 2-2. Sensores y actuadores estación de distribución	31
Tabla 2-3. Ocupación estación de transporte	35
Tabla 2-4. Sensores estación de transporte.....	44
Tabla 2-5 Esclavos admitidos por tipo datos	46
Tabla 2-6. Entradas PLC estación AS-RS	74
Tabla 2-7. Salidas PLC de almacén.....	75
Tabla 2-8. Coordenadas de almacén.....	77
Tabla 2-9. Dirección de contadores rápidos	80
Tabla 3-1. Grados de libertad brazo robótico	87
Tabla 3-2. Descripción de teclas teaching pendant	89

INTRODUCCIÓN

A lo largo de la historia se ha buscado nuevos métodos para optimizar y mejorar la calidad de los productos, comenzando con fabricación artesanal, hasta llegar a la producción totalmente automatizada.

Cabe mencionar que, debido a la gran demanda mundial existente, es inevitable automatizar diversos procesos de manufactura, enfocados en cumplir con el volumen y tiempos establecidos por las necesidades de los consumidores.

Es por esta razón que surgen los sistemas de manufactura flexibles (FMS), los cuales tienen como objetivo principal producir, ensamblar, controlar la calidad, almacenar y despachar productos. Además de la capacidad de adaptarse al acelerado cambio en las necesidades de los consumidores y de la creciente competencia mundial.

Por otro lado, los FMS garantizan mejorar la calidad de los productos terminados, junto con un aumento en la productividad. Por otro lado, promueven una reducción de los costos de producción y enfermedades profesionales por tareas repetitivas en los trabajadores.

De acuerdo con lo previamente expuesto, es que surge la necesidad de llevar a cabo la presente memoria, que tiene como objetivo la programación de una celda flexible de manufactura, utilizando como filosofía de programación "eficiencia y flexibilidad de los procesos".

OBJETIVO GENERAL

- Automatizar de manera eficiente e inteligente un proceso de manufactura flexible.

OBJETIVOS ESPECÍFICOS

- Estudiar el hardware y software presentes en la celda flexible Festo MPS 500-FMS.
- Identificar piezas por color.
- Programar un almacenamiento y despacho selectivo.
- Programar de manera eficiente un brazo robótico Mitsubishi.
- Sincronizar las distintas estaciones de la celda flexible.

CAPÍTULO 1: DEFINICIÓN Y CONCEPTOS GENERALES

1.1 DESCRIPCIÓN Y NOCIONES BÁSICAS

Con el objetivo de comprender el funcionamiento de una FMS, es que a continuación se definirán los conceptos básicos y relevantes para comenzar el estudio.

1.1.1 Producto

Se puede definir como "aquello que ha sido fabricado", es decir el resultado de diversas etapas de modificación de la materia prima. Un producto puede tener distintas características y formas de evaluación, dependiendo netamente del mercado.

1.1.2 Proceso industrial

Un proceso industrial es un conjunto de actividades que tienen en común la modificación de un producto. En otras palabras, es la totalidad de etapas necesarias para poder transformar una determinada materia prima en un objeto deseado.

1.1.3 Sistema

Es un grupo de elementos componentes, relacionados entre sí que tiene como finalidad trabajar en conjunto para obtener o garantizar un objetivo común.

1.1.4 Manufactura

Es el proceso de desarrollo o producción de un artículo. Este puede ser realizado por mano de obra de individuos o con la colaboración de máquinas.

1.1.5 Sistema de Manufactura Flexible

Un sistema de manufactura flexible, o Flexible Manufacturing System (FMS) en inglés, es un conjunto de componentes electromecánicos, que trabajan de manera coordinada para el logro de un producto.

Está compuesta por grupos de estaciones de maquinado modulares que generalmente son máquinas CNC, interconectadas entre si mediante un sistema automatizado de transporte y almacenamiento de materiales.

Para poder determinar que un sistema de manufactura es flexible se deben cumplir los siguientes criterios:

- a. Procesar distintos formatos de piezas
- b. Aceptar cambios en el programa de producción
- c. Aceptar la introducción de nuevos diseños de productos

1.2 **PARTES DE UNA CELDA DE MANUFACTURA**

La universidad Técnica Federico Santa María, cuenta con un laboratorio de celda flexible de la marca Festo, modelo MPS 500 FMS. A continuación, se verá en detalle cada una de las partes que forman este sistema, comenzando con el siguiente diagrama para graficar su secuencia de operación.

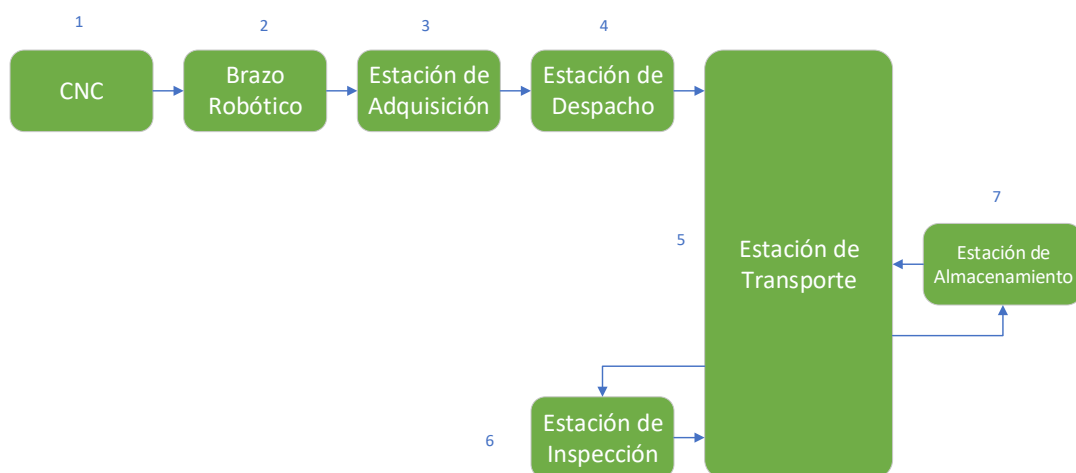


Figura 1-1. Secuencia de proceso FMS USM Concepción

1.2.1 Máquina de Control Numérico por Computador (CNC)

Son máquinas electromecánicas que se pueden programar mediante un computador para realizar diversas operaciones. Su principio de funcionamiento está basado en controlar la posición de una herramienta, generalmente en un sistema de coordenadas de tres ejes. Esta herramienta adicionalmente posee un eje para generar su rotación y así producir el desgaste de la materia prima.

Como consecuencia, se obtiene un producto terminado, generalmente recuperado por un robot industrial.



Figura 1-2. Fresadora CNC

1.2.2 Brazo Robótico

Debido a que las FMS son sistemas autónomos, necesitan de robots para realizar ciertas tareas que haría un personal humano. Los brazos robóticos son robots del tipo estacionarios, su principio de funcionamiento es el desplazamiento de un actuador dentro de un sistema de coordenadas mediante el movimiento de sus extremidades.

Cada articulación es controlada por un servomotor y tiene un rango de movimiento determinado, llamado eje de libertad. En el caso de la MPS 500 el brazo robótico cuenta con seis grados de libertad, lo que permite realizar movimientos precisos en un espacio reducido.



Figura 1-3. Brazo robótico de la FMS

1.2.3 Adquisición

La estación de adquisición opera como contenedor de entrada al proceso de manufactura. El brazo robótico toma la pieza terminada en la CNC y la almacena en el depósito. Cuando la estación de distribución demanda productos, el brazo neumático toma la pieza y la entrega a la siguiente estación.



Figura 1-4. Estación de adquisición de la FMS

1.2.4 Distribución

La estación de distribución tiene como finalidad adaptar dos niveles de la FMS y para llevar a cabo esta operación, cuenta con un elevador neumático. Posee, además, la capacidad de determinar ciertas características de un producto como el tipo de material y altura rechazando la pieza cuando no cumple con lo requerido. Si el producto pasa el chequeo, es despachada por una cinta neumática hacia la estación de transporte.



Figura 1-5. Estación de distribución de la FMS

1.2.5 Transporte

La estación de transporte cumple la función de desplazar los carros contenedores de piezas. Cuenta con seis estaciones, donde cada una es capaz de trabajar de forma independiente y paralela. Cada estación posee con un sistema de identificación de carro, de esta forma se pueden realizar procesos específicos para cada tipo de producto. Los sensores y actuadores se controlan mediante un bus de campo AS-interface.

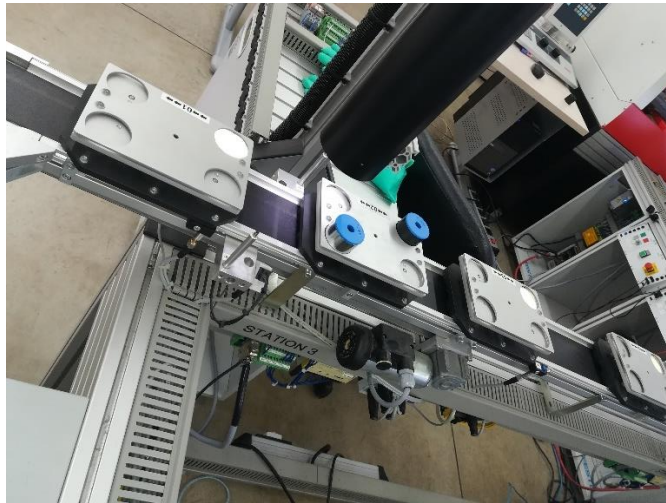


Figura 1-6. Estación de transporte de la FMS

1.2.6 Inspección y Control de Calidad

Esta etapa tiene como función principal analizar ciertas características del producto y de esta manera realizar una rotulación digital de sus propiedades, es decir, permite realizar un control de calidad. El sistema se basa en una cámara, la cual se encarga de tomar fotografías para luego realizar el análisis mediante software. Para poder establecer los parámetros correctos que se deben obtener (color, brillo, forma, entre otros), la cámara se debe programar desde un computador mediante el puerto Ethernet incorporado.



Figura 1-7. Estación de control de calidad de la FMS

1.2.7 Almacenamiento

La unidad de almacenamiento tiene como finalidad contener lotes de productos, para poder realizar esta operación se basa en controlar un mecanismo robotizado de tres ejes. Los productos se distribuyen según sus características, es decir, se almacenan según la etiqueta generada en la estación de inspección.

Su proceso es totalmente automático, lo que permite tener una rápida velocidad de respuesta para la entrada y salida del producto.



Figura 1-8. Estación de almacenamiento de la FMS

1.3 LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas, por ejemplo, un computador.

A diferencia del lenguaje utilizado por los computadores (Figura 1-9), los PLCs se basan en lenguajes de programación con estructuras más gráficas y simples de entender (Figura 1-10).

```
role_id' => $role_details['id'],
'resource_id' => $resource_details['id'],
);
if ( $this->rule_exists( $resource_details['id'], $role_details['id'] ) ) {
    if ( $access == false ) {
        // Remove the rule as there is currently no need for it
        $details['access'] = !$access;
        $this->sql->delete( 'acl_rules', $details );
    } else {
        // Update the rule with the new access value
        $this->sql->update( 'acl_rules', array( 'access' => $access ) );
    }
}
foreach( $this->rules as $key=>$rule ) {
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {
        if ( $access == false ) {
            unset( $this->rules[ $key ] );
        } else {
            $this->rules[ $key ]['access'] = $access;
        }
    }
}
```

Figura 1-9. Extracto de código de programación orientada a computador

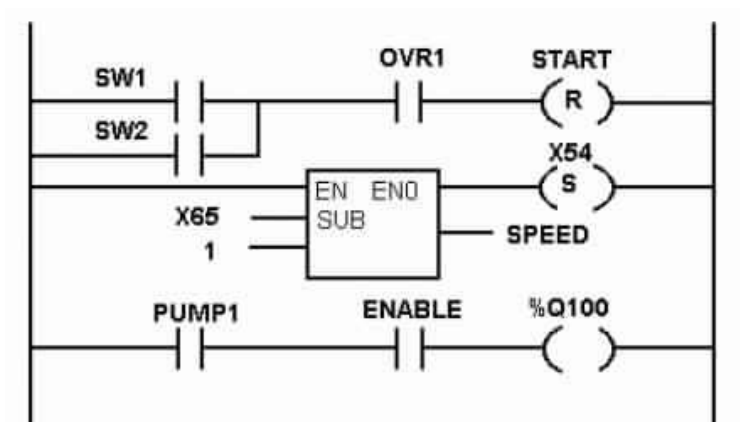


Figura 1-10. Extracto de programación orientada a PLC

1.3.1 Normativa

Con el paso del tiempo, los lenguajes de programación de PLC se han ido masificando y junto con esto, se han establecido normativas que los regulan. El estándar IEC 31131-3 es quien establece los distintos lenguajes.

1.3.2 Tipos de Lenguajes

La comisión electrotécnica internacional (IEC) ha definido cinco lenguajes de programación para PLC. Tres de ellos son del tipo visual, y dos son escritos. Los primeros, basan su programación mediante símbolos gráficos, mientras que los segundos, poseen una estructura compuesta por un listado de sentencias cíclicas.

A continuación, se detalla cada uno de los lenguajes definidos por la IEC.

A. Diagrama de Funciones Secuenciales (SFC)

Corresponde a un método gráfico de modelado y descripción de sistemas de automatización secuencial, donde el estado de cada etapa depende de las condiciones anteriores. Cuenta con dos elementos principales, las Acciones y Transiciones. Las Acciones se representan mediante rectángulos y como su nombre lo dice, son los responsables de realizar una determinada acción. Estos cuadriláteros se unen mediante una línea vertical, mientras que las Transiciones representan la condición que se debe cumplir para pasar a la siguiente etapa. Si como consecuencia de esta condicionante se obtiene un resultado verdadero, se pasa a la siguiente acción, de lo contrario se espera hasta que se cumpla.

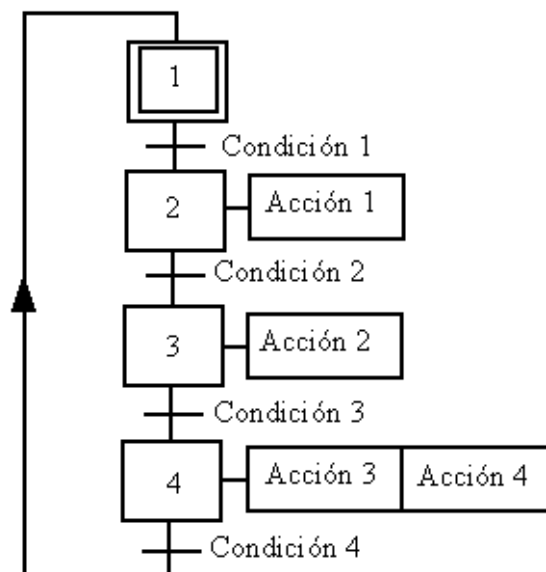


Figura 1-11. Extracto de programación SFC

B. Diagrama de Bloques de Funciones (FBD)

El lenguaje es de tipo gráfico, igual que en el diagrama anterior. Éste, permite al usuario programar rápidamente expresiones matemáticas, como lógica booleana. Es un lenguaje de alto nivel, que está orientado a una programación rápida y fácil, donde el programador solo se debe preocupar por la funcionalidad de la rutina. Es ideal para usuarios que no tengan habilidades avanzadas en programación y para procesos de baja complejidad.

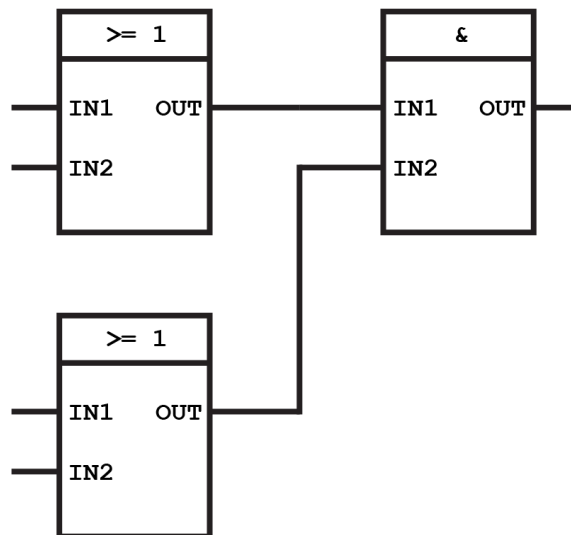


Figura 1-12. Extracto de programación por bloques

C. Diagramas del tipo Escalera (LAD)

Es el tercer lenguaje del tipo gráfico, soportado por casi todos los PLCs. Su estructura es de una conexión gráfica entre variables de tipo booleano, permitiendo establecer una analogía con controles mediante relés, donde se representa el flujo de energía en diagramas de circuitos eléctricos. Actualmente, es uno de los lenguajes más utilizados en la industria, debido a su simplicidad y soporte en los distintos PLC.

Dentro de sus características principales se encuentra el uso de barras laterales de alimentación, ubicándose en su interior elementos que componen la lógica de operación, permitiendo la inserción de elementos como contactos, bobinas y bloques de funciones especiales. Su secuencia de operación está establecida en el siguiente orden; de arriba abajo, de izquierda a derecha.

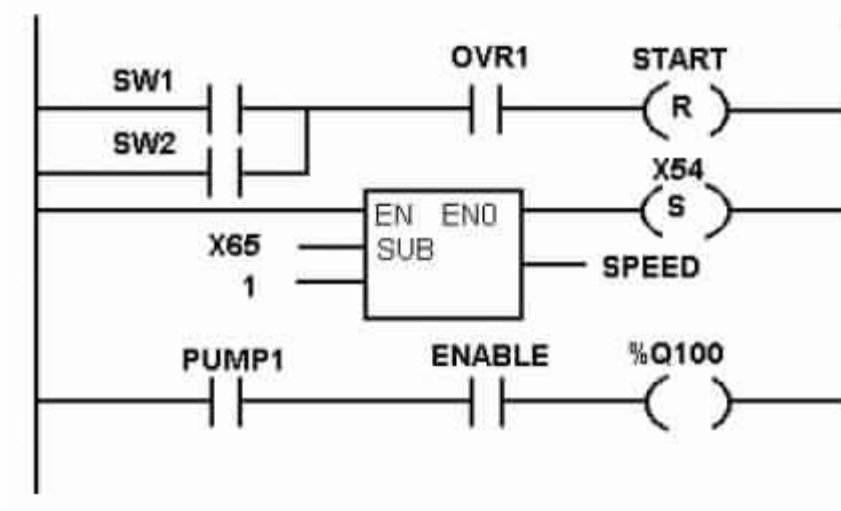


Figura 1-13. Extracto de programación ladder

D. Texto Estructurado (ST)

Este diagrama de programación está basado en lenguajes de alto nivel y es muy similar a Pascal, Basic y C. Debido a que requiere conocimiento previo de programación, no destaca por su popularidad, considerándose como un lenguaje relativamente nuevo.

Basado en lista de instrucciones, las principales ventajas de este lenguaje, es la formulación de las tareas del programa, permitiendo apreciar claramente la estructura de cada operación realizada. Se considera un lenguaje apropiado, cuando se desea programar mediante ciclos (ejemplo: if, while, for, case).

IF...	CASE... OF...	FOR... TO DO..	WHILE.. DO...	(*...*)	REGION
1	IF	"DATOS_CARROS".NUMERO_INSPECCION>9	THEN		
2		"DATOS_CARROS".NUMERO_INSPECCION := 0;			
3	END_IF;				
4					
5	IF	"DATOS_CARROS".HABILITAR_ETIQUETA AND "ST3_PIEZA_BUENA"	THEN		
6		"DATOS_CARROS".CARRO["DATOS_CARROS".NUMERO_INSPECCION] := 1;			
7	END_IF;				
8					
9	IF	"DATOS_CARROS".HABILITAR_ETIQUETA AND "ST3_PIEZA_MALA"	THEN		
10		"DATOS_CARROS".CARRO["DATOS_CARROS".NUMERO_INSPECCION] := 0;			
11	END_IF;				
12					

Figura 1-14. Extracto de programación texto estructurado

E. Lista de Instrucciones (IL o STL)

Es también un lenguaje del tipo texto, similar al ensamblador. Este es recomendado para programas de poca extensión.

Una de las principales características es que todos los operadores trabajan con un registro especial, denominado acumulador (LD, ST). La estructura de este tipo de lenguaje se puede observar a continuación:

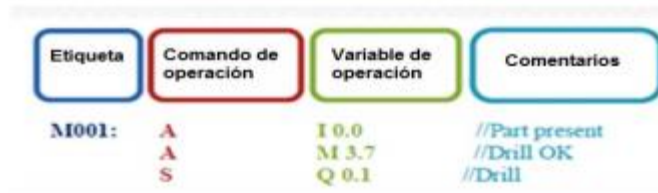


Figura 1-15. Programación lista de instrucciones

1.4 **BUSES DE CAMPO**

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción (Salazar Serna, y otros, 2011).

1.4.1 Conceptos Generales

Un bus de campo es un sistema de transmisión de información (datos) que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.

1.4.2 Tipos de Buses

Con el avance de la tecnología, se han perfeccionado y desarrollado numerosos protocolos de comunicación y buses de campo, en el listado se destacan los más utilizados.

- Profibus PA
- CANOpen
- DeviceNet
- Hart
- AS-interface

- Entre otros.

1.4.3 Bus de Campo AS-interface

El bus de campo AS-interface se encuentra ubicado en la parte inferior de la pirámide de CIM (Figura 1-16), ya que interactúa directamente en el proceso. Por otro lado, se puede clasificar en las capas 1, 2 y 7 del modelo OSI, es decir en la capa física, enlace y aplicación (MICRO, 2018).

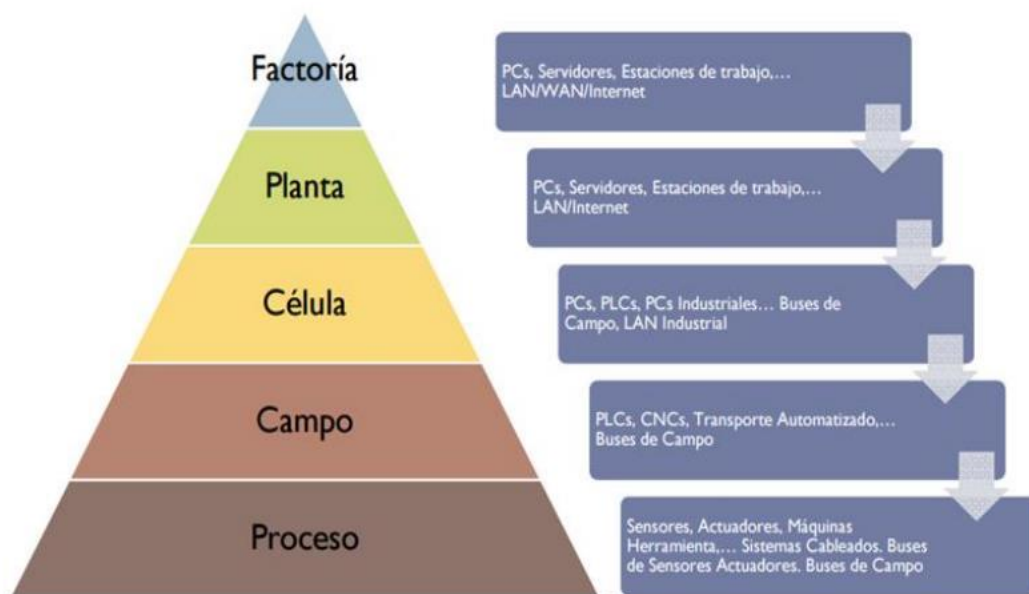


Figura 1-16. Pirámide de CIM

Este sistema de comunicaciones permite reemplazar la gran cantidad de cables que conectan los sensores y actuadores generalmente ON/OFF (binarios) por solamente un cable perfilado (Cable AS-i) y sus respectivos nodos esclavos de conexión.

La Asociación AS-interface fue formada en Alemania por 11 empresas del área de la automatización, entre ellas: Festo KG, Siemens AG, Leuze electrónic GmbH&Co. Posteriormente se han unido otros fabricantes como Eaton Corporation, Endress & Hauser y Honeywell.

Esta asociación es la encargada de crear las especificaciones eléctricas y mecánicas para trabajar con el Bus AS-i. Estas especificaciones son entregadas gratuitamente a cualquier fabricante interesado, con el fin de estandarizar la red, desarrollar mayor tecnología y certificar los productos (por eso es reconocido como un Bus estándar).

La certificación permite asegurar que el producto cumple con los estándares para trabajar con AS-i y se reconoce por el sello "AS-interface" y el número de prueba correspondiente.

Las principales características del Bus AS-i son:

- A. A través de un mismo cable se conectan múltiples actuadores y sensores, originalmente ON/OFF, además de su alimentación eléctrica.
- B. No hay necesidad de pelar ni unir cables, ya que la conexión se realiza mediante perforación de la aislación.
- C. Determinista y de acción rápida, el maestro AS-i toma solo 5 ms para intercambiar datos con hasta 31 esclavos AS-i.
- D. A cada estación esclavo AS-i se le pueden conectar hasta 8 dispositivos.
- E. Cada maestro AS-i estándar soporta hasta 124 actuadores y 124 sensores.
- F. Los maestros AS-i con direccionamiento ampliado (maestro extendido) pueden operar hasta 186 actuadores y 248 sensores binarios.

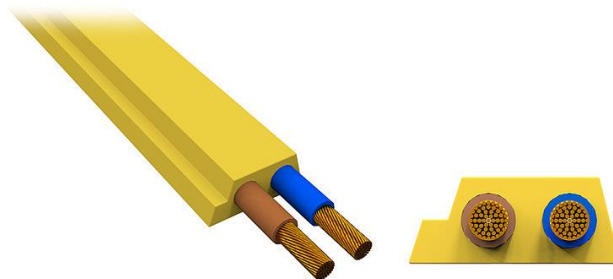


Figura 1-17. Cable AS-i

Detalles técnicos AS-interface (Tabla 1-1)

Tabla 1-1. Detalles Bus AS-i

Método de acceso	Maestro/esclavo (Polling)
Tiempo de ciclo	5 ms para 31 esclavos / 10 ms para 62 esclavos
Medio físico de transmisión	Cable bifilar, no apantallado (2x1.5mm ²)
Longitud máxima transmisión	100 metros uniendo todos los tramos
Longitud máxima transmisión con repetidor	300 metros con repetidor / 600 metros con dos repetidores en paralelo
Cantidad máxima de esclavos Standard	62 esclavos (124 actuadores y 124 sensores)
Cantidad máxima de esclavos extendido	62 esclavos (186 actuadores y 248 sensores)
Topología de red	Árbol, estrella, Bus (línea)
Protocolo	AS-i
Nivel	Nivel de campo (proceso)

Funcionamiento: Ciclo de lectura y escritura en los esclavos

La red AS-i es un sistema maestro/esclavo y su método de acceso al medio en AS-i es el sondeo (polling), es decir, el maestro "encuesta" uno por uno a cada esclavo.

1. Primero, se realiza el sondeo en todos los esclavos del tipo A (1A hasta 31A), leyendo entradas y forzando salidas. Luego, se realiza el mismo proceso en los esclavos tipo B (1B a 31B).
2. Como el sistema es determinista, es seguro que en 5 ms se habrán actualizado los datos de los esclavos tipo A y en los siguientes 5 ms, los datos de los esclavos tipo B.
3. Si bien el maestro AS-i se encuentra enlazado con la CPU del PLC, este cuenta con su propio procesador, por lo que realiza la función de actualización de datos de entrada y salida de manera independiente, guardando la información en su memoria no volátil.
4. La CPU realizará el rescate de los estados de las entradas y salidas en el periodo de Scan, tal como si fueran entradas y salidas del mismo PLC. Para esto, se debe asignar espacios de memoria para trasladar los datos del maestro AS-i al controlador.

**CAPITULO 2: PROGRAMACIÓN DE ESTACIONES DE DISTRIBUCIÓN Y
TRANSPORTE DE PIEZAS**

2.1 ESTACIÓN DE ADQUISICIÓN

La estación de adquisición (Figura 2-1), opera como contenedor de entrada al proceso de manufactura. El brazo robótico toma la pieza terminada en la CNC y la almacena en el depósito. Cuando la estación de distribución demanda productos, el brazo neumático toma la pieza y la entrega a la siguiente estación.

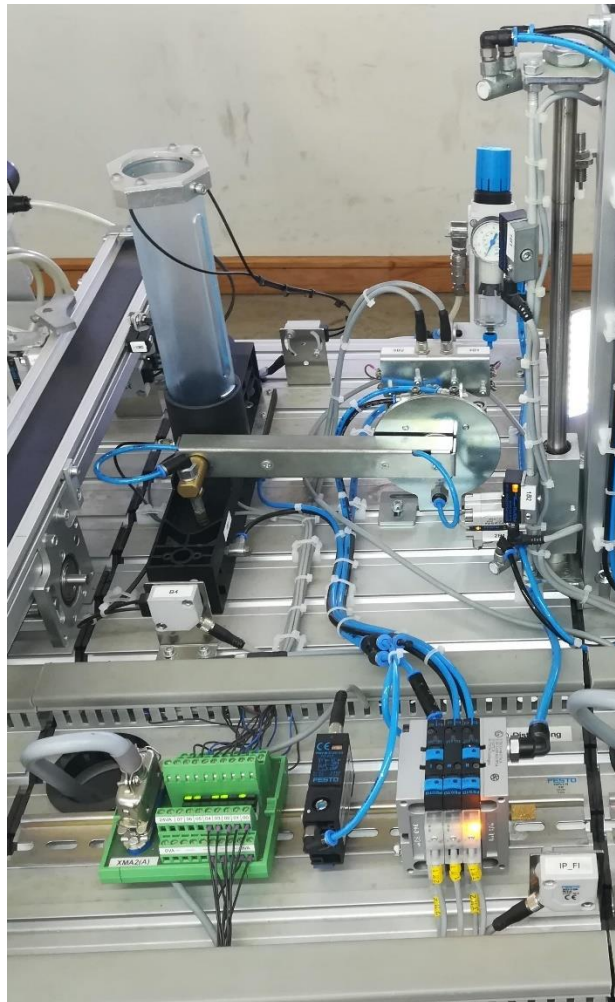


Figura 2-1. Estación de adquisición

2.1.1 Partes de la estación

A. Contenedor

El contenedor es un recipiente cuya función principal es almacenar de manera momentánea los productos entrantes a celda flexible. Cuenta con dos sensores de tipo infrarrojo, uno en la parte inferior y otro en la parte superior. El sensor inferior tiene como tarea indicar cuando hay disponibilidad de piezas en la entrada, mientras

que el sensor superior indica sobre nivel, es decir, contenedor lleno. En la siguiente imagen se aprecia el contenedor.



Figura 2-2 Contenedor de entrada

B. Cilindro Neumático

El cilindro cuenta con accionamiento neumático del tipo normal abierto y retorno por resorte. Su misión es actuar como dispensador de piezas. Para poder controlar su funcionamiento, dispone de dos sensores magnéticos para indicar los finales de carrera. Su operación comienza cuando se activa la solicitud de producto, en ese momento el vástago se extiende hasta el final de carrera, colocando la pieza en posición para ser trasladada por el brazo neumático.

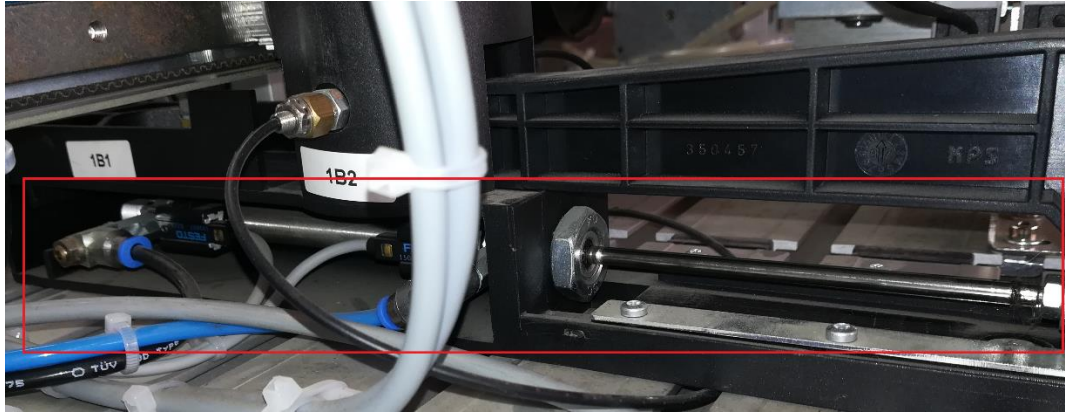


Figura 2-3 Cilindro neumático

C. Brazo Neumático

El brazo debe trasladar la pieza ya posicionada, hacia la estación de distribución. Para conocer la posición de dicho actuador, se utilizan dos interruptores finales de carrera accionados por los topes mecánicos (limit switch). Por otro lado, el brazo cuenta con una boquilla neumática, que se encarga de producir vacío y de esta forma tomar el producto, además, puede generar presión para desprender la pieza cuando sea necesario, estas funciones son controladas mediante el bloque de electroválvulas.



Figura 2-4 Brazo neumático

2.1.2 Lógica de operación

La lógica de operación programada en la estación de adquisición está dada por la siguiente estructura:

- A. Para comenzar el ciclo de trabajo, es necesario que se cumplan dos criterios. El primero es que la estación de distribución solicite una pieza, mientras que el segundo, es que haya alguna pieza en el contenedor de entrada, si se cumplen ambos criterios se da paso al ciclo de transporte.
- B. El ciclo se repetirá indefinidamente hasta que la señal de solicitud sea cancelada. En este momento el sistema quedará en espera de una nueva petición.

En el próximo diagrama de flujo se ilustra el flujo básico de la estación de adquisición.

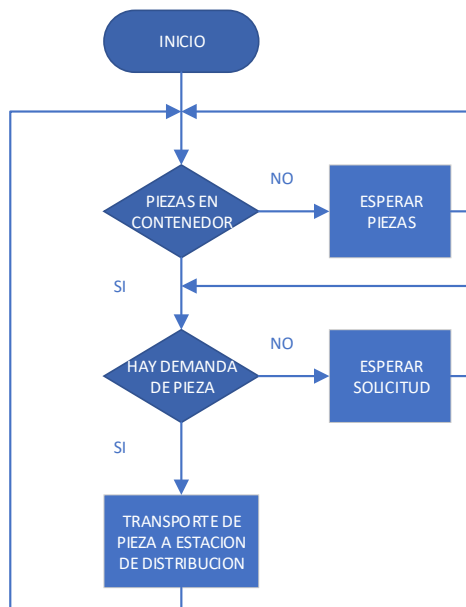


Figura 2-5 Diagrama de flujo estación de adquisición

2.1.3 Levantamiento

Para poder comenzar con la programación de la estación, es necesario determinar los sensores y actuadores involucrados en el proceso, en la siguiente tabla se detalla cada elemento, con su respectivo direccionamiento.

Tabla 2-1 Sensores y actuadores estación de adquisición

ESTACION 1/ETIQUETA	Entrada / Salida	TIPO DE SENSOR / Actuador	COMENTARIO	DIRECCION PLC
1B1	Entrada	Inductivo	Pistón tubo de entrada Abierto	I136.1
1B2	Entrada	Inductivo	Pistón tubo de entrada Cerrado	I136.2
3B1	Entrada	Switch	Adquisición de pieza	I136.4
3B2	Entrada	Switch	Pieza en destino	I136.5
10B7	Entrada	Laser	Tubo entrada Inferior	I136.6
B4	Entrada	Laser	Tubo entrada Superior	No Conectado
1M1	Salida	Electroválvula	Cierra Pistón (Abre automáticamente)	Q136.0
2M1	Salida	Electroválvula	Vacio en Boquilla	Q136.1
2M2	Salida	Electroválvula	Presion en Boquilla	Q136.2
3M1	Salida	Electroválvula	Mover Brazo a Entrada	Q136.3
3M2	Salida	Electroválvula	Mover Brazo a Salida	Q136.4

2.1.4 Programación en TIA Portal

Teniendo como referencia el direccionamiento de los periféricos, la secuencia de operación de la planta y la definición de los lenguajes de operación, se selecciona el lenguaje de programación. En este caso se considera como más adecuado el SFC, ya que se trata de un proceso secuencial y repetitivo.

Para realizar la programación se utilizó el siguiente procedimiento:

En primer lugar, es necesario configurar el hardware, para esto hay que ir al árbol de proyecto y realizar estos pasos:

- A. Agregar un nuevo dispositivo.
- B. Seleccionar la CPU que se va a utilizar que corresponde a la 314C-2 PN/DP.
- C. Agregar los módulos adicionales, que para esta estación son una fuente de poder modelo PS-307-5A y un módulo de simulación que será configurado como 16 entradas digitales.
- D. Verificar el direccionamiento que TIA Portal asigna a cada módulo, en este caso, para las entradas y salidas se utilizan los bytes 136 y 137, mientras que el módulo de simulación utiliza los bytes 0 y 1.
- E. Direccionar el puerto Ethernet, en este caso corresponde a la IP 10.4.5.82 con máscara 255.255.255.0.

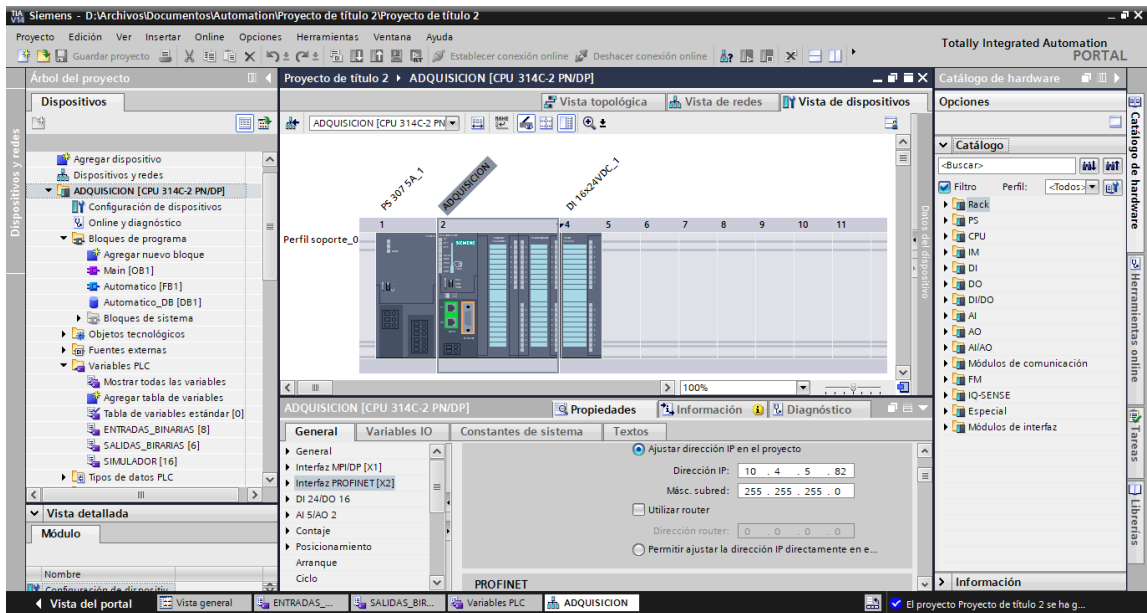


Figura 2-6 Imagen de configuración de hardware estación adquisición

Luego se debe realizar una tabla de variable, donde se asignará un TAG y se definirá el tipo de dato (bit, byte, Word, Int, etc.) a cada dirección de memoria y/o salidas y entradas que se van a utilizar.

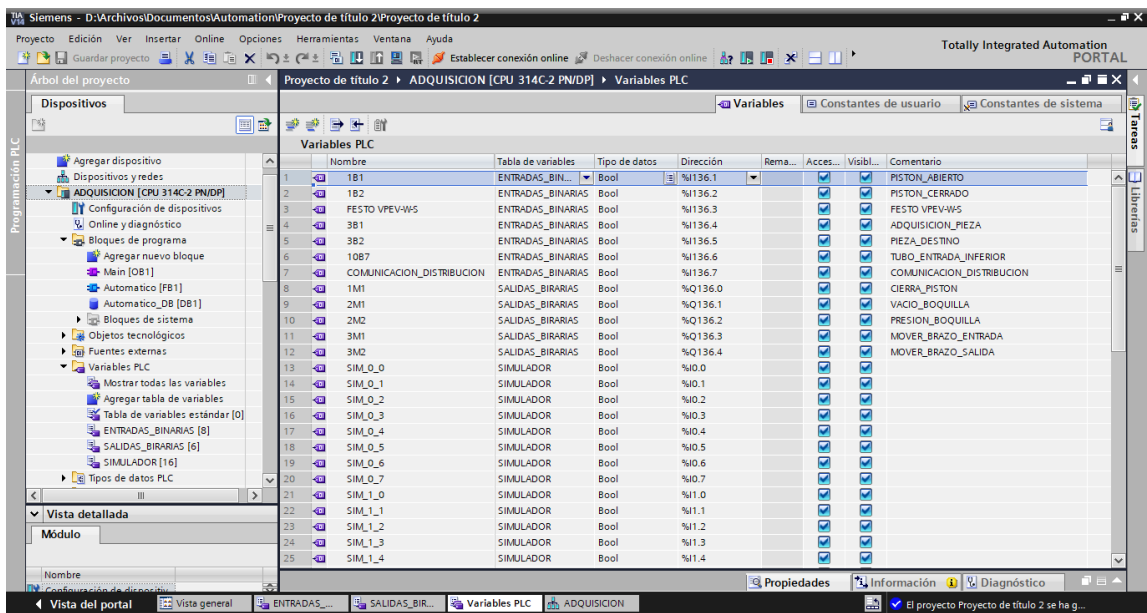


Figura 2-7 Variables estación de adquisición

Posteriormente, se agrega un nuevo "Bloque de Función" ya que estos son los únicos que admiten programación en lenguaje GRAPH.

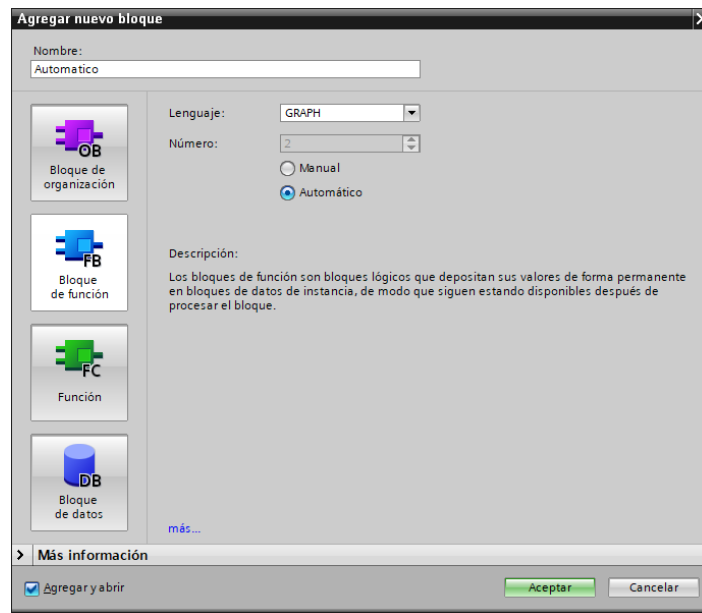


Figura 2-8 Bloque de función

Teniendo todas las configuraciones previas se puede comenzar con la programación, en la siguiente figura se puede observar un extracto de la programación GRAPH correspondiente al traslado de una pieza hacia la estación de distribución y rechazo de ésta si no cumple con la altura adecuada.

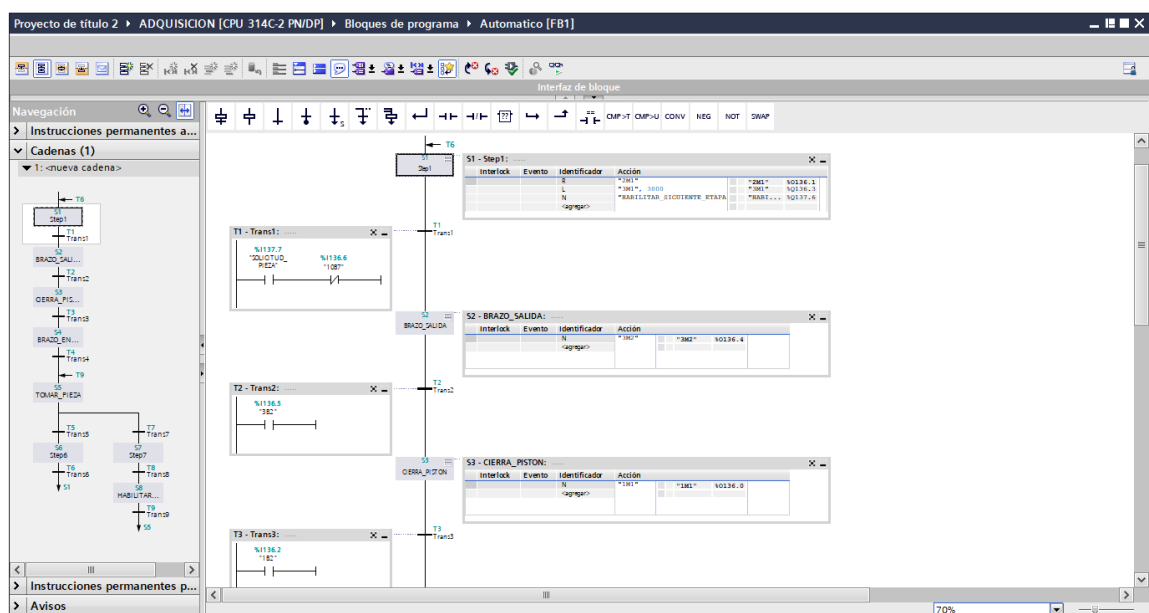


Figura 2-9 Programación estación de adquisición

2.2 ESTACIÓN DE DISTRIBUCIÓN

La estación de distribución tiene como función trasladar piezas hacia la estación de transporte, además cuenta con sensores para determinar algunas características de los materiales y altura de las piezas. Por otro lado, cuenta con un elevador que permite conectar con la siguiente estación mediante una cinta transportadora por aire, o bien rechazar el producto y almacenarlo en una línea de espera.

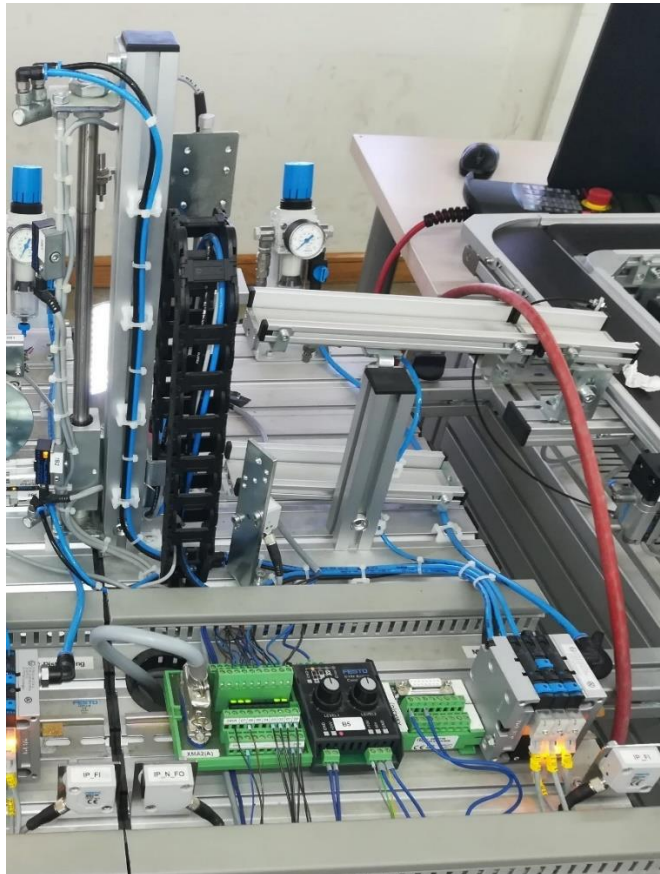


Figura 2-10. Estación de distribución

2.2.1 Partes de la estación

A. Plataforma de entrada

Es el punto de entrada del producto (Figura 2-11), cuenta con dos sensores para determinar la presencia de producto mediante el sensor láser (1) y su material a través de un sensor Capacitivo (2).



Figura 2-11. Plataforma de entrada

B. Cilindro Neumático

El cilindro neumático de simple efecto tiene dos funciones; una es rechazar una pieza que no cumple con las características adecuadas, y por otro lado debe mover el producto hacia la cinta de transporte neumática cuando éste cumple con todos los criterios.

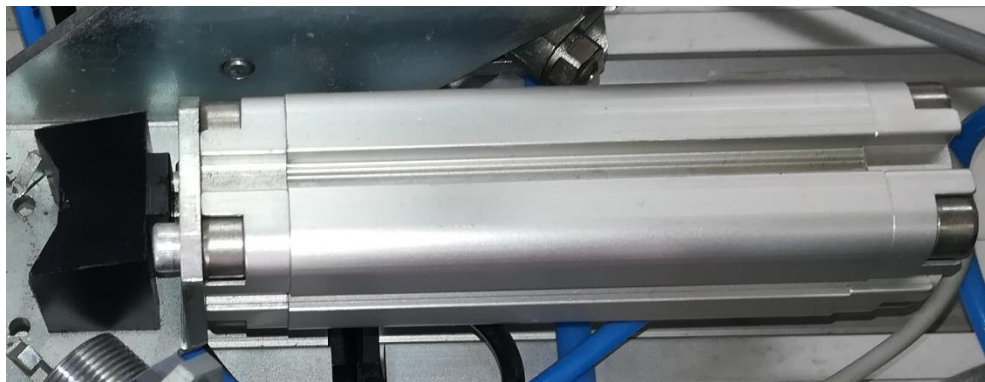


Figura 2-12. Cilindro neumático

C. Línea de Rechazo

Cuando una pieza no aprueba los requerimientos es almacenada en una línea ajena al proceso de manufactura.



Figura 2-13. Cinta de rechazo

D. Elevador

Se compone de un cilindro elevador de doble efecto que se encarga de llevar hasta el segundo nivel los productos para analizar su altura y posteriormente despacharlos si cumple con el requerimiento, o bien, llevarlos nuevamente al primer nivel para rechazarlos si no aprueban.



Figura 2-14. Elevador

E. Sensor de Altura

Es un sensor del tipo resistivo (Figura 2-14) que se conecta a un comparador (Figura 2-15). Este transductor se compone de una entrada análoga de 0-10V, tres salidas digitales que indican si el producto tiene una altura bajo el mínimo, la segunda indica si se encuentra dentro del rango aceptable (que se conecta directamente al PLC), o bien sobre el máximo permitido. Para ajustar la altura mínima y máxima del producto se utilizan dos potenciómetros que se encuentran en la parte superior.

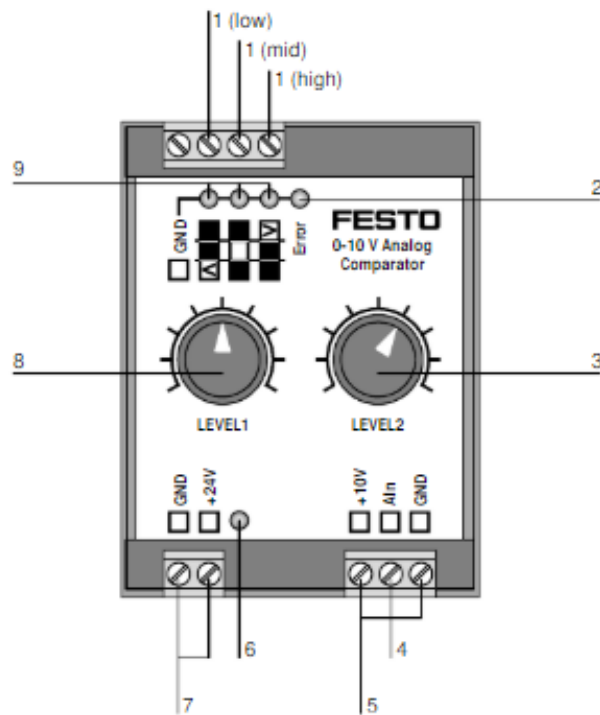


Figura 2-15. Comparador analógico

A continuación, se detallan las partes que componen el comparador analógico.

1. Salidas Digitales
2. Led para indicación de Error
3. Potenciómetro para ajustar el valor máximo
4. Entrada Analógica
5. Salida de alimentación para sensor
6. Indicador de sobrevoltaje
7. Entrada de alimentación 24VDC
8. Potenciómetro para ajustar el valor mínimo
9. Leds para indicación de salidas digitales

F. Cinta transportadora por aire

Se compone de una cinta tipo estacionaria con pequeñas perforaciones en su interior, las que se encuentran conectadas a una electroválvula neumática. La plataforma además posee un desnivel y producto de esto, al activarse la válvula se produce una pequeña ventosa que desplaza la pieza hasta un carro de transporte.

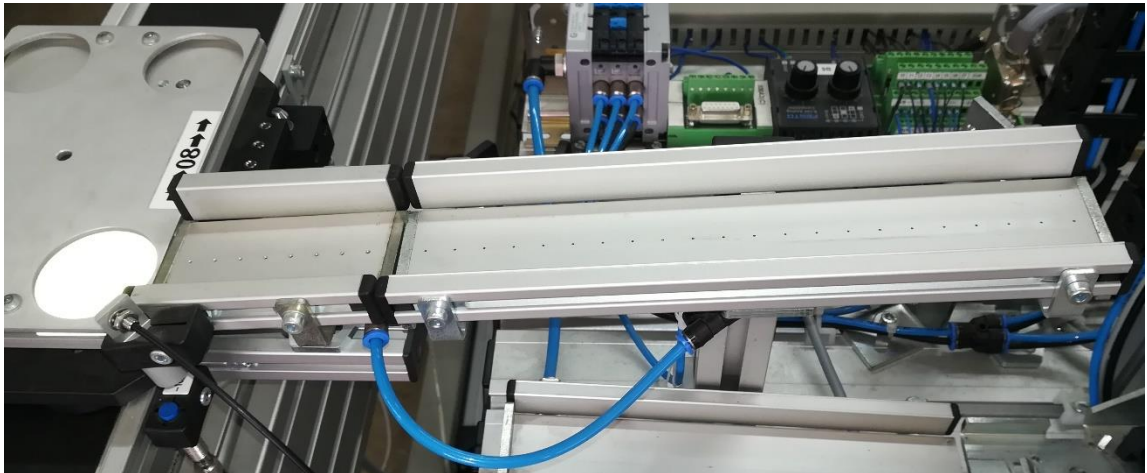


Figura 2-16. Cinta transportadora por aire

2.2.2 Lógica de Operación

La lógica de operación programada en la estación de distribución está dada por la siguiente estructura (Figura 2-17):

- A. Para comenzar el ciclo de trabajo, es necesario que se cumplan dos criterios. El primero es que la estación de adquisición se encuentre en la posición de espera, mientras que el segundo, es que haya alguna pieza en el contenedor de entrada, si se cumplen ambos criterios se da paso al ciclo de transporte.
- B. Finalizado el proceso, si la estación de transporte solicita producto, se solicitará a la estación de adquisición y de esta forma el ciclo se repetirá indefinidamente hasta que la señal de solicitud sea cancelada. En este momento el sistema quedará en espera de una nueva petición.

En el próximo diagrama de flujo se ilustra el flujo básico de la estación de distribución.

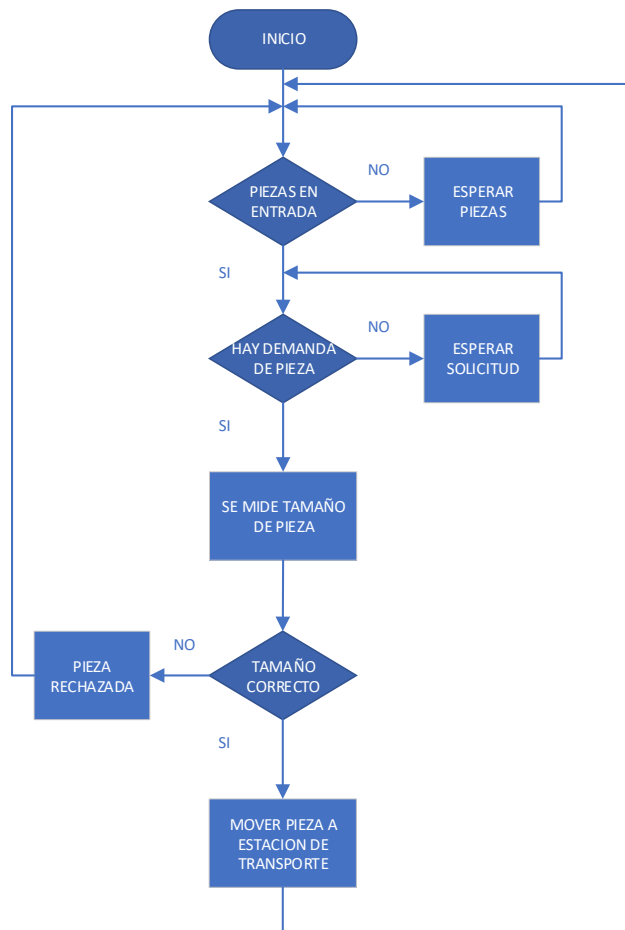


Figura 2-17. Diagrama de flujo estación de distribución

2.2.3 Levantamiento

Para poder comenzar con la programación de la estación, es necesario determinar los sensores y actuadores involucrados en el proceso, en la siguiente tabla se detalla cada elemento, con su respectivo direccionamiento.

Tabla 2-2. Sensores y actuadores estación de distribución

Estación de Distribución				
ESTACION 1/ETIQUETA	Entrada / Salida	TIPO DE SENSOR / Actuador	COMENTARIO	DIRECCION PLC
PART_AV	Entrada		Sensor de Pieza en Entrada de Estación	I136.0
SE	Entrada	Laser	Sensor de Pieza entrada y Salida	I136.1
B4	Entrada	Óptico	Sensor de Brazo	I136.2
R1	Entrada	Resistivo	Sensor de Altura de pieza	I136.3
1B1	Entrada	Inductivo	Sensor Superior de Elevador	I136.4
1B2	Entrada	Inductivo	Sensor Inferior de Elevador	I136.5
2B1	Entrada	Inductivo	Sensor Posición de pistón de despacho	I136.6
IP_FI	Entrada	Óptico	Comunicación Siguiete Estación	I136.7
1M2	Salida	Electroválvula	Bajar Elevador	Q136.0
1M1	Salida	Electroválvula	Subir Elevador	Q136.1
2M1	Salida	Electroválvula	Pistón Despachar pieza	Q136.2
3M1	Salida	Electroválvula	Activa Cinta de Aire	Q136.3
IP_N_FO	Salida	Óptico	Comunicación Estación Anterior	Q136.6

2.2.4 Programación en TIA Portal

Considerando como referencia los datos de la tabla 2-2 para el direccionamiento de los dispositivos periféricos, la secuencia de operación definida en la figura 2-17 y la selección del lenguaje adecuado para la programación del proceso, que en este caso es SFC, pues se trata de una lógica secuencial y cíclica.

El procedimiento para la programación de esta estación es el siguiente:

- A. Para comenzar es necesario configurar el hardware existente, esto se debe hacer desde el árbol de proyecto y seguir el procedimiento que se detalla a continuación:
 - a. Agregar un nuevo dispositivo
 - b. Seleccionar la CPU que se va a utilizar que corresponde a la 314C-2 PN/DP.
 - c. Agregar los módulos adicionales, que para esta estación son una fuente de poder modelo PS-307-5A y un módulo de simulación que será configurado como 16 entradas digitales.
 - d. Verificar el direccionamiento que TIA Portal asigna a cada módulo, en este caso, para las entradas y salidas se utilizan los bytes 136 y 137, mientras que el módulo de simulación utiliza los bytes 0 y 1.
 - e. Direccionar el puerto Ethernet, en este caso corresponde a la IP 10.4.5.83 con máscara 255.255.255.0.

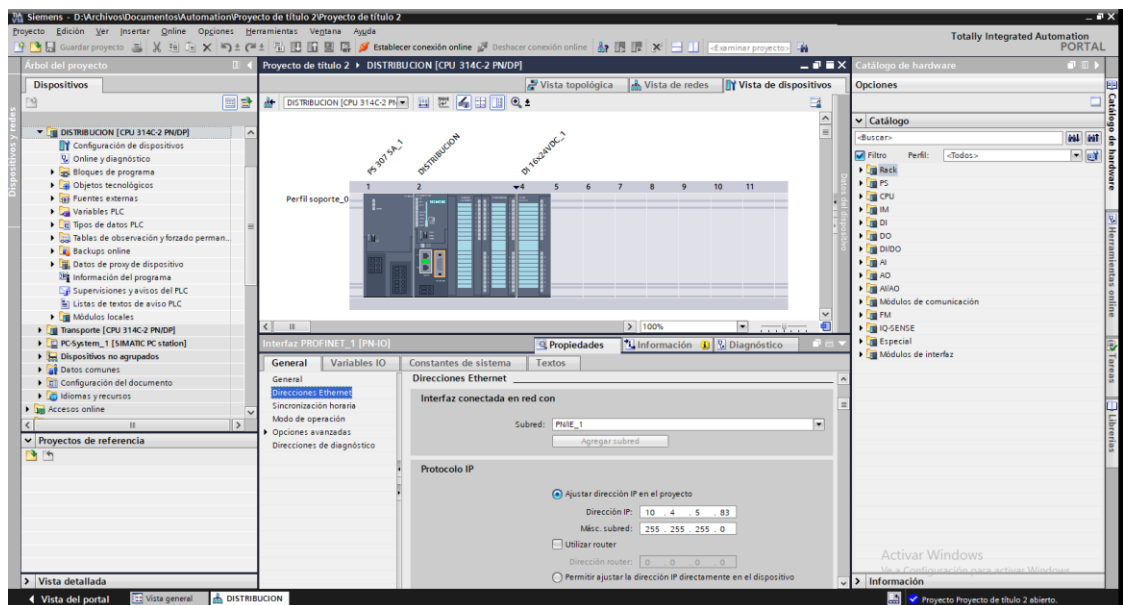


Figura 2-18. Imagen de configuración de hardware estación distribución

- B. Luego se debe realizar una tabla de variable, donde se asignará un TAG a cada variable y se definirá el tipo de dato (bit, byte, Word, Int, etc.) a cada dirección de memoria y/o salidas y entradas que se van a utilizar.

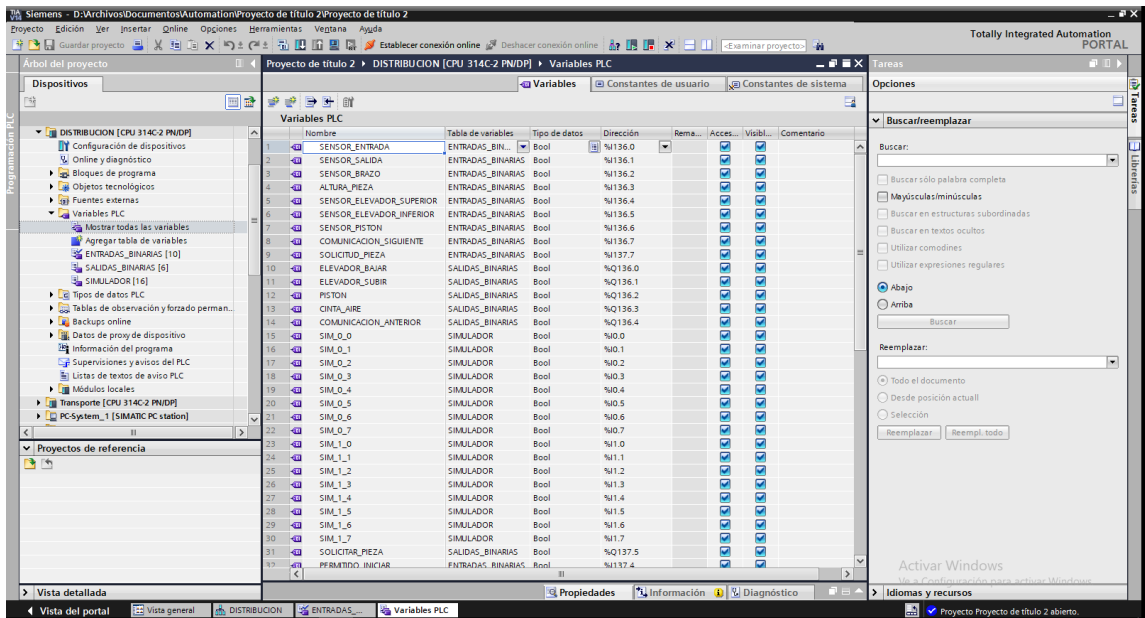


Figura 2-19. Variables estación de distribución

C. Posteriormente se debe agregar un nuevo "Bloque de Función FB" (Figura 2-20), ya que éstos son los únicos que admiten programación en lenguaje S7 GRAPH.

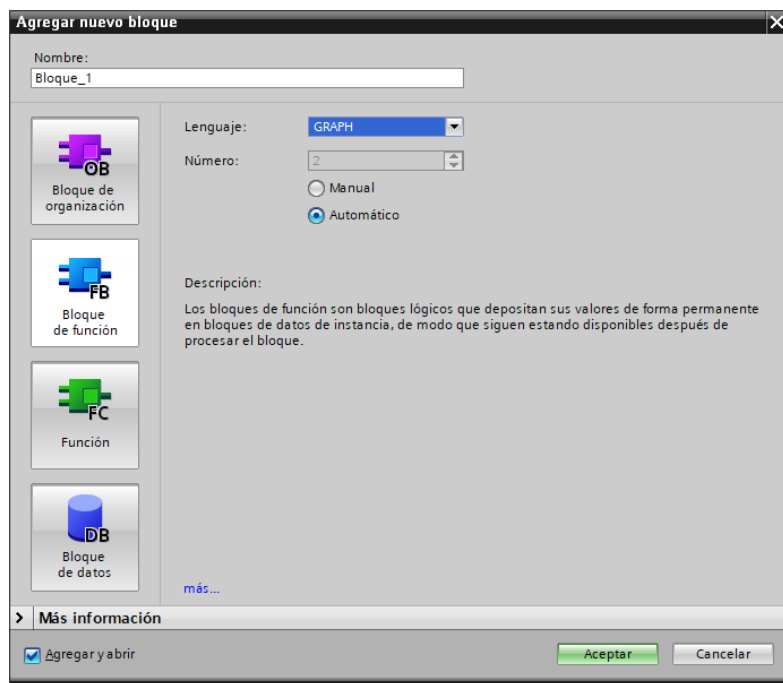


Figura 2-20. Agregar bloque de función FB

D. Finalmente, teniendo todas las configuraciones previas se puede comenzar con la programación. En la figura 2-21 se observa un extracto del código en S7 GRAPH que corresponde al movimiento de una pieza hacia la estación de transporte.

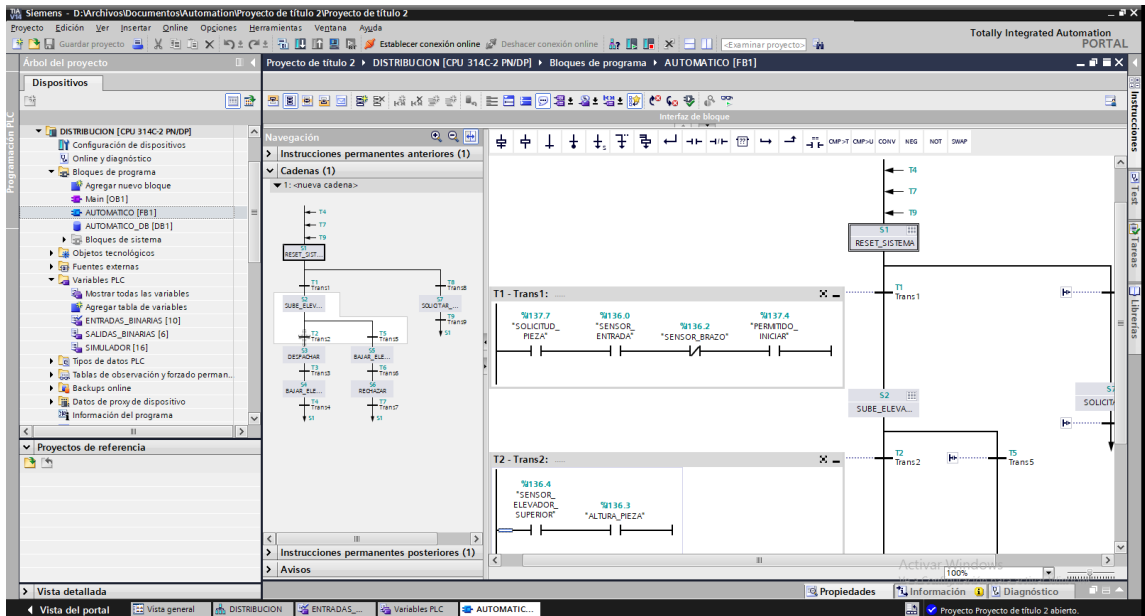


Figura 2-21. Programación estación de distribución

2.3 ESTACIÓN DE TRANSPORTE

La estación de transporte de la FMS MPS-500 tiene como objetivo llevar la materia prima o producto por las distintas estaciones de trabajo. Originalmente está diseñada para operar con seis estaciones, en el caso de la UTFSM Sede Concepción se encuentran operativas tres de ellas. En la siguiente tabla se detalla el estado de cada una:

Tabla 2-3. Ocupación estación de transporte

ESTACIÓN	DETALLE
Estación 1	Estación de distribución
Estación 2	Disponible
Estación 3	Estación de inspección
Estación 4	Disponible
Estación 5	Estación de almacenamiento (AS-RS)
Estación 6	Disponible

Como dispositivo de control se utiliza un PLC de la marca SIEMENS de la serie S7-300, modelo CPU 314C-2 PN-DP. A su vez esta CPU cuenta con dos módulos de ampliación, correspondientes a un simulador modelo 274 2XH01-0AA0 y un módulo AS-interface cuya referencia del fabricante es 343-2AH01-0XA0.

La estación de transporte utiliza un bus de campo AS-i para interconectar sus sensores y actuadores. Siendo en total diez y ocho nodos esclavos los utilizados en las seis estaciones de trabajo.



Figura 2-22. Estación de transporte

2.3.1 Partes de la estación

El punto central de la estación se encuentra alojada en un armario de control, en la siguiente imagen se numeran los distintos componentes para su posterior explicación.

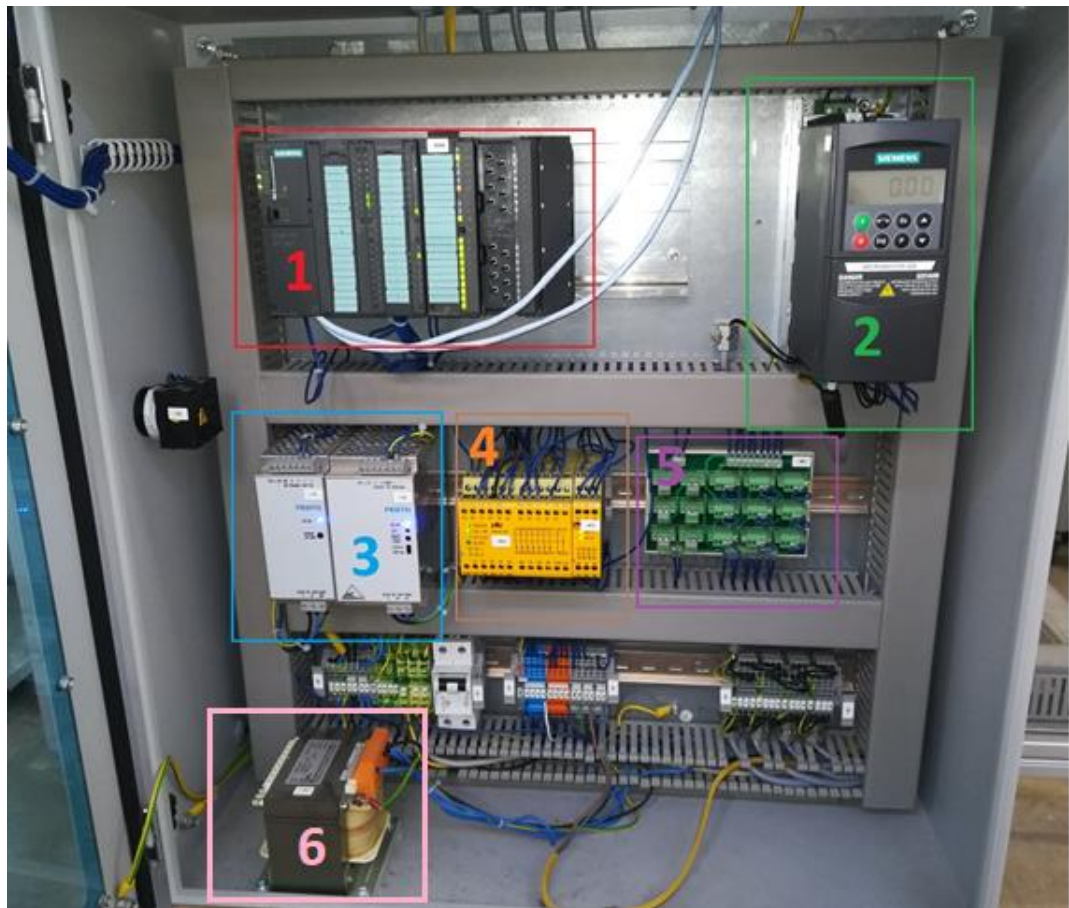


Figura 2-23. Tablero de control

1. PLC SIEMENS S7-300 CPU 314C- PN/DP

Tiene como función controlar los todos los dispositivos de campo asociados a esta estación. Además, mediante el sistema de comunicación SYS LINK y los distintos nodos de entradas y salidas AS-i coordina los procesos de las distintas estaciones de trabajo.

2. Variador de frecuencia SIEMENS MICROMASTER 420

Su tarea es controlar la velocidad de los cuatro motores que contienen las cintas transportadoras de la estación de transporte. La velocidad de movimiento configurada es fija, pero se puede modificar mediante su menú de ajustes. Este elemento es accionado directamente desde el PLC mediante una salida digital, accionando los cuatro motores simultáneamente.

3. Fuentes de poder FESTO

Debido a que la alimentación del sistema es de 220VAC se utilizan dos fuentes conmutadas para alimentar los distintos elementos.

4. Relé de seguridad PILZ PNOZ X9

Los relés de seguridad tienen como fin supervisar de forma segura las señales procedentes de dispositivos de seguridad en todo momento y desconectarse de forma rápida en caso de emergencia. En este caso se utiliza para la lógica de habilitación y parada de emergencia de la estación.

5. Tarjeta de conexiones paradas de emergencia

Esta tarjeta de seguridad se utiliza para conectar los dispositivos que accionan la parada de emergencia.

6. Transformador 380/220 VAC

La alimentación principal del tablero es de 380 VAC, por ello es necesario utilizar un transformador de control para energizar las fuentes de poder con 220 VAC.

Por otro lado, cada estación de trabajo se compone de los siguientes elementos:

A. Nodo esclavo AS-i marca SIEMENS, modelo 3RG9002

Es un dispositivo que contiene cuatro entradas y cuatro salidas digitales, en esta aplicación se utiliza para comunicar con la estación anterior o siguiente mediante la interfaz SYS LINK.

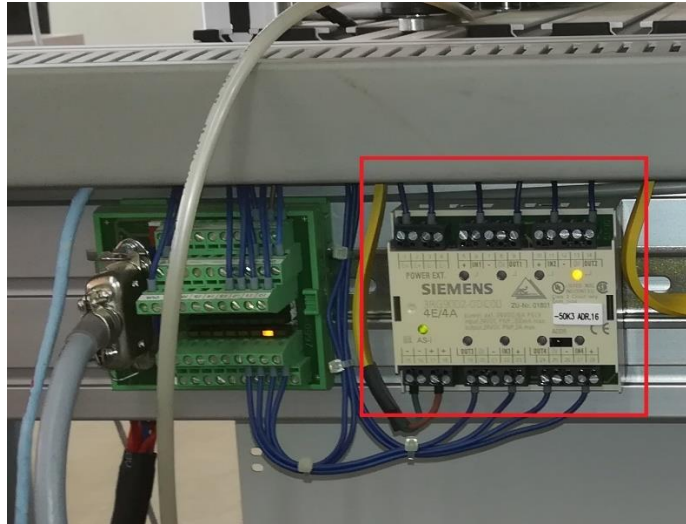


Figura 2-24. Esclavo AS-i SYS link

B. Nodo esclavo AS-i marca SIEMENS, modelo 3RK1200

Es un módulo de cuatro entradas digitales, los sensores se conectan directamente mediante un conector M12. A este aparato se conectan tres sensores de proximidad del tipo inductivo y un sensor de barrera láser.

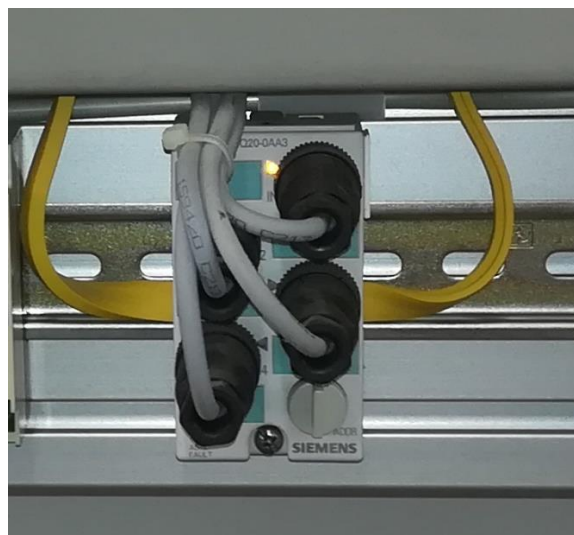


Figura 2-25. Esclavo AS-i Sensores

C. Nodo esclavo AS-i marca FESTO, modelo ASI-EVA-K1-2E1A-Z

Este nodo cuenta con una salida digital que se utiliza para energizar una electroválvula neumática.



Figura 2-26. Esclavo AS-i para pistón neumático

D. Electroválvula cinco vías, dos posiciones marca FESTO, modelo CPE10-M1BH-5L-M7

Es una electroválvula neumática de cinco vías, dos posiciones con accionamiento por solenoide de 24VDC. Tiene como función la activación de un cilindro de simple efecto.

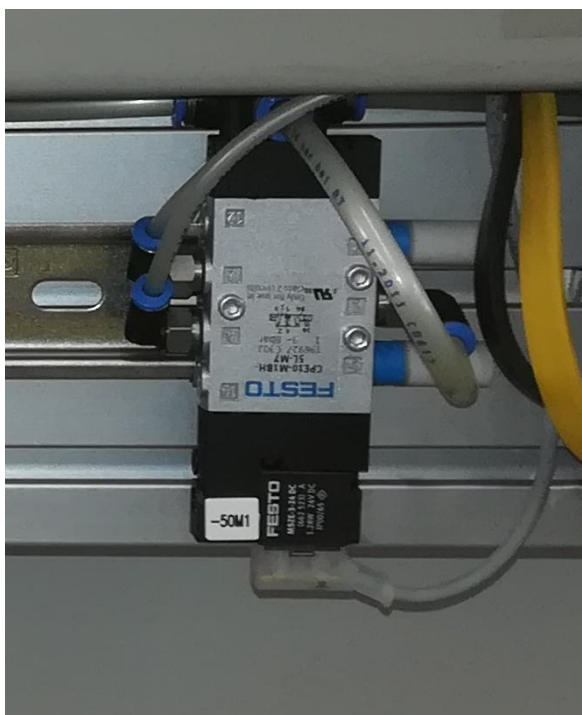


Figura 2-27. Electroválvula neumática

E. Cilindro neumático de simple efecto

El cilindro de simple efecto con accionamiento neumático permite detener un carro al llegar a la estación, o bien, autorizar el paso cuando sea necesario.

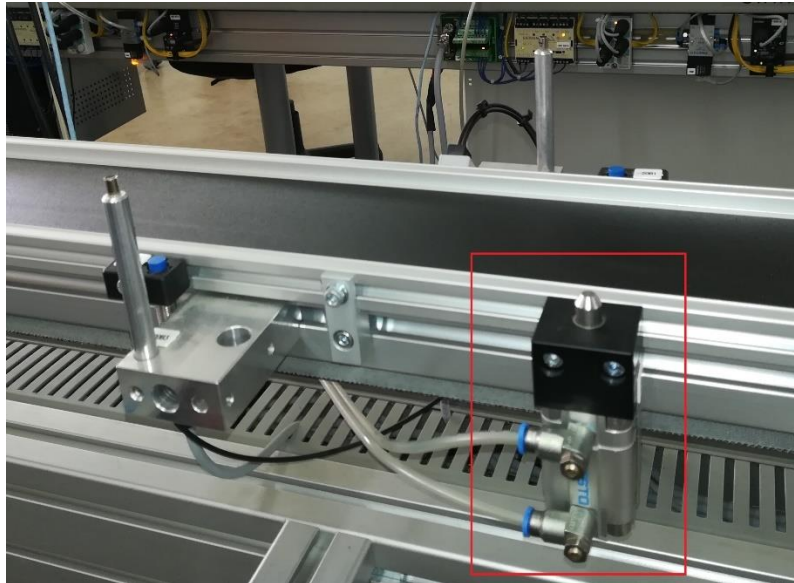


Figura 2-28. Cilindro neumático

F. Sensor de proximidad tipo inductivo

Cada estación cuenta con tres sensores de proximidad inductivos, éstos son señalados en la Figura 2-25 y posteriormente se explica su función.

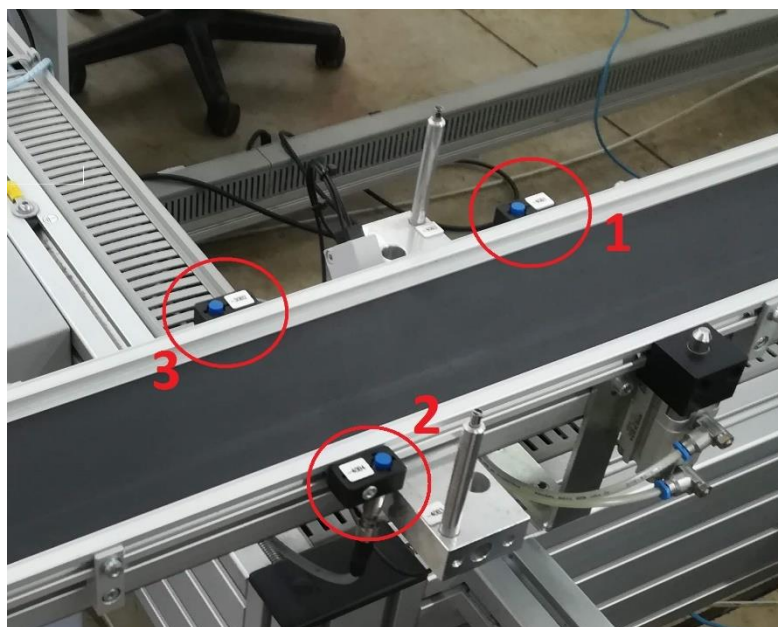


Figura 2-29. Sensores de proximidad

1. Para tope de carro

Tiene como función indicar cuando un carro se encuentra correctamente posicionado en la estación correspondiente, por otro lado, permite saber cuándo un carro ya no se encuentra en la estación. Esto es posible debido a que, en la parte inferior izquierda, cada vehículo de transporte posee una franja metálica en toda su extensión.

2. Número de Carro

Cada carro tiene una numeración, físicamente se utilizan pequeñas marcas metálicas alojadas en la banda derecha de los vehículos. El sensor inductivo detecta estas marcas y almacena los valores en un contador, de esta forma se puede saber que vehículo se encuentra en la estación.

3. Estación siguiente saturada

Tiene como función indicar a la estación anterior que no se pueden recibir más carros, ya que esta etapa se encuentra saturada. Utiliza la franja izquierda del vehículo para detectar presencia.

G. Sensor tipo barrera láser

El principio de funcionamiento de este tipo de sensor es de un haz emitido por el transmisor láser, que se encuentra direccionado hacia el receptor. Cuando un elemento interrumpe la línea visual de ambos elementos permite la conmutación su transistor de salida. En la FMS se utiliza para detectar si el carro contiene o no una pieza.

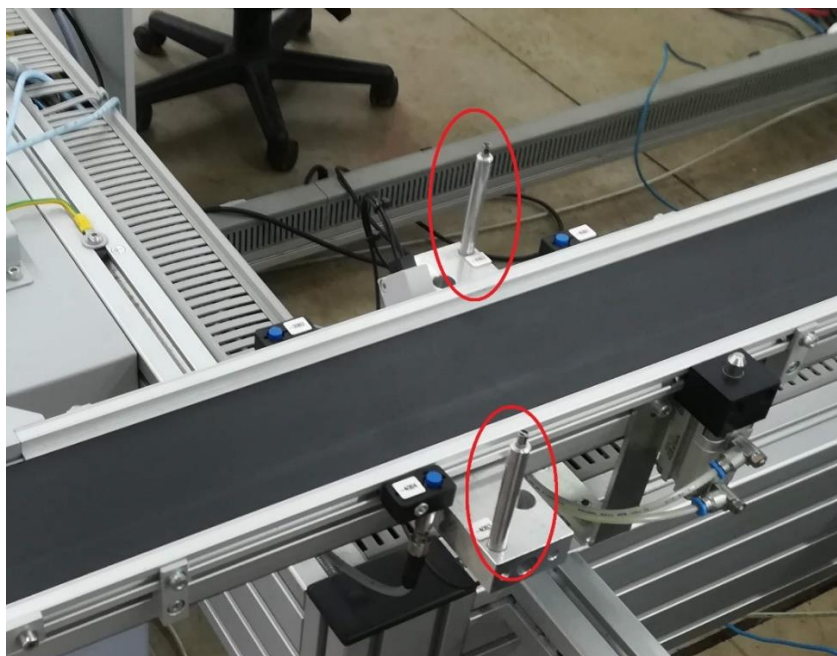


Figura 2-30. Sensor láser

2.3.2 Lógica de Operación

La lógica de operación de esta etapa está determinada por las figuras 2-31 y 2-32, dónde se puede apreciar una estructura cíclica y paralela de cada estación de trabajo (solo en el arranque se trabaja de forma secuencial). Para comenzar la operación de cada estación es necesario que se cumplan tres criterios:

- A. El primero es que el carro contenga un producto (no aplica para estaciones ST2, ST4 y ST6, ya que no existe proceso).
- B. El segundo es que la siguiente estación no se encuentre saturada con vehículos en espera.
- C. Finalmente, el tercer criterio es que se detecte un móvil correctamente posicionado en la estación. Si se cumplen todas estas condiciones se permite el transporte de un vehículo hacia el siguiente punto de trabajo.

Por otro lado, en algunas estaciones se realizan procesos especiales, para etiquetar características propias de los productos y/o vehículos:

- A. En la ST3, que corresponde a la estación de inspección, se genera un código en lenguaje SCL (Figura 2-41) para comandar una cámara industrial (Figura 2-43) y de esta forma analizar las características del producto (determinar el color). Los resultados obtenidos son almacenados en un "Bloque de datos" utilizando una matriz unidireccional tipo "Array" como contenedor. Por otro lado, se utiliza un sensor inductivo que por software es conectado a un contador ascendente para poder determinar el número del carro que llega a la estación. De este modo, se puede codificar el número de cada carro con el producto que este lleva.
- B. En la ST5, o bien, estación de almacenamiento, de forma inversa se utiliza un sensor del tipo inductivo para determinar el número del carro y utilizar el valor obtenido como puntero y de esta forma buscar en el bloque de datos la información respectiva del producto que contiene dicho vehículo. Cuando el análisis ha finalizado, se habilita el almacenamiento selectivo.

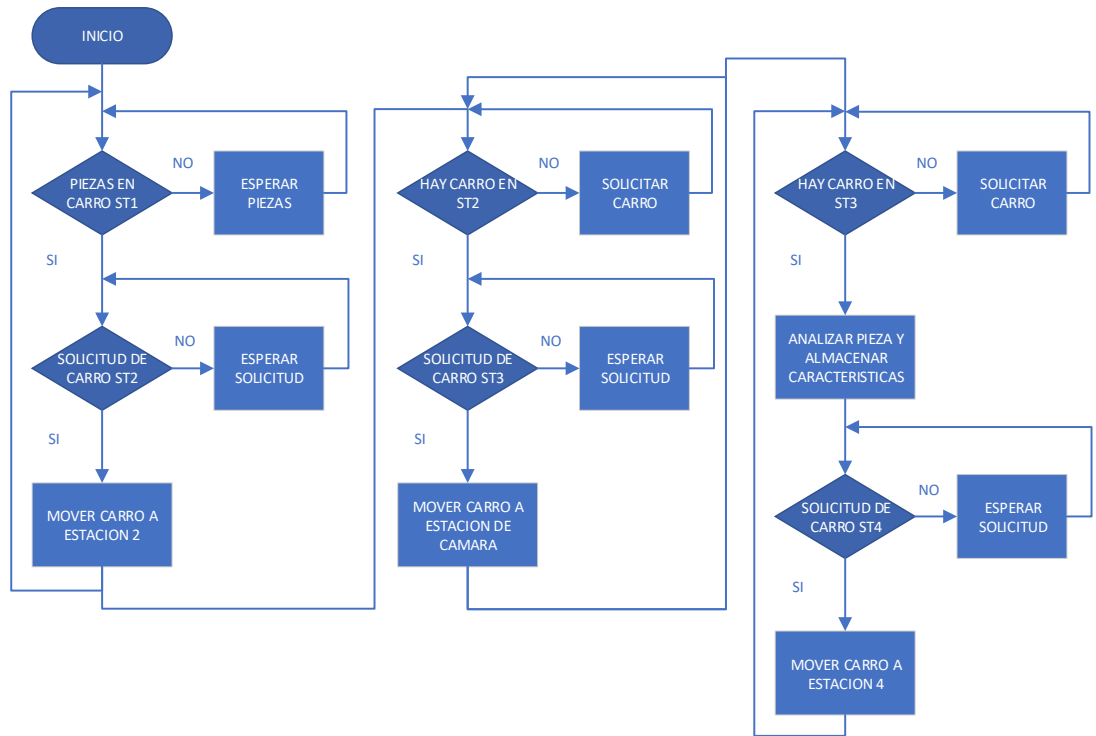


Figura 2-31. Diagrama de flujo estación de transporte desde ST1 a ST3

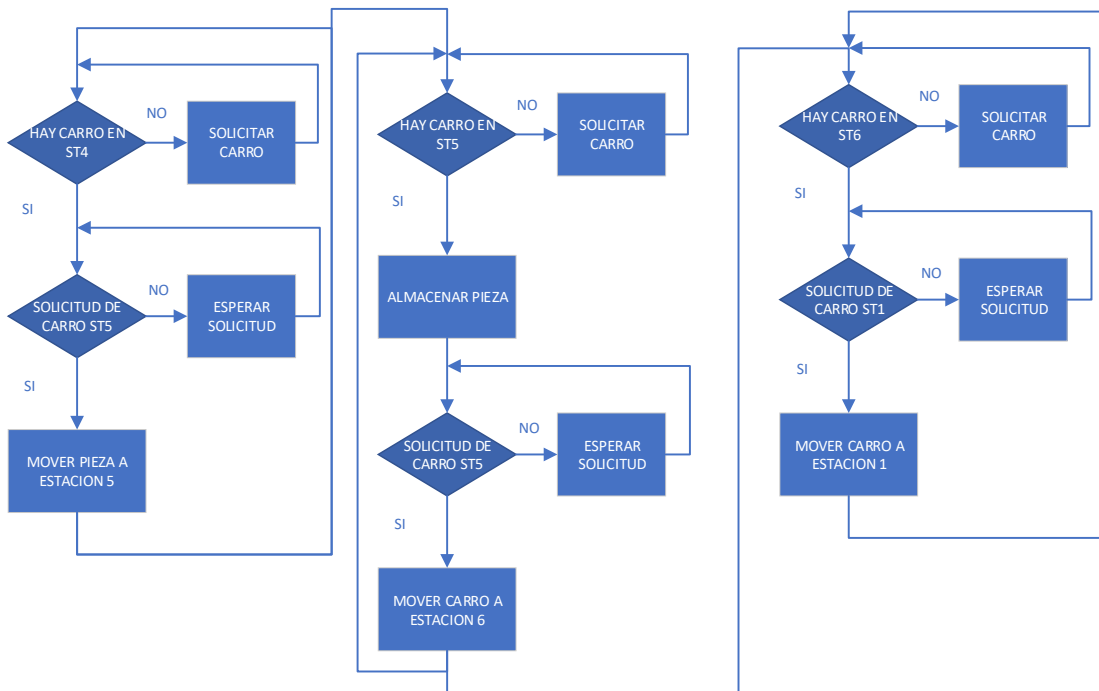


Figura 2-32. Diagrama de flujo estación de transporte desde ST4 ST6

2.3.3 Levantamiento

Para poder comenzar con la programación de la estación, es necesario determinar los sensores y actuadores involucrados en el proceso, en la tabla 2-4 se detalla cada elemento, con su respectivo direccionamiento.

Tabla 2-4. Sensores estación de transporte

Sensores y Actuadores				
ESTACION 1/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
10B1	257.4	INDUCTIVO	TOPE DE CARRO	I157.4
10B3	257.6	LASER	INDICA PRESENCIA DE PIEZA	I157.6
10B4	257.7	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I157.7
10B5	257.2	MAGNETICO	SENSOR IN/OUT ACTUADOR	I157.2
10B2	257.5	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I157.5
ESTACION 2/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
20B1	258.0	INDUCTIVO	TOPE DE CARRO	I158.0
20B3	258.2	LASER	INDICA PRESENCIA DE PIEZA	I158.2
20B4	258.3	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I158.3
20B5	259.6	MAGNETICO	SENSOR IN/OUT ACTUADOR	I159.6
20B2	258.1	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I158.1
ESTACION 3/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
30B1	260.4	INDUCTIVO	TOPE DE CARRO	I160.4
30B3	260.6	LASER	INDICA PRESENCIA DE PIEZA	I160.6
30B4	260.7	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I160.7
30B5	260.2	MAGNETICO	SENSOR IN/OUT ACTUADOR	I160.2
30B2	260.5	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I160.5
ESTACION 4/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
40B1	261.0	INDUCTIVO	TOPE DE CARRO	I161.0
40B3	261.2	LASER	INDICA PRESENCIA DE PIEZA	I161.2
40B4	261.3	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I161.3
40B5	262.6	MAGNETICO	SENSOR IN/OUT ACTUADOR	I162.6
40B2	261.1	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I161.1
ESTACION 5/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
50B1	263.4	INDUCTIVO	TOPE DE CARRO	I163.4
50B3	263.6	LASER	INDICA PRESENCIA DE PIEZA	I163.6
50B4	263.7	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I163.7
50B5	263.2	MAGNETICO	SENSOR IN/OUT ACTUADOR	I163.2
50B2	263.5	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I163.5
ESTACION 6/ETIQUETA	ENTRADAS PERIFERICAS ASI (I)	TIPO DE SENSOR	COMENTARIO	DIRECCION
60B1	264.0	INDUCTIVO	TOPE DE CARRO	I164.0
60B3	264.2	LASER	INDICA PRESENCIA DE PIEZA	I164.2
60B4	264.3	INDUCTIVO	CUENTA EL NÚMERO DEL CARRO	I164.3
60B5	265.6	MAGNETICO	SENSOR IN/OUT ACTUADOR	I165.6
60B2	264.1	INDUCTIVO	INDICA QUE SALIÓ DE ESTACIÓN	I164.1

2.3.4 Programación en TIA Portal V14 SP1

Considerando como referencia los datos de la tabla 2-4 para el direccionamiento de los dispositivos periféricos, la secuencia de operación definida en la figura 2-31 y 2-32 y la selección del lenguaje adecuado para la programación del proceso, que en este caso es LADER y SCL, pues se trata de una lógica paralela de cada estación. El procedimiento para la programación de esta etapa es el siguiente:

Es necesario configurar el hardware existente, esto se debe hacer desde el árbol de proyecto y seguir el procedimiento que se detalla a continuación:

- a. Agregar un nuevo dispositivo

- b. Seleccionar la CPU que se va a utilizar que corresponde a la 314C-2 PN/DP.
- c. Agregar los módulos adicionales, que para esta estación son una fuente de poder modelo PS-307-5A, un módulo AS-interface y un módulo de simulación que será configurado como 16 entradas digitales.
- d. Verificar el direccionamiento que TIA Portal asigna a cada módulo, en este caso, para las entradas y salidas se utilizan los bytes 136 y 137, mientras que el módulo de simulación utiliza los bytes 0 y 1.
- e. Direccionar el puerto Ethernet, en este caso corresponde a la IP 10.4.5.84 con máscara 255.255.255.0.

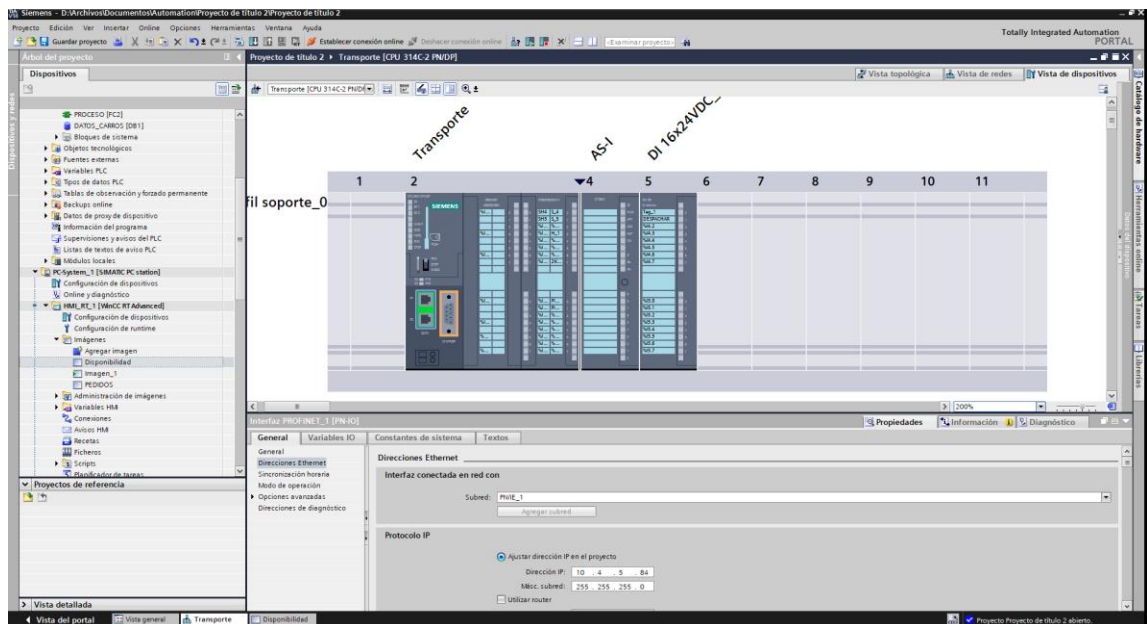


Figura 2-33. Hardware estación Transporte

2.3.4.1 Lectura y escritura de esclavos AS-interface

Inicialmente es necesario crear los bloques de datos necesarios y las subrutinas que se utilizarán en los distintos procesos, utilizando los mismos procedimientos que en las estaciones anteriores (Adquisición y Despacho). Posteriormente es importante entender el funcionamiento del módulo AS-i 343-2AH01-0XA0 y el modo de operación de la CPU S7-300 314C.

En el instante en que el PLC se pasa desde el estado STOP a RUN se ejecuta solo una vez el bloque de arranque OB100. En la figura 2-34 se observa el diagrama con el ciclo de trabajo. Posteriormente se realiza una lectura de los módulos de entradas, cuyos valores son almacenados en una imagen periférica (PAE). Luego se ejecutan el programa principal, subrutinas, y eventos del sistema. Posteriormente se realiza una escritura de la memoria periférica de salida en los módulos de salida (PAA).

Los módulos de entradas y salidas convencionales realizan automáticamente la lectura y escritura según corresponda. Pero en el caso el módulo AS-i no es así, es por ello que se debe realizar una lectura desde la imagen periférica de entrada y escritura en la imagen periférica de salida, con el fin de poder tener el efecto esperado en los dispositivos esclavos.

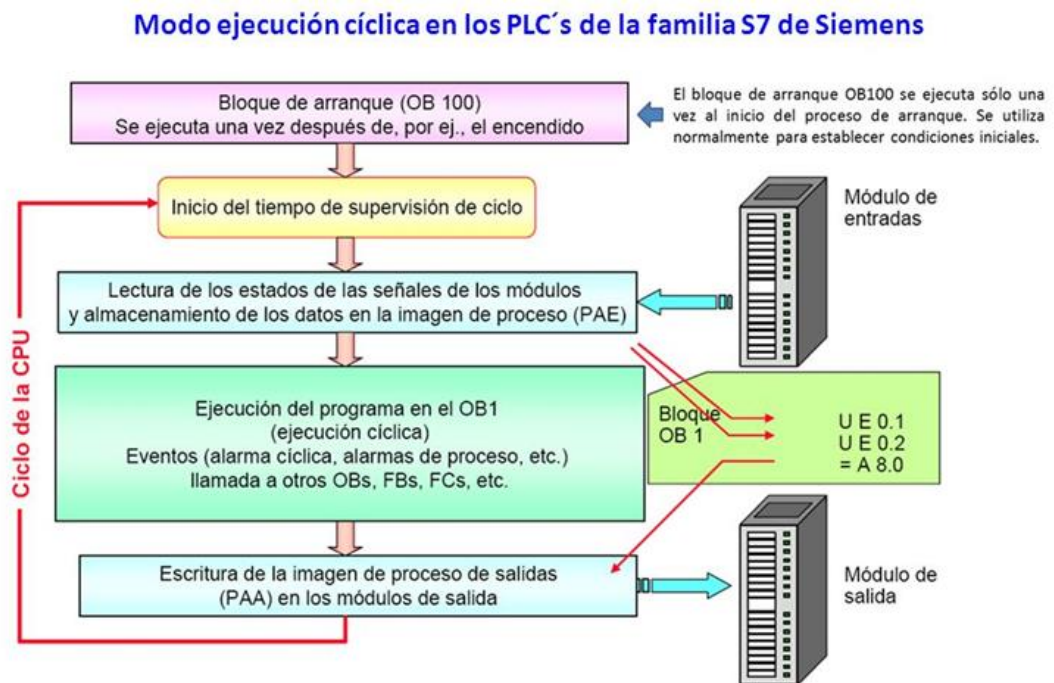


Figura 2-34. Ciclo de trabajo de una CPU Simatic S7-300

Otro aspecto importante y que se debe considerar es que cada módulo AS-i admite hasta 31 esclavos, cada uno utiliza cuatro bits, esto quiere decir que para poder leer y escribir es necesario seleccionar el tipo de dato adecuado para almacenar dichos valores en la tabla 2-5 se resume la cantidad de esclavos que se pueden almacenar en cada tipo.

Tabla 2-5 Esclavos admitidos por tipo datos

Tipo de dato	Longitud (bit)	Cantidad de Esclavos
Byte	8	2
Word	16	4
DWord	32	8

Conociendo estos detalles se puede comenzar con la programación en TIA Portal, para ello se deben seguir los siguientes pasos:

- A. Abrir la vista de dispositivo.
- B. Seleccionar el módulo AS-i y seguir la ruta Propiedades => General => Sinóptico de direcciones.
- C. En la figura 2-35 se puede observar gráficamente la ruta descrita.

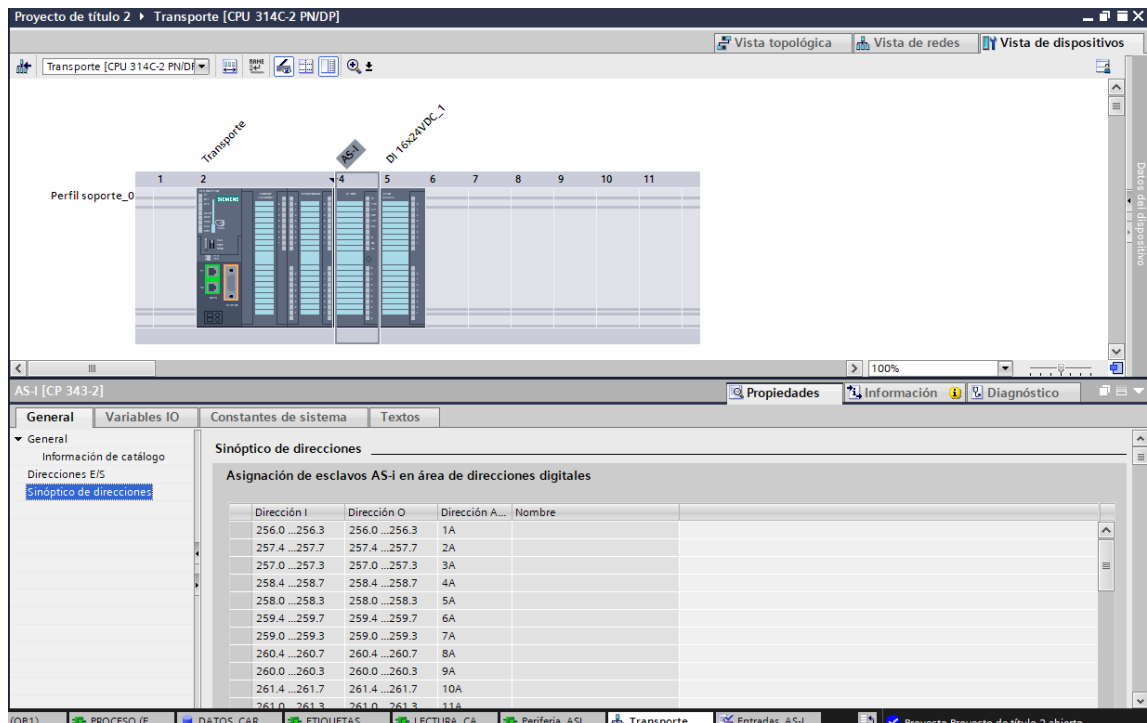


Figura 2-35. Direcciones esclavos AS-i

Luego se debe realizar una lectura y escritura de la imagen periférica según las direcciones determinadas en la figura 2-35, en este caso se comenzará con el valor de memoria 257, ya que el primer esclavo utilizado tiene dirección 2. Por otro lado, se utilizará como tipo de dato DWord, con el fin de leer una mayor cantidad de esclavos con una sola instrucción.

El proceso de lectura de una imagen periférica consiste en ejecutar una instrucción "MOVE" que en su entrada tiene la dirección de los esclavos AS-i que se desea leer seguido de ":P" y en la salida una dirección de memoria de trabajo del PLC. En el costado izquierdo de la figura 2-36 se puede apreciar el proceso de lectura de una imagen periférica.

El proceso para la escritura es inverso al de lectura, es decir, en la entrada de la función MOVE debe ir el valor que se desea escribir en la imagen periférica de salida.

Y en la salida de dicha función debe ir la dirección del módulo AS-i seguido por ":P". En el costado derecho de la figura 2-36 se puede observar la nomenclatura.

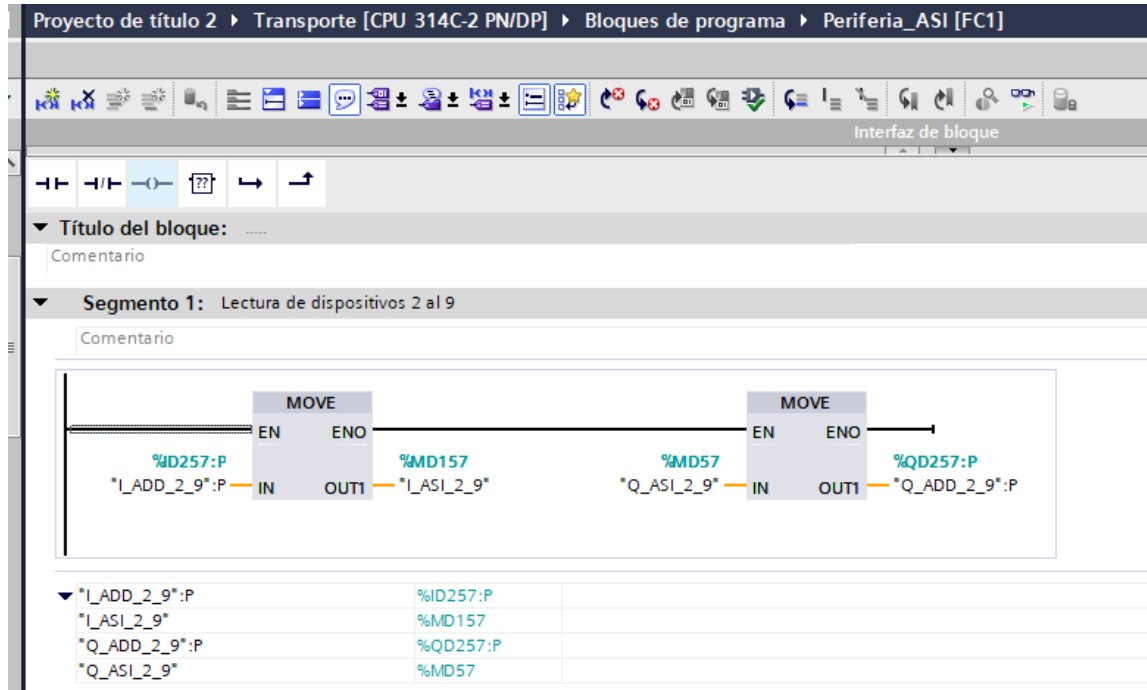


Figura 2-36. Lectura y escritura de esclavos AS-i

2.3.4.2 Estación 1

La estación número 1 (ST1) es el punto de ingreso de las piezas. En este punto es necesario que el carro se posicione correctamente (se debe activar sensor inductivo número 3 de la figura 2-38) y no contenga producto, esto es posible gracias al sensor del tipo laser que se señala con el número 4 en la figura 2-38, de lo contrario es despachado automáticamente hacia la ST2, accionando el movimiento de las cintas transportadoras y activando el cilindro neumático 5 de la figura 2-38. Si el vehículo en espera se encuentra vacío, se da la orden al proceso anterior (Despacho) para que inicie su secuencia.

La figura 2-38 corresponde a la estación número dos, debido a esto, el sensor número 1 está conectado a la ST1 y en este trabajo se utiliza para indicar que la ST2 está saturada, por ende, se debe detener el proceso previo hasta que haya disponibilidad. En la imagen la figura 2-37 se puede apreciar la programación en TIA Portal.

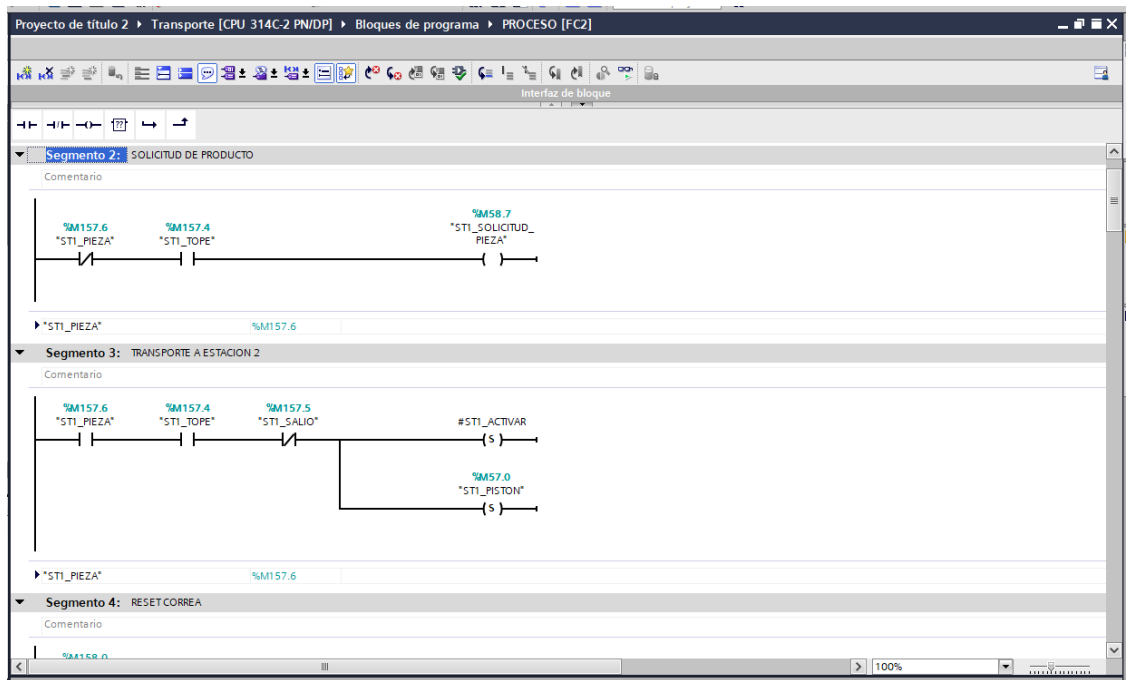


Figura 2-37 Programa estación 1

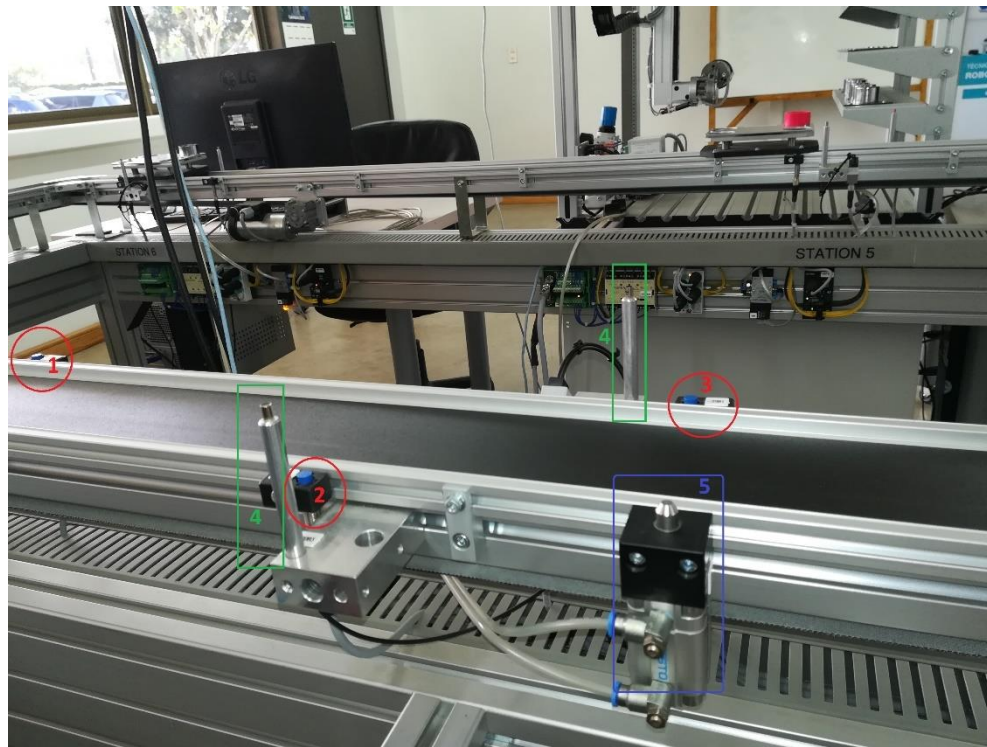


Figura 2-38. Estación número 2

2.3.4.3 Estación 2

La estación 2 trabaja de forma similar a la ST1, con la diferencia de que en esta posición no se realizan procesos, esto quiere decir que el carro que pase por dicha etapa será despachado inmediatamente al menos que se encuentre detenido el proceso en la estación 3. De ser así el sistema se mantendrá en espera hasta que se produzca un espacio. En la figura 2-39 se observa el programa de esta etapa.

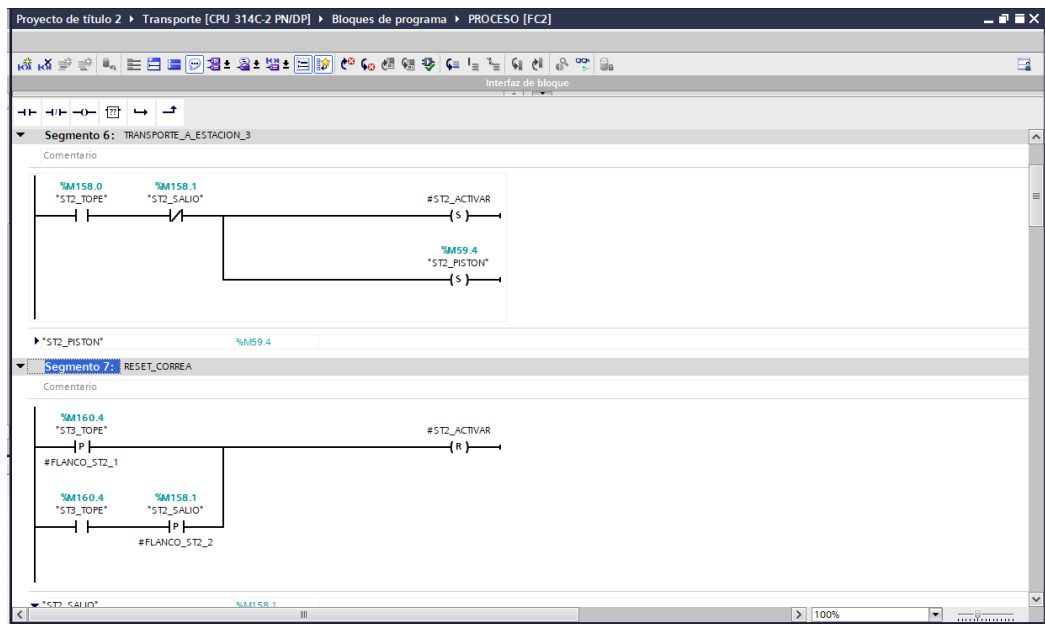
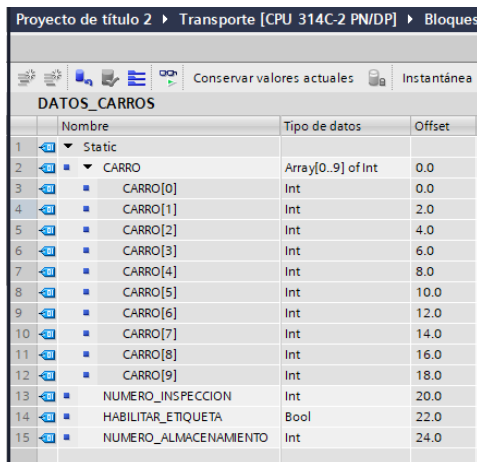


Figura 2-39 Programa estación 2

2.3.4.4 Estación 3

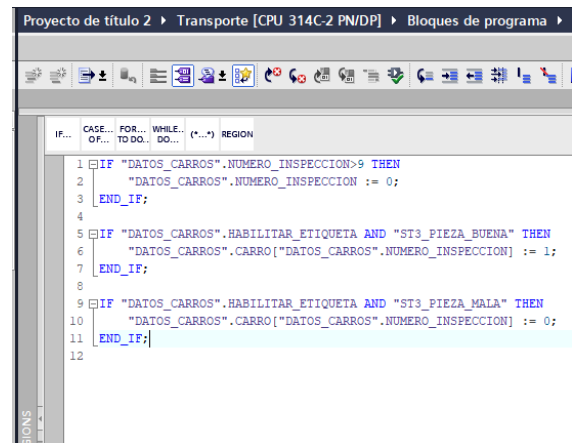
La estación 3 es el punto de inspección y control de calidad, en este lugar se realiza el siguiente proceso:

- A. Cuando un carro se aproxima, mediante el sensor 2 de la figura 2-38 se determina el número identificador, este valor se almacena en un Bloque de datos (Figura 2-40).
- B. Cuando el móvil, se encuentra correctamente posicionado se activa un pulso de 100 ms para señalar a la cámara que debe tomar una muestra.
- C. Con dicha muestra la cámara industrial SBOC-Q-R3C-WB (Figura 2-43) analiza los resultados en base al programa determinado (en este caso si la pieza es de color rojo).
- D. Como resultado se obtiene la activación de una salida digital de la estación de inspección. Con este resultado se almacena en el bloque de datos un valor "1" si la pieza es de color rojo o un "0" si no lo es. En la figura 2-41 se puede observar el código utilizado para etiquetar los móviles.



	Nombre	Tipo de datos	Offset
1	Static		
2	CARRO	Array[0..9] of Int	0.0
3	CARRO[0]	Int	0.0
4	CARRO[1]	Int	2.0
5	CARRO[2]	Int	4.0
6	CARRO[3]	Int	6.0
7	CARRO[4]	Int	8.0
8	CARRO[5]	Int	10.0
9	CARRO[6]	Int	12.0
10	CARRO[7]	Int	14.0
11	CARRO[8]	Int	16.0
12	CARRO[9]	Int	18.0
13	NUMERO_INSPECCION	Int	20.0
14	HABILITAR_ETIQUETA	Bool	22.0
15	NUMERO_ALMACENAMIENTO	Int	24.0

Figura 2-40. Bloque de datos



```
IF... CASE... FOR... WHILE... (*..*) REGION
OF... TO DO... DO...

1 IF "DATOS_CARROS".NUMERO_INSPECCION>9 THEN
2   "DATOS_CARROS".NUMERO_INSPECCION := 0;
3 END_IF;
4
5 IF "DATOS_CARROS".HABILITAR_ETIQUETA AND "ST3_PIEZA_BUENA" THEN
6   "DATOS_CARROS".CARRO["DATOS_CARROS".NUMERO_INSPECCION] := 1;
7 END_IF;
8
9 IF "DATOS_CARROS".HABILITAR_ETIQUETA AND "ST3_PIEZA_MALA" THEN
10  "DATOS_CARROS".CARRO["DATOS_CARROS".NUMERO_INSPECCION] := 0;
11 END_IF;
12
```

Figura 2-41. Etiqueta de carros

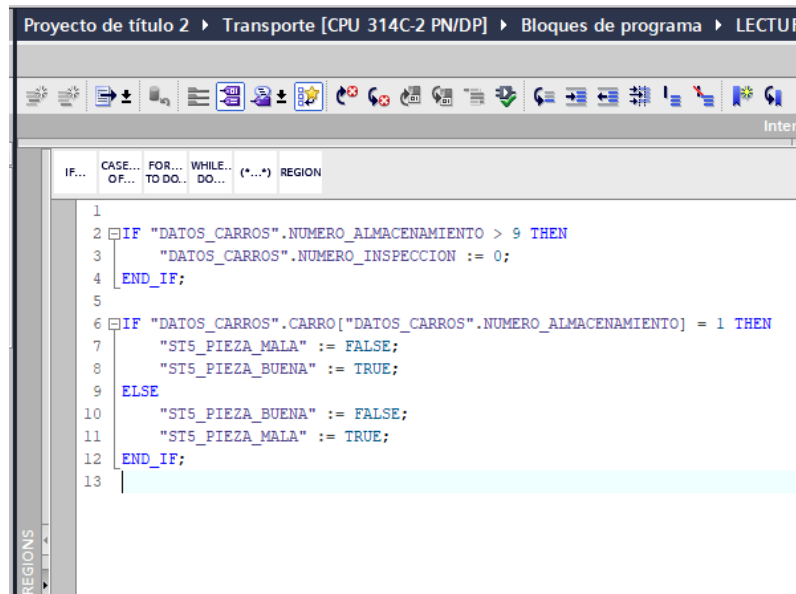
2.3.4.5 Estación 4

La estación 4 utiliza la misma lógica de operación que la ST2, ya que se encuentra vacía. Debido a esto, solo bloquea el paso de los carros cuando la ST5 se encuentra saturada.

2.3.4.6 Estación 5

La estación 5 o de almacenamiento (Figura 2-59) cuenta con un almacén de treinta y cinco posiciones (siete horizontales y cinco niveles verticales). Utiliza un brazo robotizado de tres grados de libertad (X, Y, Z) con un gripper neumático para tomar el producto desde el carro. El siguiente procedimiento detalla la operación de la ST5:

- A. Cuando un carro se aproxima, mediante el sensor 2 de la figura 2-35 se determina el número identificador, este valor se almacena en un Bloque de datos (Figura 2-40).
- B. Cuando el vehículo se encuentra correctamente posicionado busca en el bloque de datos si el contenido corresponde o no a una pieza de color rojo (Figura 2-42).
- C. Con el resultado se da la orden a la estación de almacenamiento para tomar y guardar según el criterio seleccionado. Para indicar que la pieza es de color rojo y dar la orden de almacenar se utiliza la interfaz SYS link.



```
1
2 IF "DATOS_CARROS".NUMERO_ALMACENAMIENTO > 9 THEN
3   "DATOS_CARROS".NUMERO_INSPECCION := 0;
4 END_IF;
5
6 IF "DATOS_CARROS".CARRO["DATOS_CARROS".NUMERO_ALMACENAMIENTO] = 1 THEN
7   "ST5_PIEZA_MALA" := FALSE;
8   "ST5_PIEZA_BUENA" := TRUE;
9 ELSE
10  "ST5_PIEZA_BUENA" := FALSE;
11  "ST5_PIEZA_MALA" := TRUE;
12 END_IF;
13
```

Figura 2-42 Lectura etiqueta de carros

2.3.4.7 Estación 6

La estación 6 utiliza la misma lógica de operación que la ST2 y la ST4, ya que se encuentra vacía. Debido a esto, solo bloquea el paso de los carros cuando la ST1 se encuentra saturada.

2.4 ESTACIÓN DE INSPECCIÓN

Según un artículo desarrollado por la universidad de Santiago en el año 2002, "La inspección visual automática (AVI) se define como un proceso de control de calidad que, mediante técnicas de procesamiento digital de imágenes y reconocimientos de patrones, determina automáticamente si un producto se desvía de las especificaciones de fabricación" (Mery, 2002). Así es posible garantizar que los productos que aprueben el control de calidad cumplan con las características establecidas para su producción.

Los sistemas AVI que se utilizan para el control de calidad de un producto debe cumplir principalmente los siguientes dos criterios:

- **Eficiencia:** Se debe detectar el total o mayor cantidad de productos defectuosos y simultáneamente rechazando la menor cantidad de productos en buen estado, de acuerdo con las especificaciones de producción.
- **Rapidez:** La velocidad de producción debe ser acorde a los ritmos de producción, de tal forma que la inspección no desacelere el proceso productivo.

Para el análisis de un producto se pueden involucrar múltiples criterios de inspección basados en elementos patrones. Gracias a esto es posible comparar diversas características como lo son:

- Dimensiones físicas (diámetro, radio, área, etc.)
- Imperfecciones en las superficies.
- Geometría de un producto

2.4.1 Módulo de visión de FMS



Figura 2-43. Estación de visión USM

La celda flexible de la USM sede Concepción cuenta con una estación de inspección. Cuya función es analizar ciertas propiedades de las piezas. A continuación, las partes que componen la estación (SCHOBER, 2010).

A. Cámara de la serie SBO-Q:

Es una cámara industrial de la marca FESTO (Figura 2-43), modelo SBOC-Q-R3C-WB, entre sus principales características se pueden destacar:

- Identificación de piezas.
- Localización de piezas.
- Detección de Color.
- Lectura de códigos 1D y 2D.
- Entradas y salidas digitales.
- Puerto Ethernet (Con soporte para ModBus).
- Resolución 752x480.
- Capacidad máxima de 256 programas de inspección.
- Capacidad máxima de 256 características por programa de inspección.

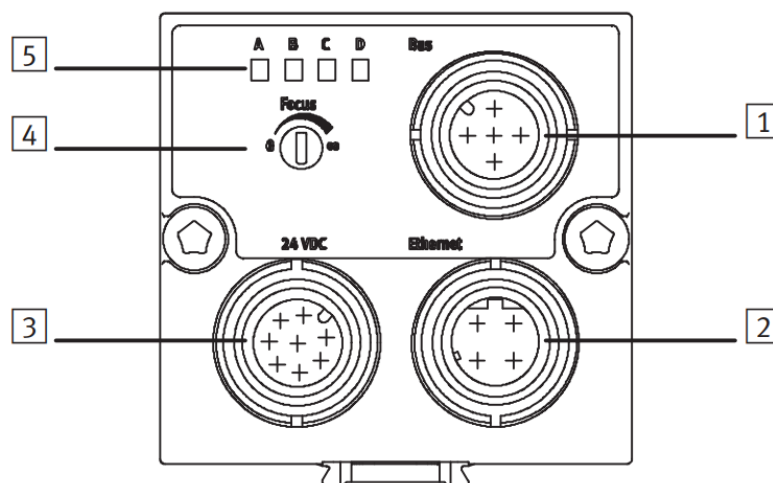


Figura 2-44 Conexiones eléctricas

En la figura 2-44 se indica cada uno de los puertos de conexiones eléctricas, además de los dispositivos de indicación y ajuste. A continuación, se indica la función de cada uno de ellos (Schober, 2010):

1. Conexión de bus de campo del sistema de visión artificial SBO-Q.
2. Conexión Ethernet.
3. Voltaje de alimentación, entradas y salidas digitales.
4. Potenciómetro para ajuste de Focus
5. Leds de estado:

- A. Cámara lista para funcionar.
- B. Indica si existe tráfico Ethernet.
- C. Indicador de actividad.
- D. Indicador de entrega.

B. Interfaz SYS Link

De igual forma que la estación de distribución se utiliza una interfaz SYS Link como medio de comunicación con la interfaz AS-i de la estación de transporte (Figura 2-45).

1. Conector SYS Link.
2. Terminal de entradas y salidas (se conecta a un módulo esclavo AS-i en la estación de transporte).
3. Cable SYS Link.

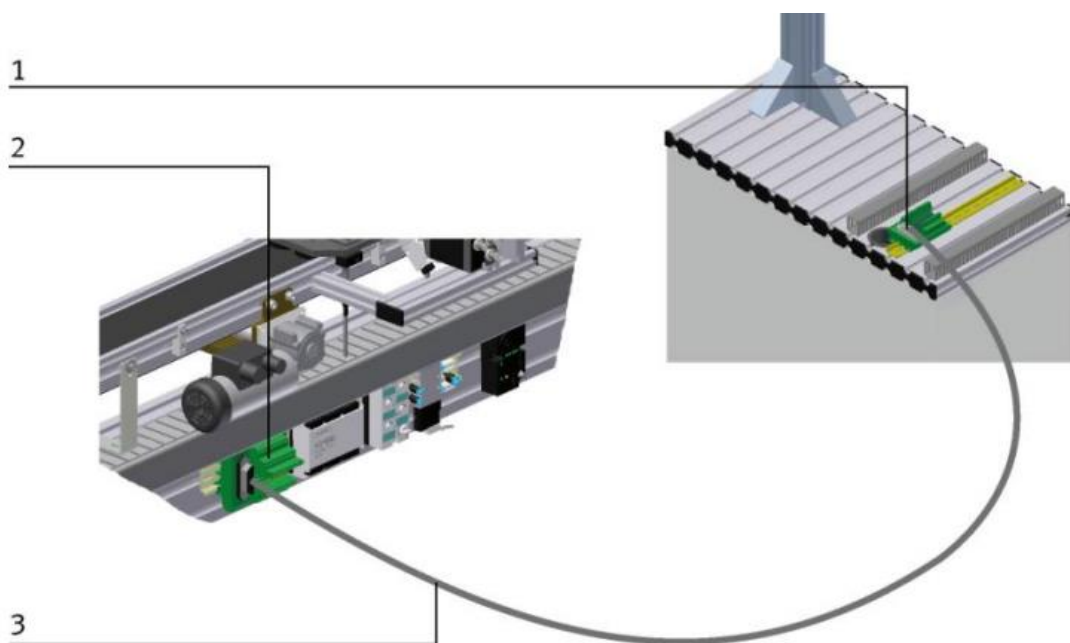


Figura 2-45. Conexión mediante interfaz SYS Link

2.4.2 Programación de cámara de Visión

Para poder configurar completamente la cámara FESTO dispone de un paquete de software donde programar distintas características en el proceso de evaluación. A continuación, se detalla la función de cada uno de ellos:

- **SBO-Device Manager:** Es una interfaz que permite configurar la dirección de red del dispositivo.
- **CheckKon:** Este software es utilizado para ver, documentar y establecer el modo de operación de la cámara. Entre sus principales funciones se pueden señalar:
 - Indicación y modificación de parámetros del sistema.
 - Selección de nuevos programas de control.
 - Configuración de entradas y salidas.
 - Documentación del sistema.
- **CheckOpti:** Esta aplicación se utiliza para determinar criterios para los programas de control. Su operación es simple y es posible analizar directamente muestras de referencia o patrones.
- Además, este software permite cargar a uno de los 256 espacios de memoria el programa creado.

2.4.2.1 Configuración de Dirección de Red

Inicialmente se debe configurar la dirección de red asignada por la institución, ya que el laboratorio se encuentra en una VLAN con direcciones IP clase A, en este caso se asigna la IP 10.4.5.96 con máscara de subred 255.255.255.0. para esto se debe seguir el siguiente procedimiento:

- Configurar la tarjeta de red del computador que contiene el paquete de software de FESTO dentro del rango de la VLAN, en este caso se asigna 10.4.5.85.
- Abrir el software SBO-DeviceManager y presionar el botón "Search" (Se destaca con color rojo en la figura 2-46) para buscar los dispositivos disponibles en la red.

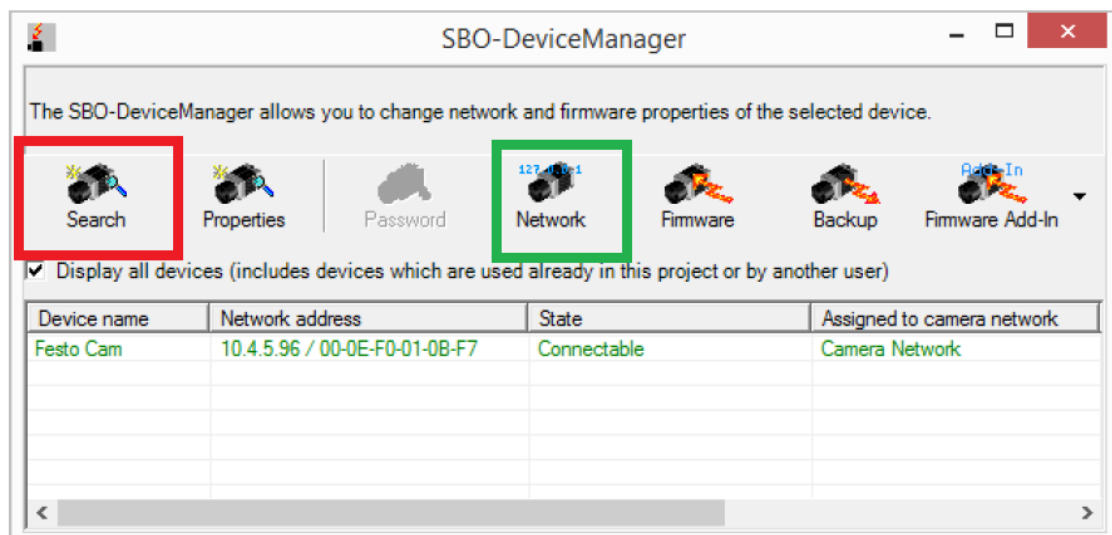


Figura 2-46. Software SBO-DeviceManager, dispositivos disponibles en la red

- C. Luego se debe marcar el dispositivo que se desea configurar (en este caso solo aparece uno) y presionar el botón "Network" (destacado con color verde en la figura 2-46).
- D. Finalmente, en la ventana "Change network setting of the device" se deben ingresar los parámetros correspondientes al proyecto y presionar el botón "OK" para guardar los datos. En la figura 2-47 se observa la interfaz de configuración.

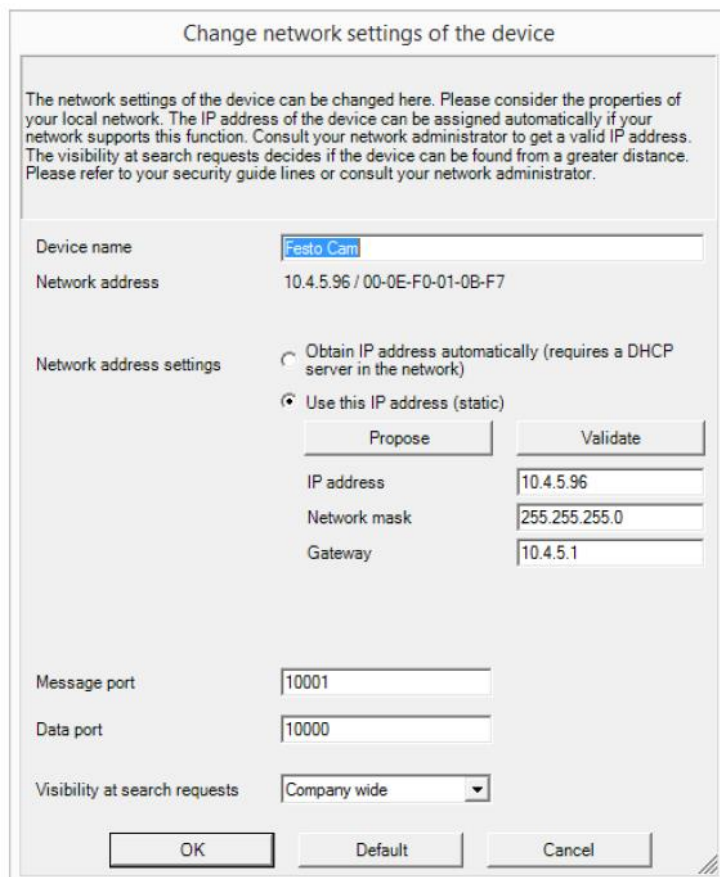


Figura 2-47. Interfaz de configuración de red

2.4.2.2 Configuración de programa de inspección

El software utilizado es CheckOpti, este software posee múltiples funciones, pero en esta memoria solo se explicará el proceso utilizado para determinar si un producto es o no de color rojo.

- A. Para comenzar a configurar los criterios de discriminación es necesario abrir el software y crear un nuevo proyecto (Figura 2-48), luego se debe presionar "Next".

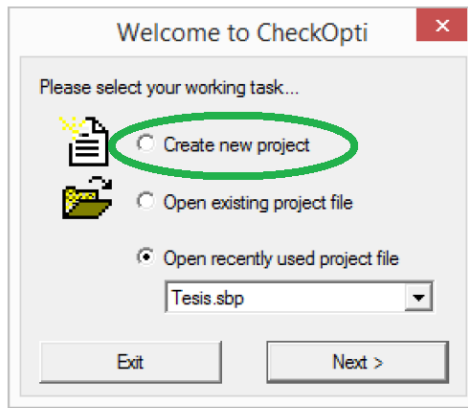


Figura 2-48. Ventana "Welcome to CheckOpti"

- B. Aparecerá una nueva ventana (figura 2-49), la que corresponde a la configuración de comunicación con la cámara, aquí se deben configurar los datos correspondientes según el proyecto como el modelo del dispositivo y la interfaz Ethernet.

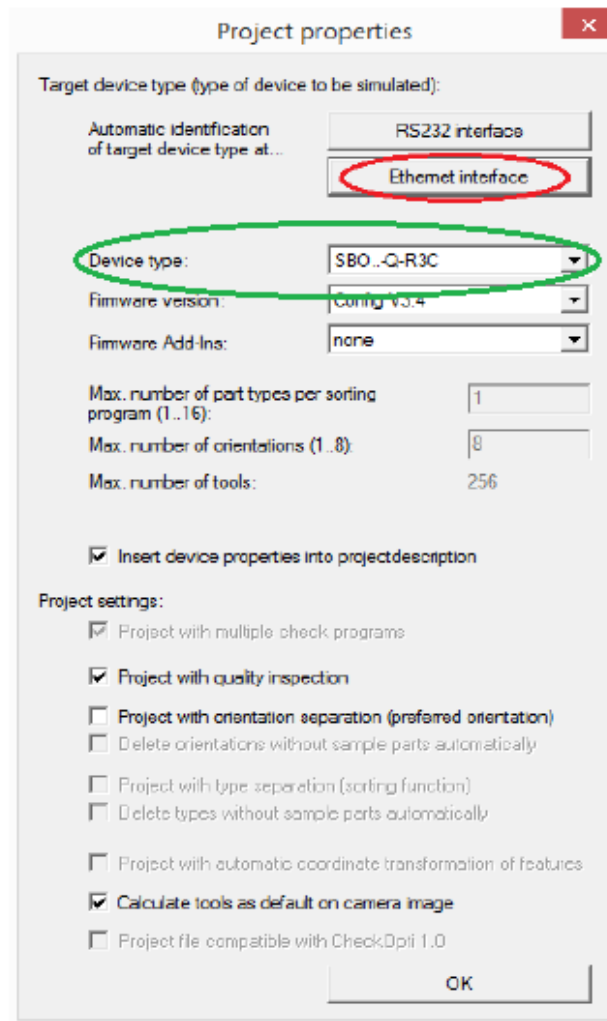


Figura 2-49. Ventana de propiedades del proyecto

C. Identificar las ventanas de configuración que se utilizarán para el análisis de color. En la figura 2-50 se muestra la barra de herramientas de CheckOpti y se destacan los botones que se utilizarán.

1. Part List
2. Part Contour
3. Teach-Data
4. Check programm manager
5. Communication manager
6. Tool properties



Figura 2-50. Barra de herramientas CheckOpti

D. Antes de crear los filtros, es necesario tomar muestras de objetos patrones, con el fin de establecer un criterio real y compatible con la instalación física de la cámara. Para esto se debe seguir el siguiente procedimiento:

- Desplegar la ventana "Part List" (número 1 según figura 2-50).
- Presionar el botón 1 de la figura 2-51 que corresponde a "New picture".
- Posteriormente aparecerá la ventana "Communication Manager", aquí se deben seleccionar los parámetros igual que la figura 2-52. Una vez configurado se debe presionar el botón "Trigger" (Marca verde de figura 2-52), esto envía una orden a la cámara para tomar una muestra y almacenarla para su posterior análisis. Se debe repetir este proceso las veces necesarias según la cantidad de muestras que se desean analizar.

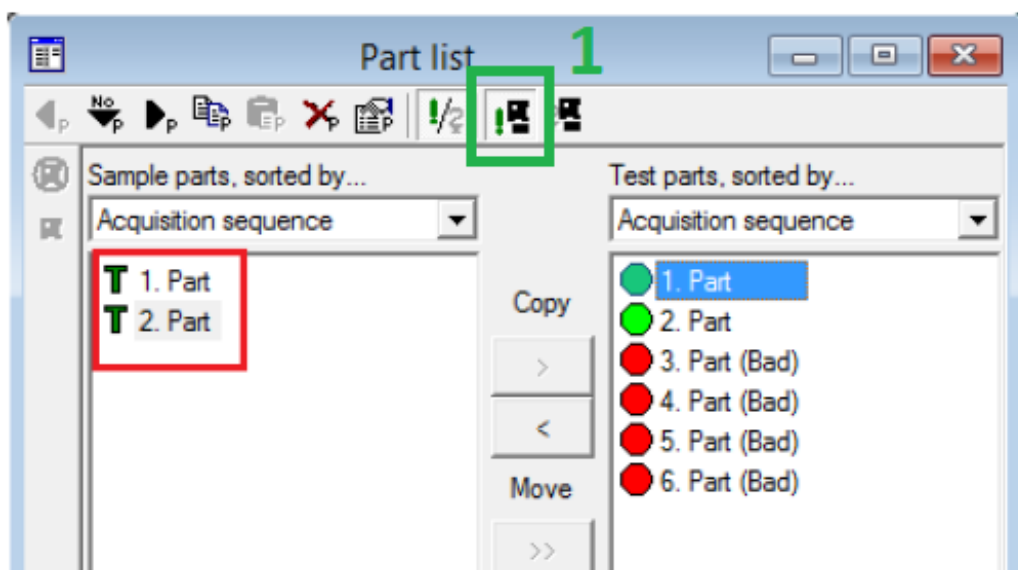


Figura 2-51. Ventana "Part List"

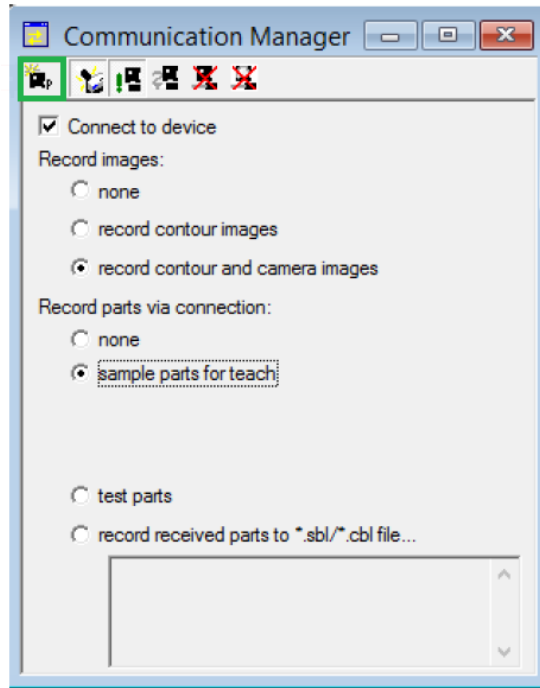


Figura 2-52. Ventana "Communication Manager"



Figura 2-53. Barra de funciones de Part Contour

- E. El siguiente paso es crear un filtro, para lograrlo se debe seleccionar la ventana "Part Contour" en la barra de herramientas (Figura 2-50). Aparecerá un nuevo menú y dentro de él una barra de funciones (Figura 2-53), allí se debe seleccionar el botón resaltado en rojo para proceder a configurar el filtro RGB.

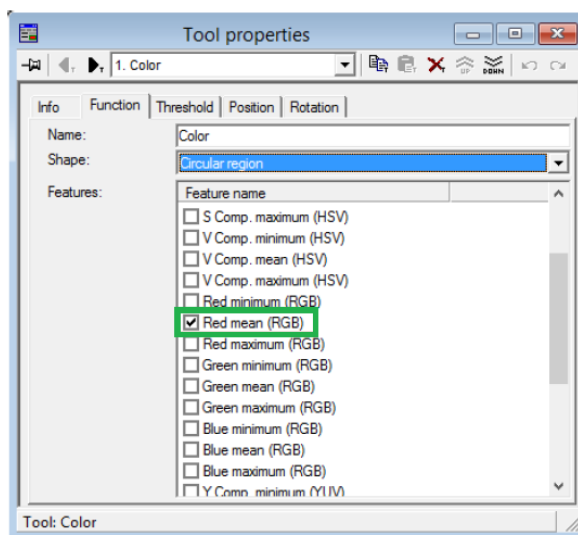


Figura 2-54. Ventana de selección de filtro RGB

- F. En la figura 2-54 se muestra la ventana de configuración del filtro, en este caso se seleccionará "Red mean (RGB)".
- G. Luego se debe configurar las salidas digitales para poder indicar a la estación de transporte si la pieza cumple o no con el criterio de inspección. Esto se realiza presionando el botón "Tool properties" indicado en la figura 2-50, en esta memoria se configuran dos salidas de igual forma que en la figura 2-55, donde se resaltan los parámetros que se deben modificar.
- Salida 1: Se activa cuando la muestra evaluada cumple con el criterio de inspección, es decir, la pieza es de color rojo
 - Salida 2: Se activa cuando la muestra no cumple con los parámetros establecidos.

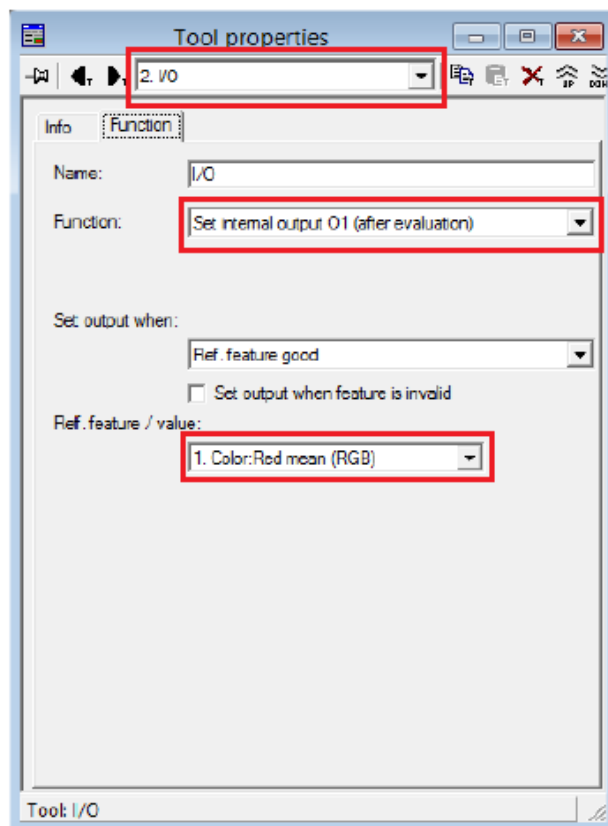


Figura 2-55. Herramienta de propiedades I/O Access

- H. Finalmente, cuando se ha terminado la programación se debe descargar el proyecto en un espacio de memoria de la cámara, para esto se debe presionar el botón "Check program manager". En esta ventana (Figura 2-56) se selecciona el programa que se desea transferir y luego se selecciona la opción "copy to device".

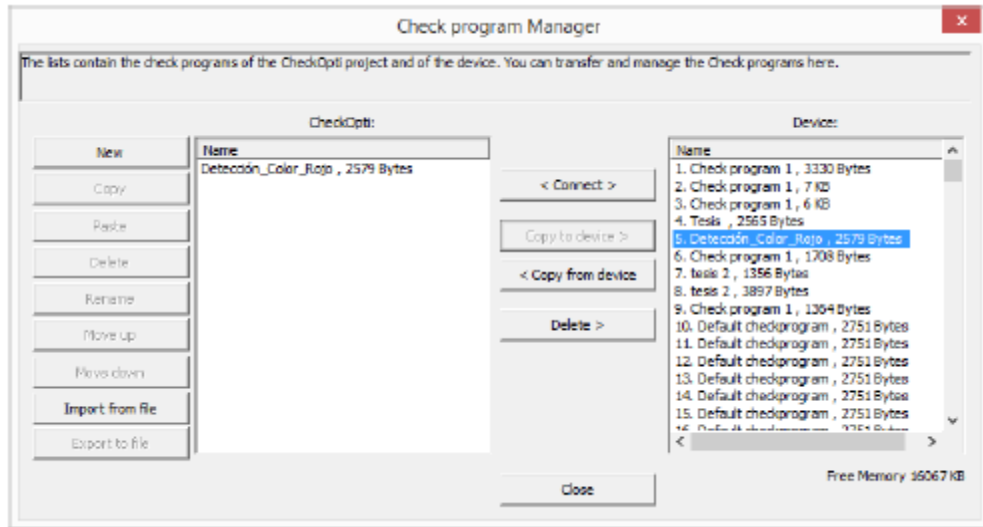


Figura 2-56. Interfaz "Check program manager"

2.4.2.3 Configuración de operación de la cámara

El software CheckKon entre otras cosas permite configurar el modo de operación de la cámara, en este caso se modificará el programa de evaluación que se está ejecutando. Para esto se debe presionar el botón "System parameter" en la barra de herramientas y luego se debe reemplazar en valor del programa de inspección seleccionado según corresponda. En la figura 2-27 se observa gráficamente.

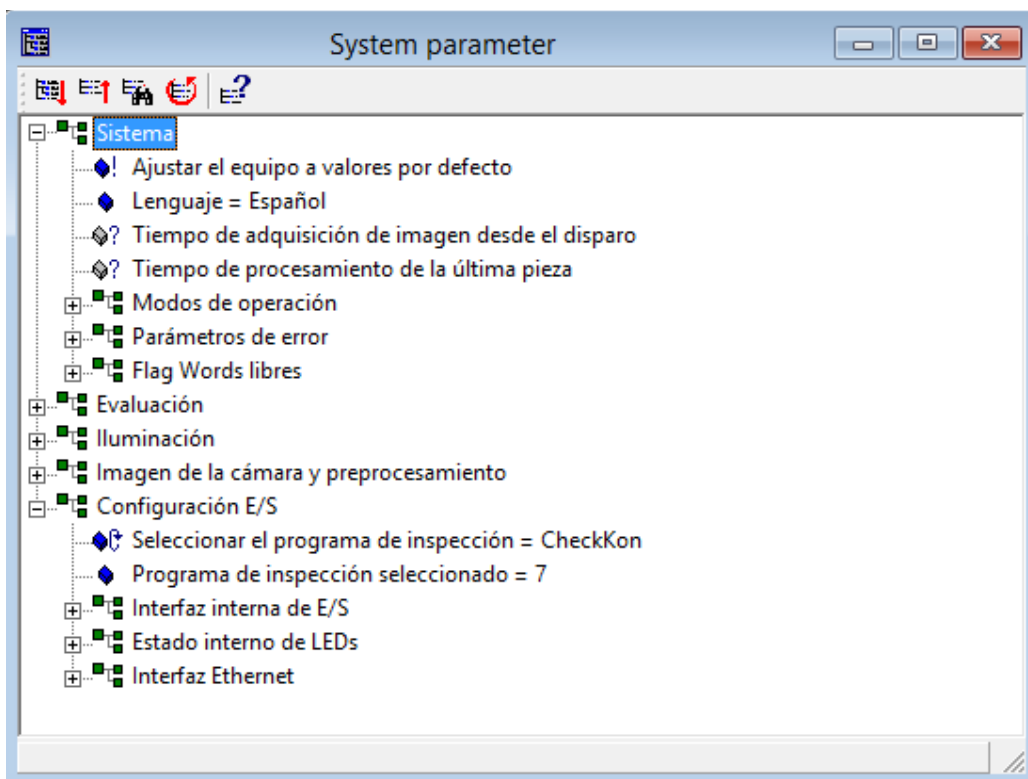


Figura 2-57. Ventana "System parameter"

2.5 ESTACIÓN DE ALMACENAMIENTO

Los sistemas de manufactura flexible se caracterizan por utilizar un muy alto nivel de automatización, es por lo que la mayoría de las veces cuentan con sistemas de almacenaje automático, gracias a esto se puede conseguir una optimización de dicho proceso y no cuellos de botella.

Industrialmente son sistemas que están diseñados para ser operados automáticamente por medio de transelevadores o paletas, los que permiten el almacenamiento y despacho de pallets o cajas.

Para desplazarse alrededor del almacén utilizan elevadores robotizados, los que permiten ganar altura y desplazarse rápidamente y a su vez permiten la manipulación de la carga.

Otra característica importante es que los operadores no requieren desplazarse por los pasillos, esto produce un aumento en la seguridad y se disminuye la probabilidad de accidentes.

Algunas de las ventajas de los sistemas de almacenamiento son:

- Reducción de costos.
- Máximo control de stock ya que se elimina el error humano.
- Preparación de los pedidos con gran rapidez.
- Reducción de superficie en los almacenes, ya que se aumenta la altura.
- Incremento de la productividad de la actividad logística.
- Menor costo de mantenimiento, debido a reducción de daños.
- Aumento de la seguridad del personal y de las mercancías.



Figura 2-58. Almacén automático

2.5.1 Estación de almacenamiento AS-RS

La estación AS-RS de la FMS cuenta con un almacén de treinta y cinco posiciones (siete horizontales y cinco niveles verticales). Utiliza un brazo robotizado de tres grados de libertad (X, Y, Z) para desplazarse en el rack de almacenamiento, dos de estos ejes cuentan con un motor eléctrico acoplados a una caja reductora para producir el movimiento, mientras que el tercero cuenta con un cilindro neumático de simple efecto, además posee un gripper neumático para tomar y depositar el producto.

La función principal de esta estación es tomar las piezas que llegan a la estación 5 del proceso de transporte y almacenarlas según el criterio programado, en este caso de forma selectiva por color. Además, la estación es capaz de despachar mercancías cuando el computador principal lo requiere.

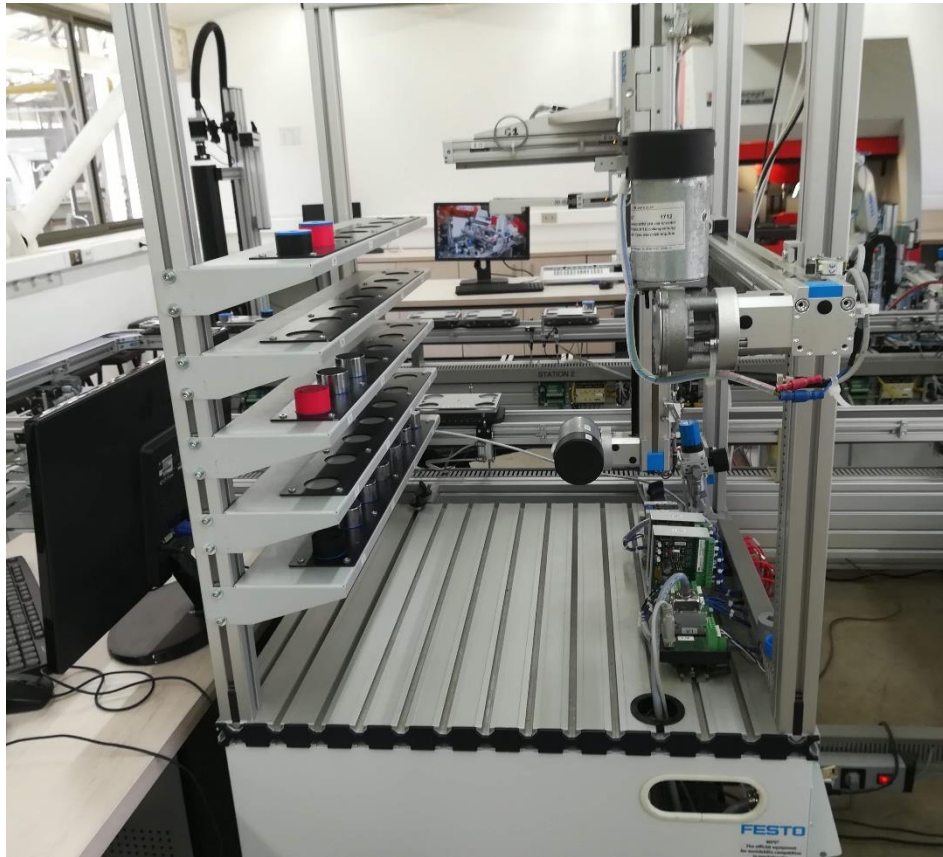


Figura 2-59. Estación de almacenamiento

2.5.2 Partes de la estación AS-RS

A continuación, se explicará brevemente la función que cumple cada componente de la estación de almacenamiento.

2.5.2.1 Rack

El rack es una estructura estática que tiene como función servir de contenedor y así almacenar los componentes producidos por la celda de manufactura. Se compone de múltiples niveles verticales, obteniendo de esta forma una gran capacidad de almacenamiento en una superficie reducida. En el caso de la estación AS-RS se pueden utilizar hasta treinta y cinco posiciones en el almacén.

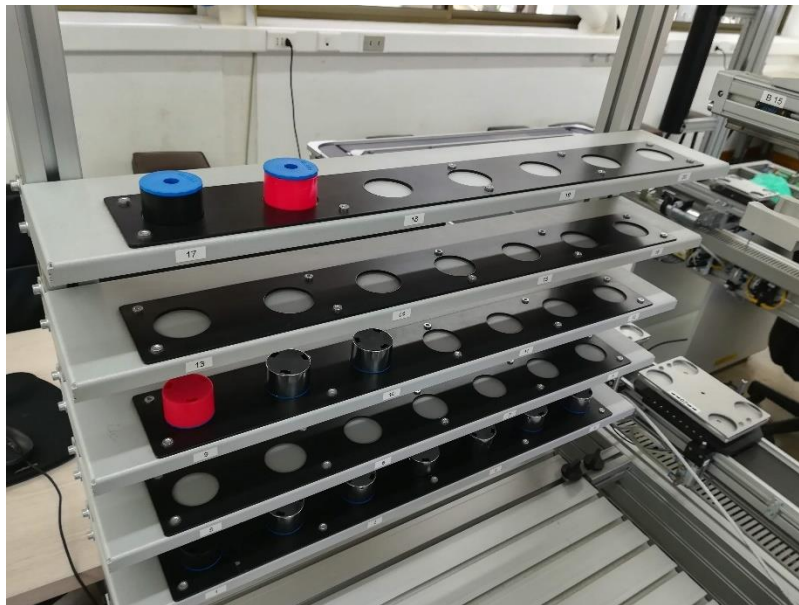


Figura 2-60. Rack de almacenamiento

2.5.2.2 Brazo móvil de tres ejes

El brazo móvil es el encargado de tomar un producto desde un carro para luego almacenarlo en algún lugar dentro del almacén. O bien, tomar un producto de un almacén para depositarlo en un carro para ser despachado. Para poder realizar dicha tarea necesita de un conjunto de cuatro elementos:

- Gripper Neumático.
- Cilindro neumático para el eje Y.
- Motor eléctrico para el eje X.
- Motor eléctrico para el eje Z.

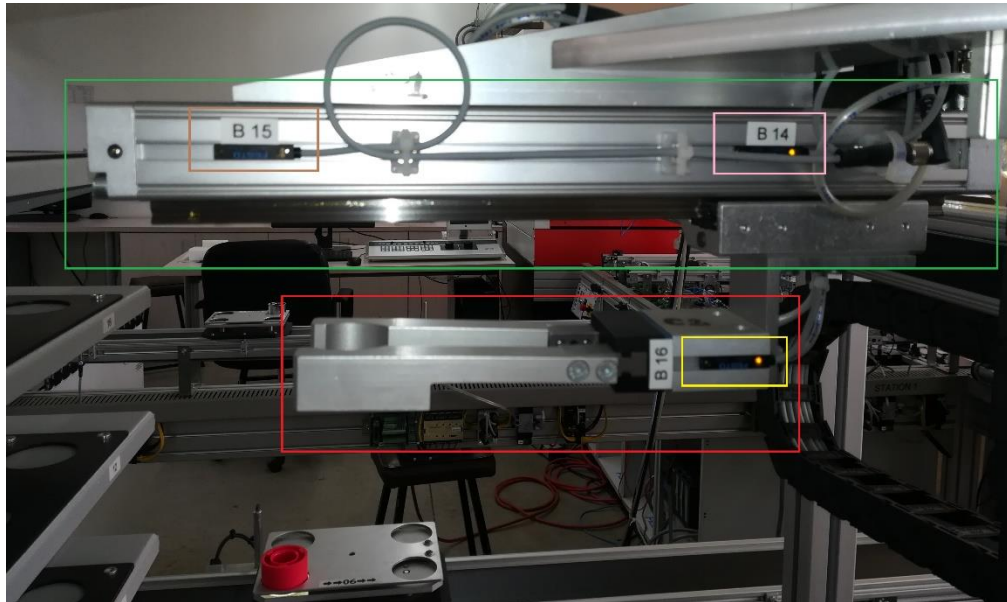


Figura 2-61. Brazo móvil

2.5.2.3 Cilindro neumático

El cilindro neumático tiene como función desplazar el gripper hacia el carro o el rack para tomar o depositar un producto. En la figura 2-61 se puede observar dentro del recuadro verde.

2.5.2.4 Gripper neumático

El gripper es el encargado de tomar o soltar las piezas que se desean trasladar. En la figura 2-61 se aprecia dentro del rectángulo rojo.

2.5.2.5 Motor eléctrico con caja reductora

Para el desplazamiento en los ejes X y Z se utilizan dos motores eléctricos de corriente continua de 24V. Cada motor cuenta con una caja reductora, y además con un encoder en el otro extremo.

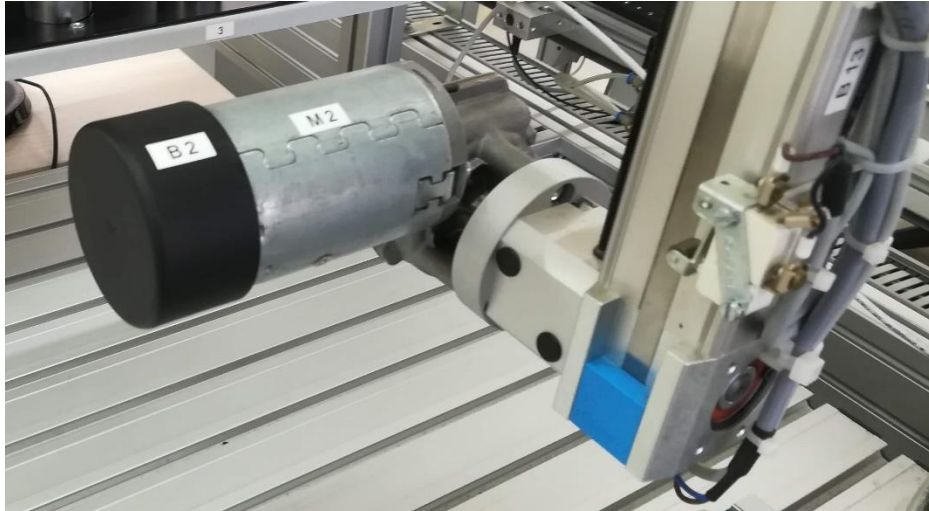


Figura 2-62. Motor eléctrico 24V

2.5.2.6 Encoder

Cada motor cuenta con un encoder acoplado, este elemento se utiliza como feedback y de esta manera conocer en todo momento el posicionamiento del brazo. Este elemento se conecta al PLC y se puede tener una lectura utilizando un contador rápido.



Figura 2-63. Encoder

2.5.2.7 Sensor Reed switch para posicionamiento

Con el fin de establecer los límites de movimientos en cada eje se utilizan sensores del tipo magnético en cada extremo, estos se utilizan solo como realimentación y se pueden utilizar dentro de la programación ya que se encuentran conectados a las entradas del PLC. En la figura 2-61 se pueden observar con TAG B14, B15 y B16.

2.5.2.8 Límites de carrera

También llamados limit switch son elementos electromecánicos que tienen como función detener el funcionamiento de los motores eléctricos cuando se ha llegado al final del recorrido permitido. La estación AS-RS cuenta con cuatro de estos elementos, dos para el eje X y dos para el eje Z.

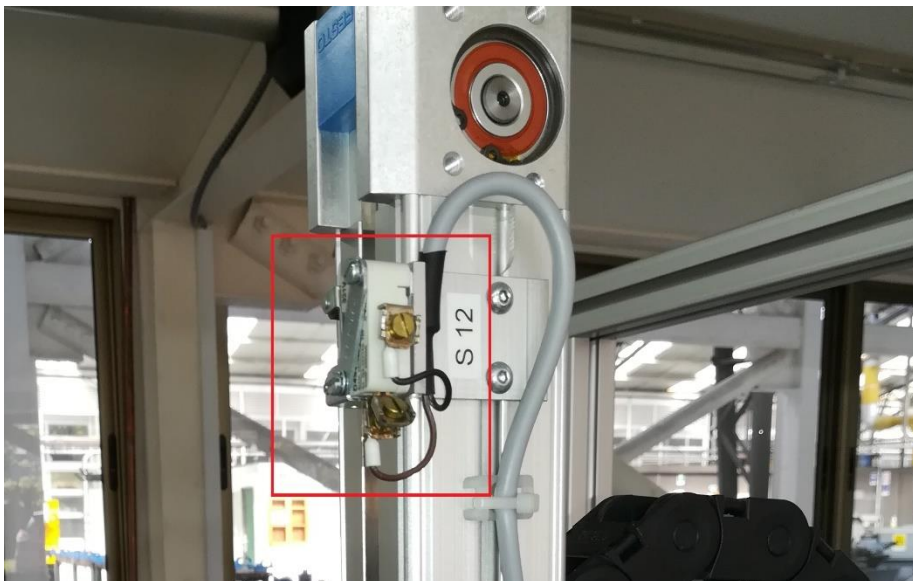


Figura 2-64. Límite de carrera

2.5.2.9 Tarjetas de control de motores DC RS Alpha 5

Cada motor eléctrico es controlado por un driver DC RS Alpha 5, esta tarjeta trabaja con una alimentación de 24VDC y tiene las siguientes características:

- Entrada para habilitación de movimiento
- Entrada para seleccionar el sentido de giro
- Entrada para cambiar la velocidad de giro del motor
- Salida para un motor DC



Figura 2-65. Driver de motores

2.5.2.10 Llave de seguridad

La llave de seguridad es un interruptor que permite la habilitación del poder hacia los motores eléctricos.



Figura 2-66. Llave de seguridad

2.5.2.11 Panel de control

Es un panel que cuenta con tres pulsadores (dos de ellos con led incorporado), además cuenta con un selector para el modo automático o manual. Adicionalmente posee dos luces para indicación. Todos estos elementos se encuentran conectados directamente a las entradas y salidas del PLC.

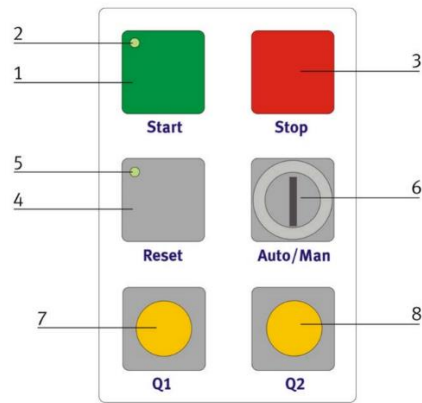


Figura 2-67. Panel de control AS-RS

2.5.2.12 Interfaz SYS Link

Se utiliza para comunicar la estación de almacenamiento con los los módulos esclavos AS-i de la estación de transporte.



Figura 2-68. Interfaz SYS Link

2.5.2.13 Relés de seguridad

Estos dispositivos de seguridad se encargan de permitir la alimentación de los motores, se encuentran conectados directamente a ellos y a su vez a los finales de carrera.



Figura 2-69. Relés de seguridad

2.5.2.14 Interfaz de conexiones encoder

Este elemento se utiliza para alimentar los encoder con 24 VDC y también conectar los encoder a las entradas digitales del PLC (XI 1 de la figura 2-70).

La interfaz XMA1 se utiliza como intermediario entre el controlador y los sensores y actuadores, se alimenta con 24 VDC.



Figura 2-70. Interfaz de conexiones

2.5.2.15 PLC 314C

Esta estación utiliza un PLC Siemens S7-300 CPU 314C PN/DP, además cuenta con una fuente de poder de 220VAC/24VDC y un módulo de simulación 274 2XH01-0AA0.



Figura 2-71. PLC estación AS-RS

2.5.3 Lógica de operación

La lógica de operación de la estación de almacenamiento está determinada por la figura 2-72, o bien por el procedimiento que se detalla a continuación:

- A. Para comenzar el proceso de almacenamiento debe haber disponibilidad en el Rack, de no ser así el sistema queda en espera hasta que se libera un espacio.
- B. Si hay disponibilidad en el almacén se procede a almacenar el producto.
- C. Una vez terminado el paso B, el sistema determina si existe demanda, si es así se toma la mercancía solicitada y se despacha en el carro disponible en la estación de transporte. De no existir solicitud de piezas el brazo vuelve a su posición inicial y se mantiene en espera hasta que llegue un nuevo vehículo

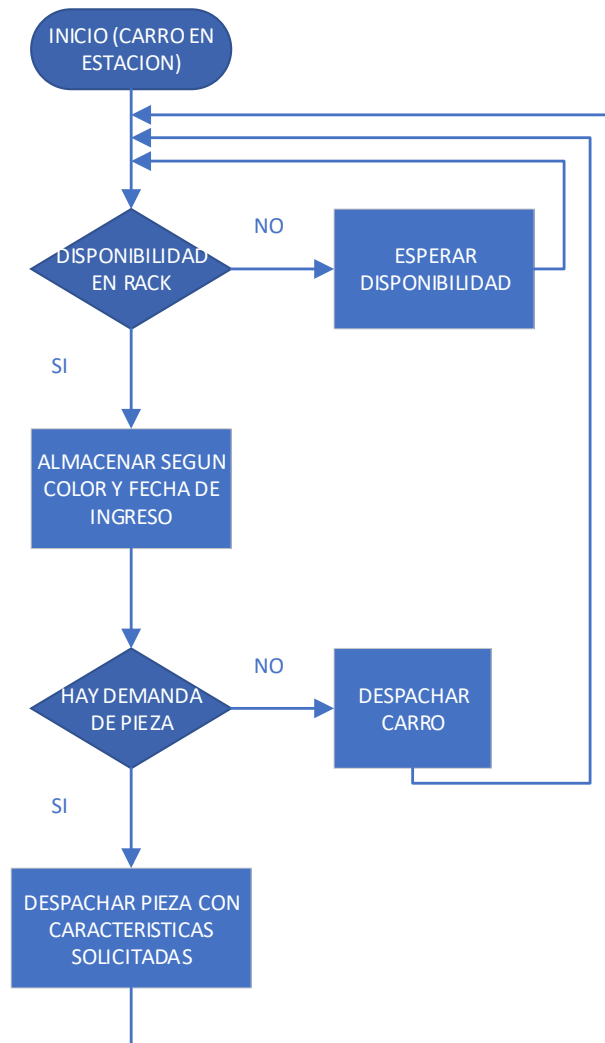


Figura 2-72. Lógica de operación de almacén

2.5.4 Levantamiento

Antes de comenzar con la programación es importante conocer los sensores y actuadores involucrados como también el direccionamiento que tiene cada uno de ellos. En la tabla 2-6 se detalla la información asociada a las entradas, mientras que la tabla 2-6 detalla las salidas.

Tabla 2-6. Entradas PLC estación AS-RS

Estación de Almacenamiento			
ENTRADA / SALIDA	TIPO DE SENSOR / ACTUADOR	COMENTARIO	DIRECCION PLC
Entrada	Encoder	ENCODER_X_1	%I136.0
Entrada	Encoder	ENCODER_X_2	%I136.1
Entrada	Encoder	ENCODER_Z_1	%I136.3
Entrada	Encoder	ENCODER_Z_2	%I136.4
Entrada	Pulsador	START	%I137.0
Entrada	Pulsador	STOP	%I137.1
Entrada	Switch	AUTO/MAN	%I137.2
Entrada	Pulsador	RESET	%I137.3
Entrada	Salida AS-i	CARRO	%I137.4
Entrada	Salida AS-i	PIEZA_MALA	%I137.6
Entrada	Salida AS-i	PIEZA_BUENA	%I137.7
Entrada	ReedSwitch	SENSOR_SUP_Z	%I138.0
Entrada	ReedSwitch	SENSOR_INF_Z	%I138.1
Entrada	ReedSwitch	SENSOR_IZQ_X	%I138.2
Entrada	ReedSwitch	SENSOR_DER_X	%I138.3
Entrada	ReedSwitch	SENSOR_INT_Y	%I138.4
Entrada	ReedSwitch	SENSOR_EXT_Y	%I138.5
Entrada	ReedSwitch	SENSOR_GRIPPER_CERRADO	%I138.6
Entrada	ReedSwitch	SENSOR_GRIPPER_ABIERTO	%I138.7

Tabla 2-7. Salidas PLC de almacén

Estación de Almacenamiento			
ENTRADA / SALIDA	TIPO DE SENSOR / ACTUADOR	COMENTARIO	DIRECCION PLC
Salida	Motor	X_LEFT	%Q136.0
Salida	Motor	X_RIGHT	%Q136.1
Salida	Motor	Z_UP	%Q136.2
Salida	Motor	Z_DOWN	%Q136.3
Salida	Cilindro neumático	Y_FRONT	%Q136.4
Salida	Cilindro neumático	OPEN_GRIPPER	%Q136.5
Salida	Driver Motor	X_FAST	%Q136.6
Salida	Driver Motor	Z_FAST	%Q136.7
Salida	Led	LUZ_START	%Q137.0
Salida	Led	LUZ_RESET	%Q137.1
Salida	Led	LUZ_Q1	%Q137.2
Salida	Led	IUZ_Q2	%Q137.3
Salida	Entrada AS-i	DESPACHAR_CARRO	%Q137.4
Salida	No utilizada	SALIDA_2	%Q137.5
Salida	No utilizada	SALIDA_3	%Q137.6
Salida	No utilizada	SALIDA_4	%Q137.7

2.5.5 Programación en TIA Portal

Teniendo como referencia el direccionamiento de los periféricos, la secuencia de operación de la estación y los alcances de la programación, se puede desarrollar una metodología para la elaboración del código. Para comenzar en esta sección se asumirá que el proyecto en TIA Portal se ha creado y se han configurado el Software y Hardware correctamente (PLC utiliza la dirección IP 10.4.5.85).

2.5.5.1 Mapa de Almacén

Como el brazo de la estación de almacenamiento utiliza encoder para conocer la posición actual, se debe establecer un punto de referencia o bien una posición inicial y así generar un mapa de la estación. En este trabajo se ha fijado como posición de inicio el lugar exacto en que se encuentra la pieza en el carro de transporte (gráficamente se puede observar en la figura 2-73).

Cuando se ha establecido las coordenadas [0,0] se puede buscar manualmente la posición 1 (uno) en el almacén, que en este caso corresponde a 5.244 pulsos en el eje X y 738 pulsos en el eje Y. Existen dos formas para generar el mapa completo, uno es mover manualmente el brazo por cada una de las posiciones y documentar cada lectura, o bien calcular la cantidad de pulsos basándose en las propiedades geométricas de cada elemento. En este caso, se ha comprobado que la segunda opción es más eficiente y exacta, es por ello que a continuación se detalla el procedimiento.

A. Para poder calcular la cantidad de pulsos que existe entre cada posición del rack se ha utilizado el procedimiento que se encuentra en el manual proporcionado por FESTO (Wolfgang Eckart, 2002). El proceso cuenta con varios pasos, pero en esta ocasión solo fue necesario el primero, los datos que se deben considerar son los siguientes:

- Recorrido en los ejes por cada rotación del motor: 63 [mm/revolución]
- Cantidad de pulsos por cada rotación del motor: 30 [pulsos/revolución]
- Relación de la caja de engranajes: 55:1

B. Se debe aplicar la siguiente formula:

$$Al_v = \frac{s_v \times f \times i}{s}$$

Donde,

- S = Recorrido en los ejes por cada rotación del motor
 f = Cantidad de pulsos por cada rotación del motor
 i = Relación de la caja de engranajes
 Sv = Distancia en los ejes entre cada posición

C. En la figura 2-73 se indica la distancia que existe entre cada posición del almacén, en el eje X corresponde a 87,6 [mm] y en el eje Z 90 [mm]. Con estos valores se puede proceder al cálculo.

$$Al_{v(eje\ X)} = \frac{87,6 \times 30 \times 55}{63} = 2292 \text{ [pulsos]}$$

$$Al_{v(eje\ Z)} = \frac{90 \times 30 \times 55}{63} = 2357 \text{ [pulsos]}$$

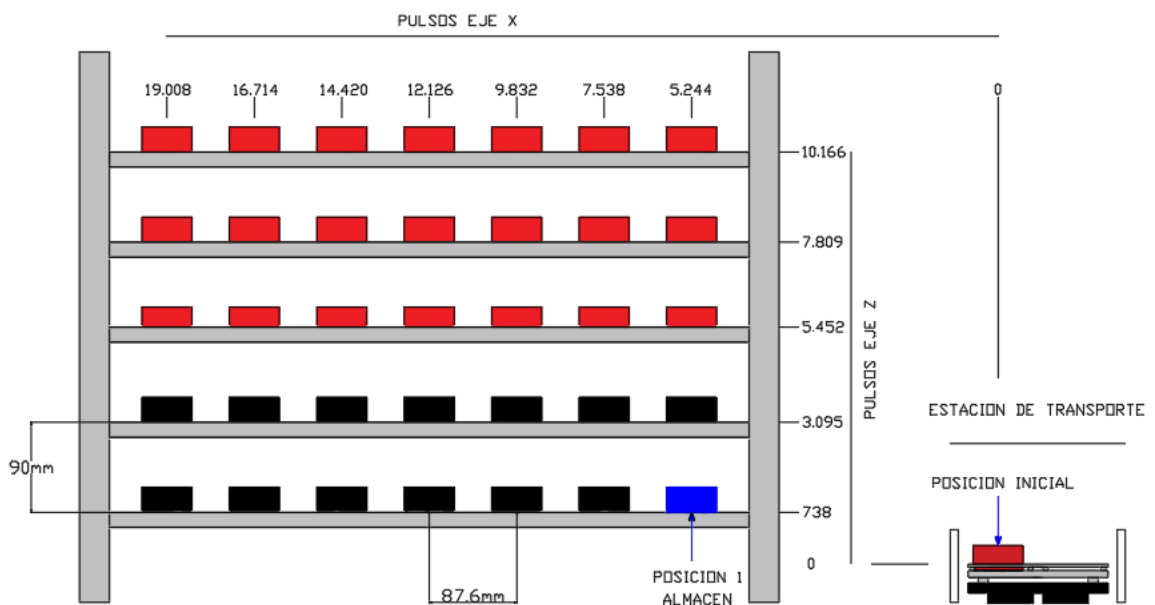


Figura 2-73. Mapa del almacén

- A. Con los datos obtenidos es posible completar la tabla 2-8 con coordenadas en ambos ejes, así mismo, en TIA Portal se deben utilizar dos variables del tipo "ARRAY" con una longitud de [0-35] para almacenar la posición de cada uno de los puntos en el rack (en la figura 2-74 se muestra un extracto del bloque de datos).

Tabla 2-8. Coordenadas de almacén

POSICION	COORDENADA X	COORDENADA Z
1	5244	738
2	7538	738
3	9832	738
4	12126	738
5	14420	738
6	16714	738
7	19008	738
8	5244	3095
9	7538	3095
10	9832	3095
11	12126	3095
12	14420	3095
13	16714	3095
14	19008	3095
15	5244	5452
16	7538	5452
17	9832	5452
18	12126	5452
19	14420	5452
20	16714	5452
21	19008	5452
22	5244	7809
23	7538	7809
24	9832	7809
25	12126	7809
26	14420	7809
27	16714	7809
28	19008	7809
29	5244	10166
30	7538	10166
31	9832	10166
32	12126	10166
33	14420	10166
34	16714	10166
35	19008	10166

Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Visible en ...	Valor de a...	Comentario
20	POSICION_X[17]	Int	34.0	9832	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	POSICION_X[18]	Int	36.0	12126	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	POSICION_X[19]	Int	38.0	14420	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	POSICION_X[20]	Int	40.0	16714	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	POSICION_X[21]	Int	42.0	19008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	POSICION_X[22]	Int	44.0	5244	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	POSICION_X[23]	Int	46.0	7538	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	POSICION_X[24]	Int	48.0	9832	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	POSICION_X[25]	Int	50.0	12126	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	POSICION_X[26]	Int	52.0	14420	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	POSICION_X[27]	Int	54.0	16714	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	POSICION_X[28]	Int	56.0	19008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	POSICION_X[29]	Int	58.0	5244	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	POSICION_X[30]	Int	60.0	7538	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	POSICION_X[31]	Int	62.0	9832	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	POSICION_X[32]	Int	64.0	12126	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	POSICION_X[33]	Int	66.0	14420	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	POSICION_X[34]	Int	68.0	16714	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	POSICION_X[35]	Int	70.0	19008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	POSICION_Z	Array[0..35] of Int	72.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	POSICION_Z[0]	Int	72.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	POSICION_Z[1]	Int	74.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	POSICION_Z[2]	Int	76.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	POSICION_Z[3]	Int	78.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
44	POSICION_Z[4]	Int	80.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
45	POSICION_Z[5]	Int	82.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
46	POSICION_Z[6]	Int	84.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
47	POSICION_Z[7]	Int	86.0	738	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
48	POSICION_Z[8]	Int	88.0	3095	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
49	POSICION_Z[9]	Int	90.0	3095	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
50	POSICION_Z[10]	Int	92.0	3095	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figura 2-74. Tabla de coordenadas en TIA Portal

2.5.5.2 Estrategia de programación

Con la finalidad de optimizar y facilitar la estructura de la programación se crearán subrutinas para cada operación. A continuación, se detallará cada uno de los subprocesos.

- Subrutina Encoder
 - Antes de comenzar la programación es importante considerar que toda subrutina que se desee ejecutar en la programación debe estar incluida en el bloque principal OB1, o bien, dentro de otra subrutina que se encuentre dentro de este.
 - Los encoder de los ejes X y Z se encuentran conectados en los canales 0 y 1 respectivamente en la CPU. En la figura 2-73 se indica la ruta que se debe seguir para realizar la configuración y los ajustes importantes que se deben considerar.

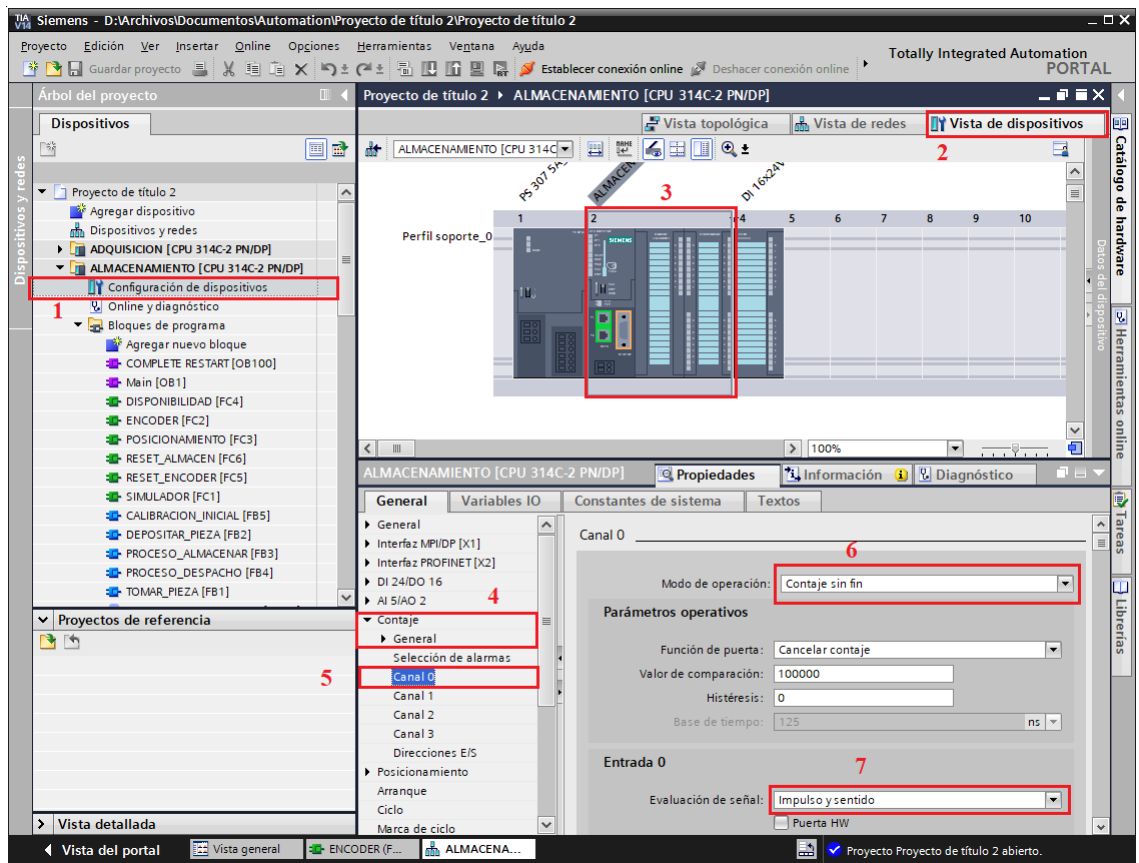


Figura 2-75. Configuración contadores

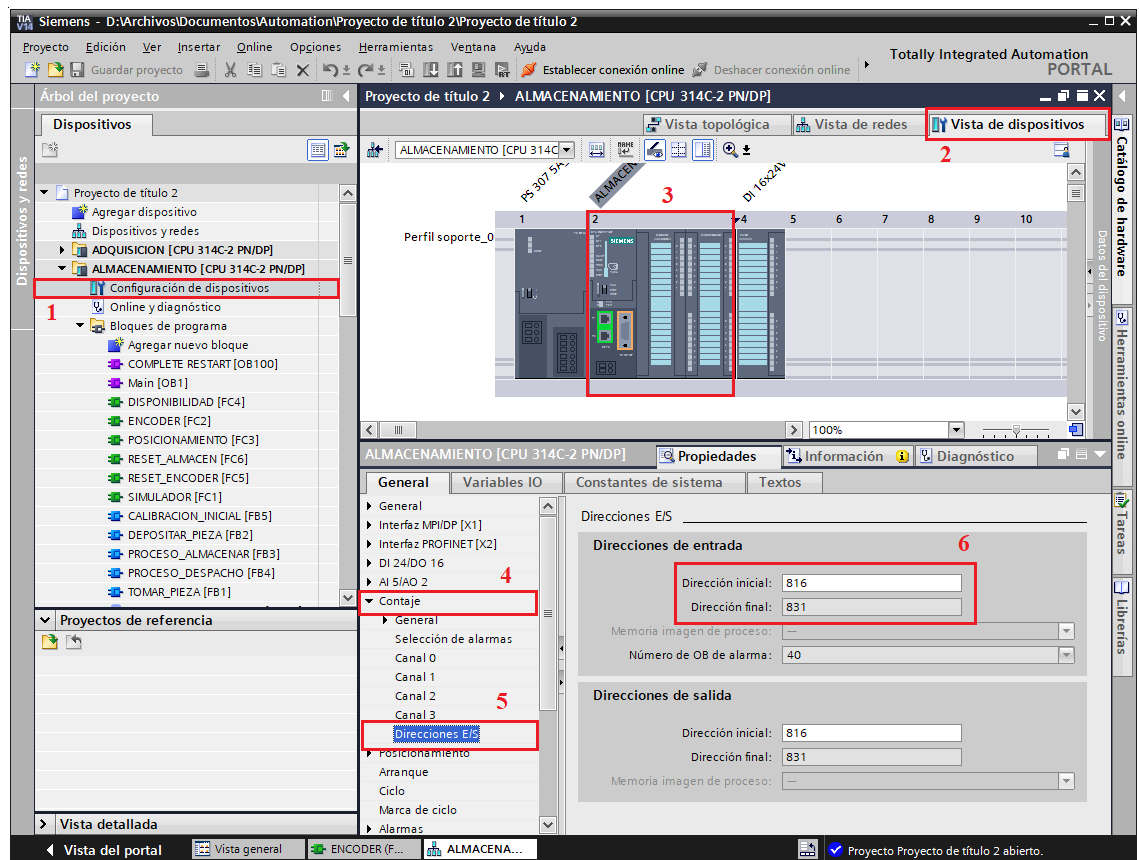


Figura 2-76. Dirección de contadores

- Luego es necesario buscar el direccionamiento en la memoria del PLC que tiene asignado cada encoder, para conocer este valor se debe seguir la ruta de la figura 2-74. Se puede apreciar que el primer contador tiene como dirección inicial el Byte 816, debido a que los datos se almacenan en una variable del tipo DInt la dirección del segundo contador comienza en el Byte 820. En la tabla 2-8 se resume las direcciones de cada canal.

Tabla 2-9. Dirección de contadores rápidos

Canal	Dirección Decimal	Dirección Hexadecimal
Canal 0	816	W#16#330
Canal 1	820	W#16#334
Canal 2	824	W#16#338
Canal 3	828	W#16#33C

- Teniendo la dirección de cada encoder se debe configurar el bloque de función especial para contadores rápidos. Este se puede encontrar en la ruta señalada de la figura 2-75.
- Para poder utilizar el contador rápido se debe configurar los siguientes parámetros:
 - LADDR: Aquí se indica la dirección la dirección en la memoria asignada al contador que se desea utilizar, la nomenclatura debe ser en formato hexadecimal.
 - CHANNEL: Se debe ingresar el número del canal utilizado, el valor debe ser en formato Decimal.
 - SW_GATE: Es la puerta de software para iniciar o detener el contador. Es importante que este terminal se encuentre activo para que el contador pueda trabajar.
 - COUNTVAL: Almacena el valor actual del conteo, el tipo de datos que se utiliza en este punto debe ser Dint.

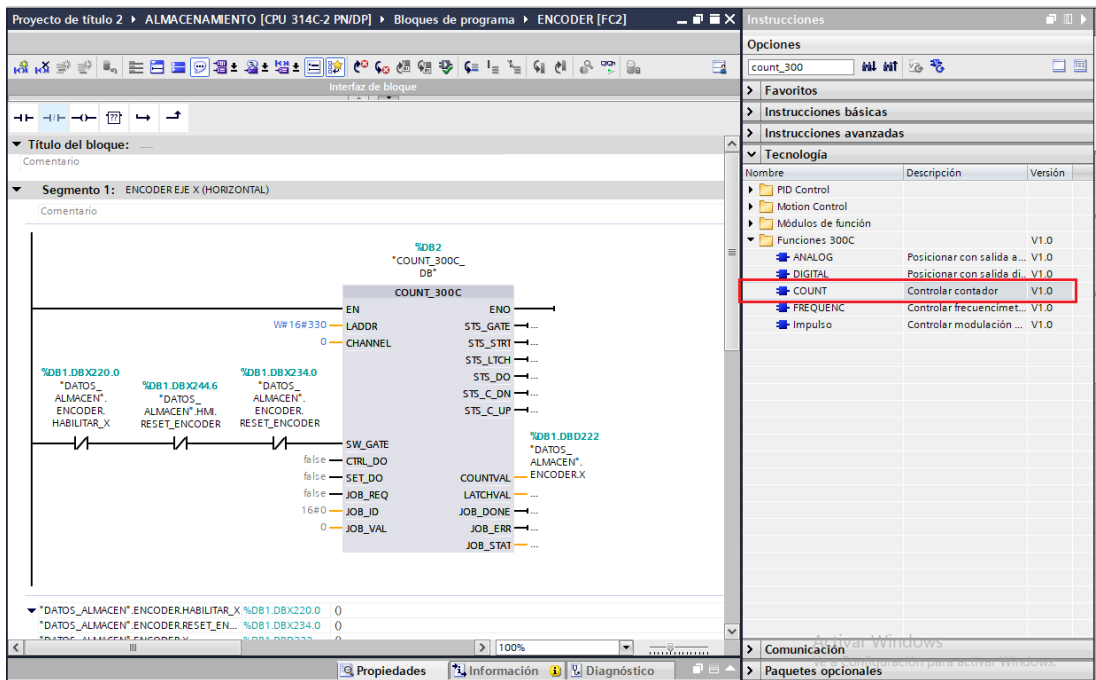


Figura 2-77. Contador rápido

A. Subrutina de Disponibilidad

Con el fin de analizar rápidamente todas las posiciones del almacén se ha realizado un programa en SCL utilizando ciclos FOR para analizar las matrices unidireccionales (ARRAY) que contienen las coordenadas de cada posición almacenadas en un bloque de datos. Cabe mencionar, que realizar este proceso con programación LADDER o S7-GRAPH es bastante más complejo, es por lo que se recomienda utilizar esta estructura o alguna similar

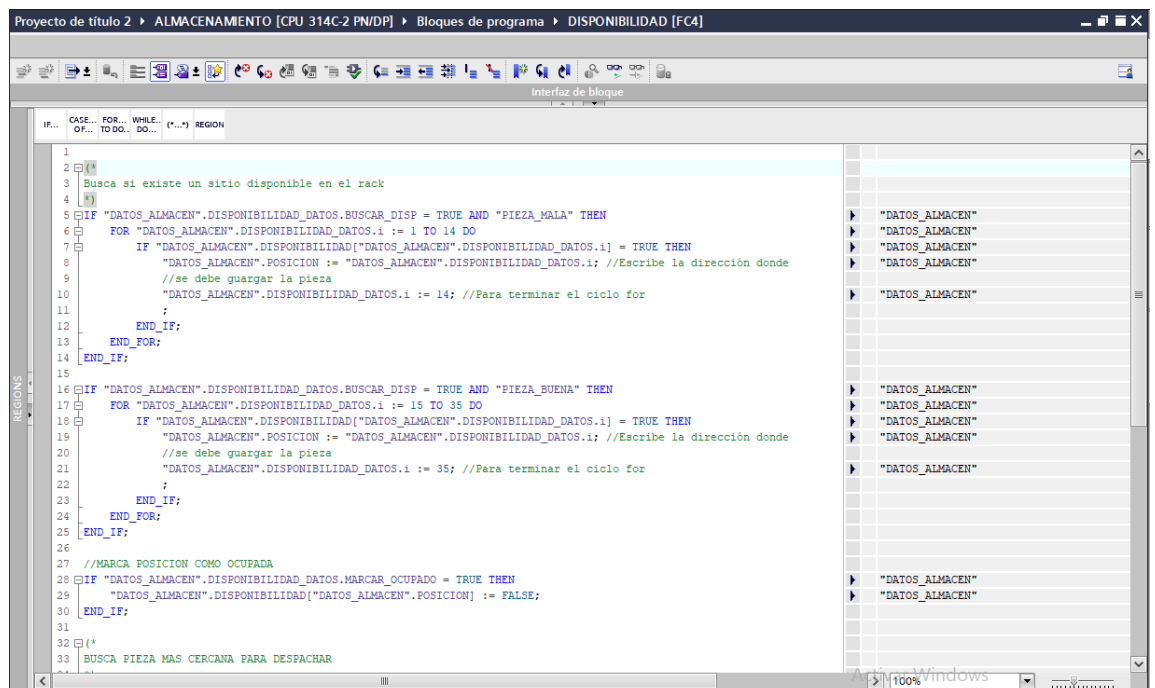
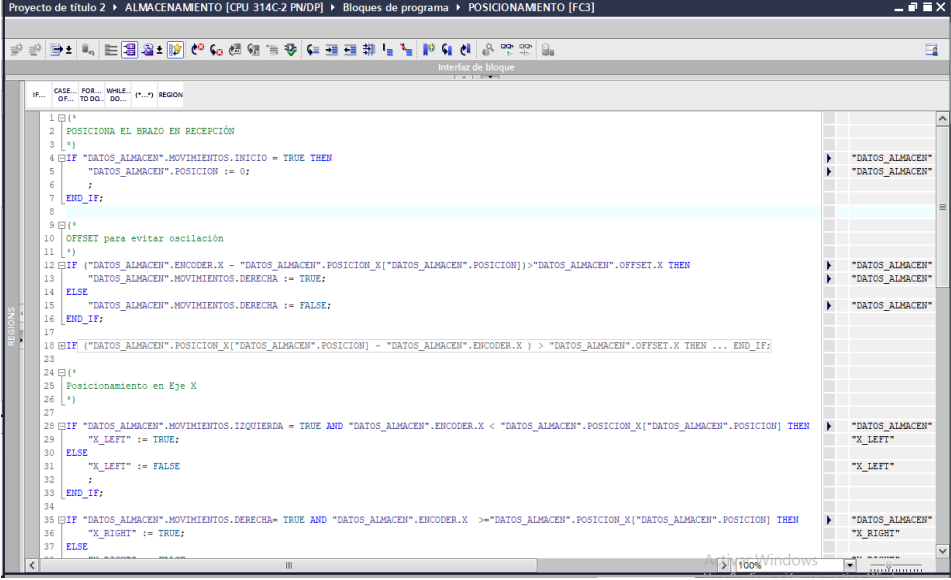


Figura 2-78. Subrutina de disponibilidad

B. Subrutina de Posicionamiento

Cuando se ha determinado la posición adecuada para almacenar la pieza entrante o donde se debe ir a buscar pieza saliente, el brazo debe desplazarse rápidamente a dicho lugar. Para poder realizar esta acción es necesario realizar un pequeño código en SCL, ya que de esta forma es posible utilizar una variable externa para apuntar a un valor dentro de un vector. En la figura 2-79 se observa un extracto de ejemplo.



```
Proyecto de título 2 > ALMACENAMIENTO [CPU 314C-2 PN/DP] > Bloques de programa > POSICIONAMIENTO [FC3]
Interfaz de bloque
REGIONE
1 IF... CASE... FOR... WHILE... (*-*) REGION
2 OF... TODOS... DO...
3
4 (*
5 POSICIONA EL BRAZO EN RECEPCIÓN
6 *)
7
8 IF "DATOS_ALMACEN".MOVIMIENTOS.INICIO = TRUE THEN
9 "DATOS_ALMACEN".POSICION := 0;
10
11 END_IF;
12
13 (*
14 OFFSET para evitar oscilación
15 *)
16
17 IF ("DATOS_ALMACEN".ENCODER.X - "DATOS_ALMACEN".POSICION_X("DATOS_ALMACEN".POSICION)) > "DATOS_ALMACEN".OFFSET.X THEN
18 "DATOS_ALMACEN".MOVIMIENTOS.DERECHA := TRUE;
19
20 ELSE
21 "DATOS_ALMACEN".MOVIMIENTOS.DERECHA := FALSE;
22
23 END_IF;
24
25 (*
26 POSICIONAMIENTO en Eje X
27 *)
28
29 IF "DATOS_ALMACEN".MOVIMIENTOS.ISQUIERDA = TRUE AND "DATOS_ALMACEN".ENCODER.X < "DATOS_ALMACEN".POSICION_X("DATOS_ALMACEN".POSICION) THEN
30 "X_LEFT" := TRUE;
31
32 ELSE
33 "X_LEFT" := FALSE;
34
35 END_IF;
36
37 IF "DATOS_ALMACEN".MOVIMIENTOS.DERECHA = TRUE AND "DATOS_ALMACEN".ENCODER.X >= "DATOS_ALMACEN".POSICION_X("DATOS_ALMACEN".POSICION) THEN
38 "X_RIGHT" := TRUE;
39
40 ELSE
41 "X_RIGHT" := FALSE;
42
43 END_IF;
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Figura 2-79. Subrutina de posicionamiento

C. Subrutina de almacenamiento y/o despacho

El proceso de almacenamiento o despacho se realiza de forma secuencial, es por ello que estas subrutinas solamente se utilizan para llamar a los movimientos individuales previamente programados. De esta forma es más sencillo encontrar y solucionar posibles problemas futuros. En la figura 2-80 se observa un extracto del proceso de almacenamiento, donde se puede apreciar lo simple que es la secuencia.

D. Calibración Inicial

Como función especial se ha agregado una calibración o referenciación automática del brazo. El proceso es simple y utiliza la siguiente lógica:

- Al arrancar el PLC por primera vez no se conoce la posición física del brazo es por esto que se realiza un movimiento de ocho segundos hacia la izquierda (considerando que en el peor de los casos el brazo estará en el tope de la derecha), de esta manera nos aseguramos de que el sensor (reedswitch posición inicial) que indica que la posición inicial está a la derecha.

- Luego se utiliza la misma lógica en el eje Z, es decir, se mueve cuatro segundos hacia arriba.
- Finalmente se realizan movimientos hacia la derecha y abajo hasta que se activen sus respectivos sensores. Cuando los sensores se han activado, se resetean los contadores y establece ese punto como coordenadas [0,0].

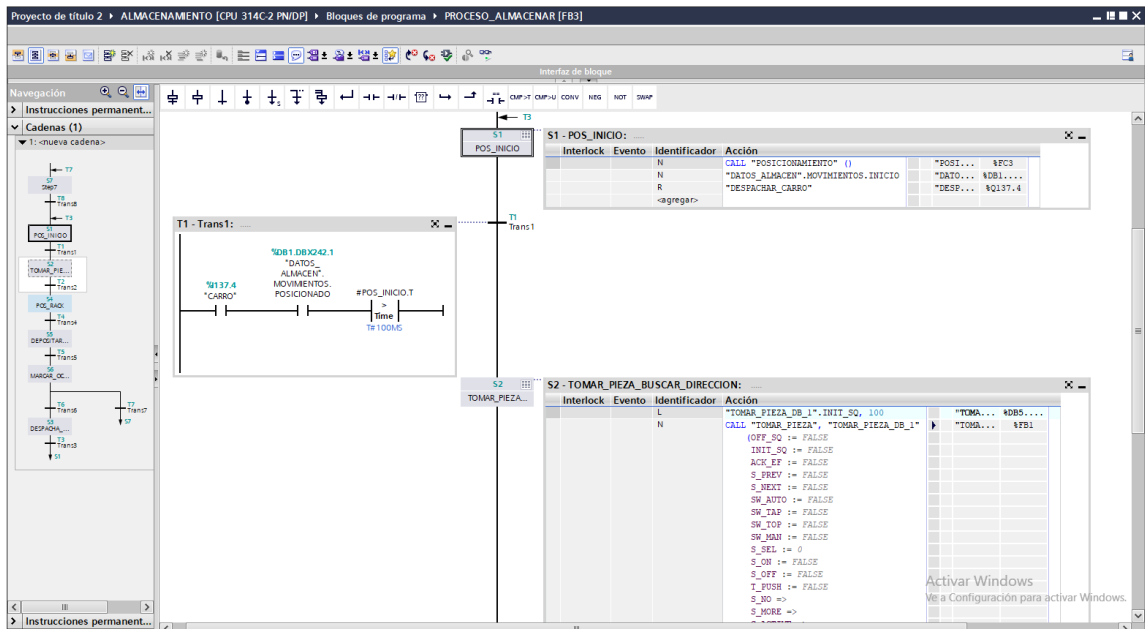


Figura 2-80. Subrutina de almacenamiento

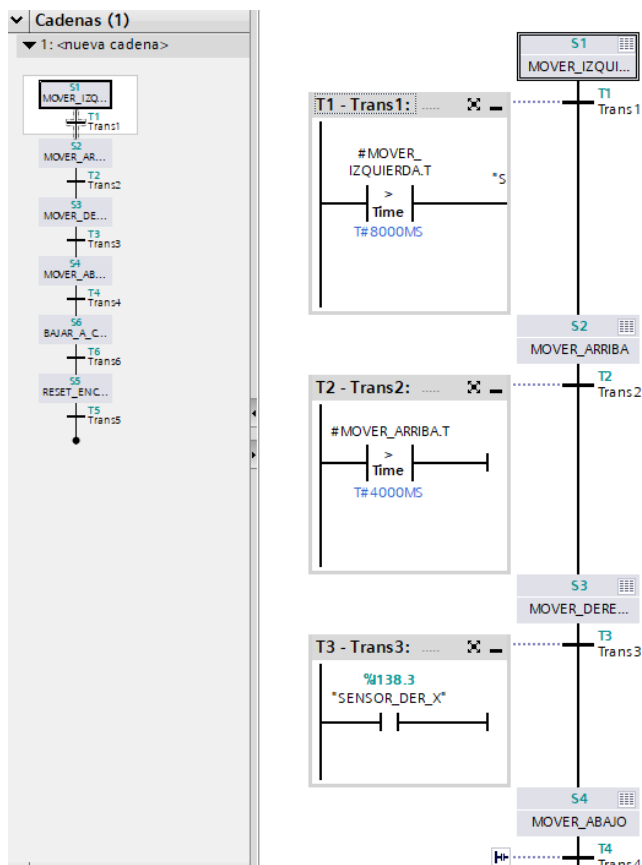


Figura 2-81. Subrutina de calibración inicial

**CAPITULO 3: ROBOTS INDUSTRIALES Y SU APLICACIÓN EN CELDAS
FLEXIBLES DE MANUFACTURA**

3.1 ROBOTS INDUSTRIALES

Un robot industrial "se puede definir como el conjunto de elementos electromecánicos que propician el movimiento de un elemento terminal (gripper o herramienta), sea para cumplir una función o solo para manipular un objeto" (M. Alonzo, y otros, 2014).

Existen distintos tipos de clasificaciones de los robots, en esta memoria se hablará específicamente del brazo robótico antropomórfico. En este caso se puede observar que su forma de construcción es similar a la de un brazo humano, replicando el movimiento de las articulaciones como también la disposición de cada elemento. Por otro lado, se puede decir que "Un brazo robótico está compuesto por una estructura mecánica, transmisores, actuadores, sensores, elementos finales y un controlador" (M. Alonzo, y otros, 2014). En la figura 3-1 se puede observar la analogía recientemente señalada.

El desarrollo de la robótica ha sido fundamental en algunas industrias para poder producir masivamente ciertos productos, principalmente se pueden ver de forma masificada en la industria automotriz, electromecánica y electrónica potenciada mayormente en el proceso de ensamble, pintura, soldadura, etc.

La robótica si bien es una materia reciente y muchas veces desconocida técnicamente, su implementación asociada múltiples beneficios y se espera que su crecimiento aumente considerablemente con el paso de los años.

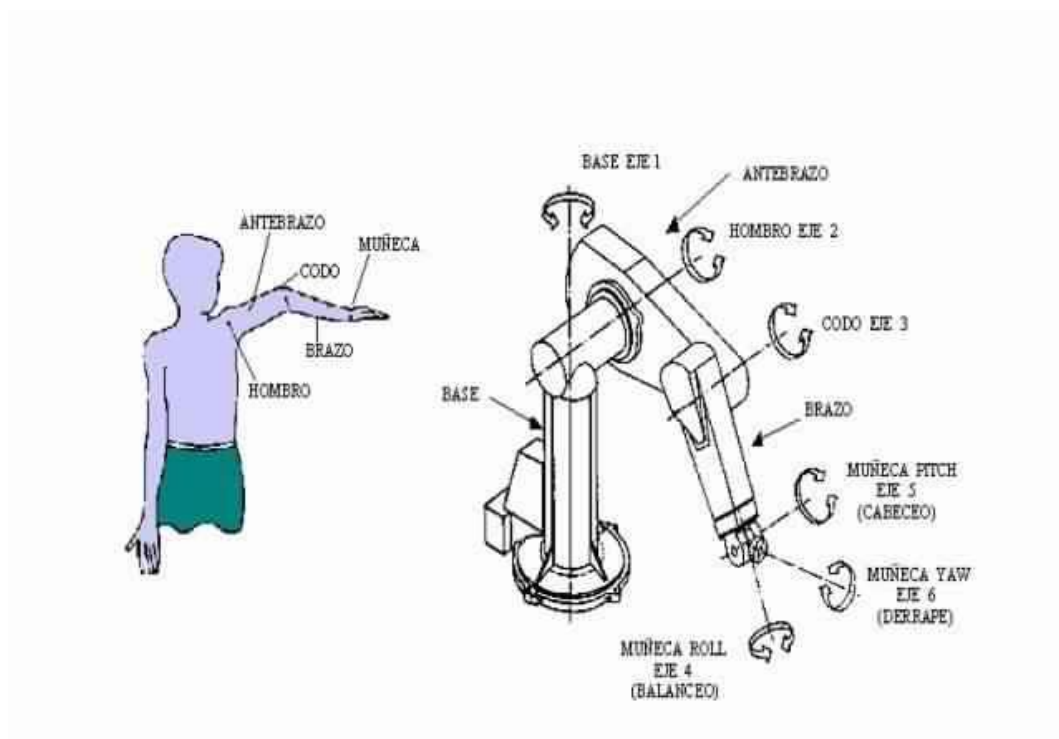


Figura 3-1. Analogía brazo robótico - brazo humano



Figura 3-2. Robots en la industria automotriz

3.2 **BRAZO ROBÓTICO MITSUBISHI RV-2SDB**

La celda flexible de manufactura de la USM Concepción cuenta con un brazo robótico de la marca MITSUBISHI modelo RV-2SDB. Básicamente, debido a su posición de instalación se utiliza para el movimiento de materias primas hacia la CNC y posteriormente traslado hacia la línea de transporte y almacenamiento.

Este robot es del tipo estacionario, está diseñado para operar con hasta 2kg de carga y posee seis grados de libertad, logrando con esto una precisión de $\pm 0,2$ [mm], adicionalmente posee un gripper neumático como herramienta para tomar o depositar piezas (en la figura 3-4 se pueden apreciar gráficamente las articulaciones).



Figura 3-3. Brazo robótico

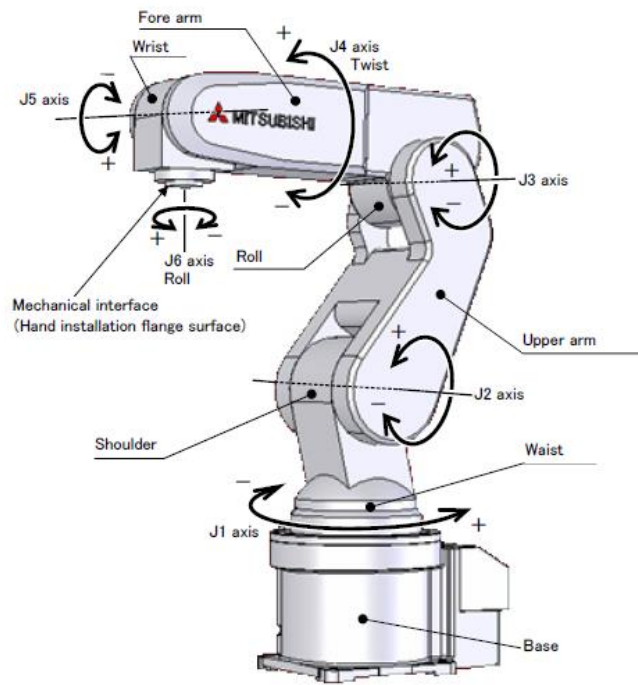


Figura 3-4. Grados de libertad brazo robótico

Tabla 3-1. Grados de libertad brazo robótico

EJE	Movimiento total (grados)	Rango movimiento (grados)
J1	480	-240 / +240
J2	240	-120 / +120
J3	160	0 / +160
J4	400	-200 / +200
J5	240	-120 / +120
J6	720	-360 / +360

3.2.1 Unidad procesadora

El brazo robótico es controlado por la unidad CR1DA-711, esta unidad se encarga de controlar todos los sensores y actuadores involucrados en el control y movimiento del robot. En la tabla 3-2 se describe la función de cada elemento que compone este controlador.

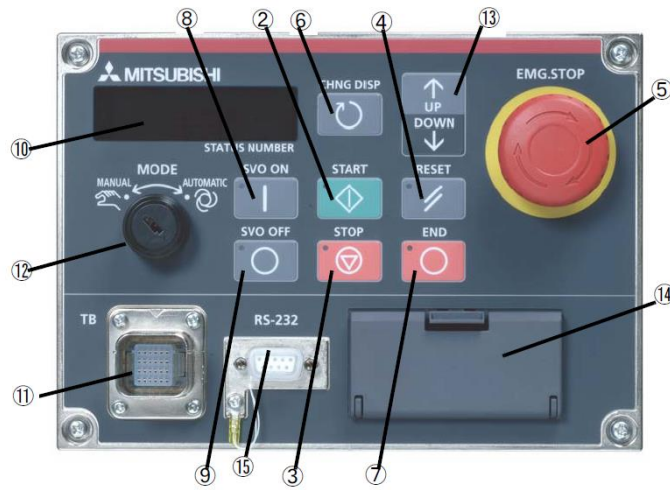


Figura 3-5. Controlador CR1DA-711

Tabla 3-2. Componentes de controlador CR1DA-711

NÚMERO	NOMBRE	DESCRIPCIÓN
1	Botón ON/OFF	Permite encender o apagar la unidad procesadora.
2	Botón de START	Ejecuta el programa seleccionado en el robot
3	Botón de STOP	Detiene inmediatamente el robot, el servo no se detiene.
4	Botón de RESET	Este botón borra los errores.
5	Interruptor de PARADA DE EMERGENCIA	Este interruptor detiene totalmente el robot en caso de emergencia.
6	Botón CHNGDIP	Este botón cambia la pantalla en el panel en el orden: "override" => "line number" => "Program No." => "user information" => "manufacturer information".
7	Botón END	Esto detiene que el programa sea ejecutado en la última línea o al END del estado.
8	Botón SVO.ON	Enciende el servo.
9	Botón SVO.OFF	Apaga el servo.
10	STATUS.NUMBER (panel de visualización)	Muestra el N° de alarma, N° de programa, override value %, etc.
11	Conector T/B	Conector especial para T/B.
12	Interruptor MODE	Permite cambiar el modo de operación, automático o manual.
13	Botón UP/DOWN	Permite deslizar hacia arriba o abajo los detalles que aparecen en el "STATUS.NUMBER" mostrados en el panel.
14	Cubierta interfaz	Cubierta para interfaz USB y montaje de batería.
15	Interfaz RS-232	Puerto para comunicación RS-232.

3.2.2 Teaching Pendant

Esta unidad modelo R32TB se utiliza para modificar parámetros del sistema, para referenciar el brazo robótico, simular movimientos, modificar y crear programas de control. Si bien realizar programaciones desde esta unidad es bastante más complejo en comparación al software CIROS STUDIO, en algunas situaciones es mucho más útil y práctica debido a su capacidad de desplazamiento. En la tabla 3-2 se describe la función de cada tecla.

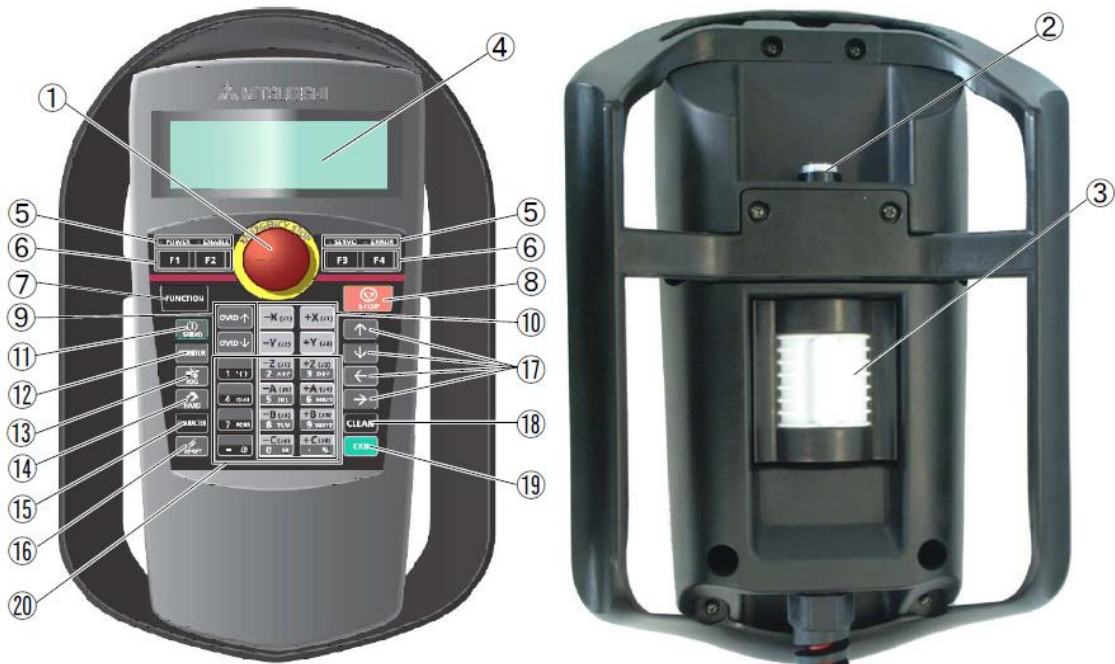


Figura 3-6. Teaching pendant

Tabla 3-2. Descripción de teclas teaching pendant

NÚMERO	NOMBRE	DESCRIPCIÓN
1	Interruptor EMERGENCY STOP	Detiene el servo del robot y detiene inmediatamente la operación.
2	Interruptor ENABLE / DISABLE	Habilita o deshabilita las operaciones del robot con el mando local.
3	DEADMAN SWITCH	Este pulsador debe estar accionado para la conexión del servo accionamiento, de lo contrario se apaga automáticamente.
4	Panel de visualización LCD	Muestra el estado del robot y varios menús de ajuste.
5	Lámpara indicadora de estado	Muestra el estado del robot o del T/B.
6	F1, F2, F3 y F4	Permite seleccionar la función correspondiente en relación con lo mostrado en el display.
7	Botón FUNCTION	Cambia la función mostrada en la parte inferior del display.

8	Botón STOP	Interrumpe la ejecución del programa en el robot, pero el servo no se detiene.
9	Teclas OVRD	Permiten la variación de la velocidad máxima con la que se puede mover el brazo robótico.
10	Teclas para el modo JOG	Estas teclas permiten mover de forma independiente cada una de las articulaciones del robot en modo JOG, o bien, mediante los números se pueden ingresar valores en los menús de ajuste y programación.
11	Botón SERVO	Permite accionar el servo desde el T/B, para que esto se concrete es necesario mantener presionado en todo momento el interruptor DEADMAN.
12	Botón MONITOR	Cambia al modo monitor y muestra el menú monitor.
13	Tecla JOG	Cambia al modo JOG y muestra el menú JOG
14	Tecla HAND	Cambia al modo HAND y muestra el menú Hand.
15	Tecla CHARACTER	Llama al menú edición y alterna entre números y letras.
16	Tecla RESET	Reposición de un código de error
17	Teclas de navegación	Mueve el cursor en la dirección correspondiente.
18	Tecla CLEAR	Borra el carácter situado en la posición del cursor.
19	Tecla EXE	Introducción de datos o movimiento del robot en modo directo.
20	Tecla de carácter	Sobrescribe el carácter situado en la posición del cursor.

3.3 SOFTWARE CIROS STUDIO

CIROS STUDIO es un entorno de desarrollo orientado a la automatización y robótica, se basa en la virtualización de procesos como medio de aprendizaje. Los principales ámbitos de aplicación son (FESTO, 2019):

- **Programación de robot:** Una de las propiedades importantes es que posee una plataforma de programación de robots. Para ello el programa posee un programa didáctico integrado y una amplia gama de robots.
- **Formación en programación de PLC:** La plataforma además posee más de 30 modelos de procesos de estaciones preconfigurados. Los que pueden ser controlados directamente con el PLC S/ virtual integrado, desde SIMATIC S/-PLCSIM o a través de un PLC físico a través de EasyPort.
- **Formación en planificación y control de la producción:** Permite conectar la simulación con la unidad de control de nivel superior de equipos reales. Esta aplicación se centra en la planificación de plantas de producción, el diseño y optimización de sistemas MES (manufacturing executing system) y la gestión de la producción (FESTO, 2019).
- **Formación en localización y reparación de fallas:** Permite la simulación de fallas de diversos procesos, incluyendo falla de sensores.

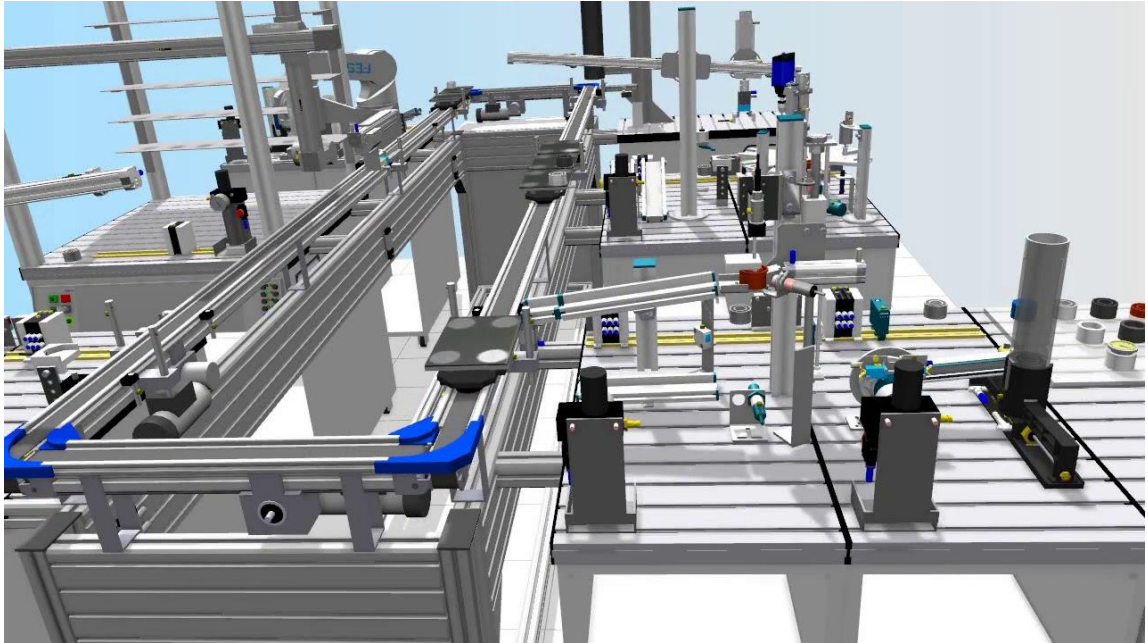


Figura 3-7. Software CIROS

3.4 **PROGRAMACIÓN DE BRAZO ROBÓTICO**

La estructura del programa para esta aplicación es simple, se ha condicionado al brazo robótico a buscar una pieza cuando la CNC haya terminado el proceso de maquinado y la deposite en el contenedor de entrada de la estación de adquisición. Mientras no haya piezas terminadas el robot hará tareas anexas, es decir, simulará hacer otro proceso, en este caso en moverá otras piezas. En la figura 3-8 se puede apreciar el diagrama de flujo del proceso.

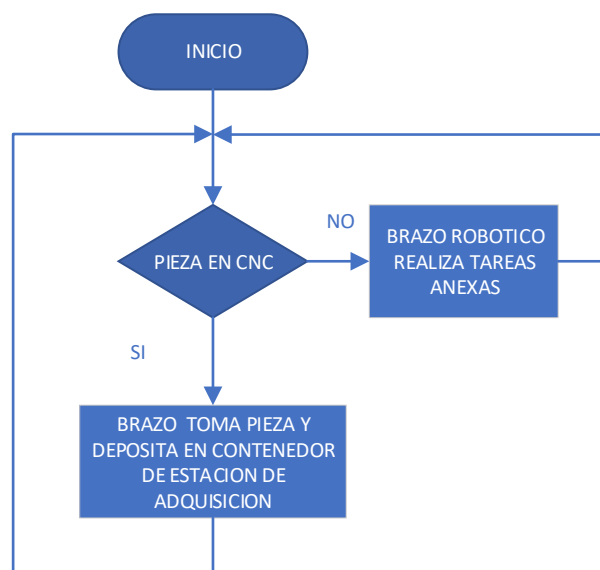


Figura 3-8. Diagrama de flujo de brazo robótico

3.4.1 Creación de un proyecto

El primer paso que debe realizar es la creación de un nuevo proyecto. Para eso se debe abrir la plataforma CIROS STUDIO con la llave HASP conectada en un puerto USB, de lo contrario el software no cargará correctamente. Con el software abierto se debe seguir la ruta indicada en la figura 3-9. Luego se desplegará una ventana con tres pasos, el primero está destinado a la documentación del proyecto, como nombre, descripción, entre otros.

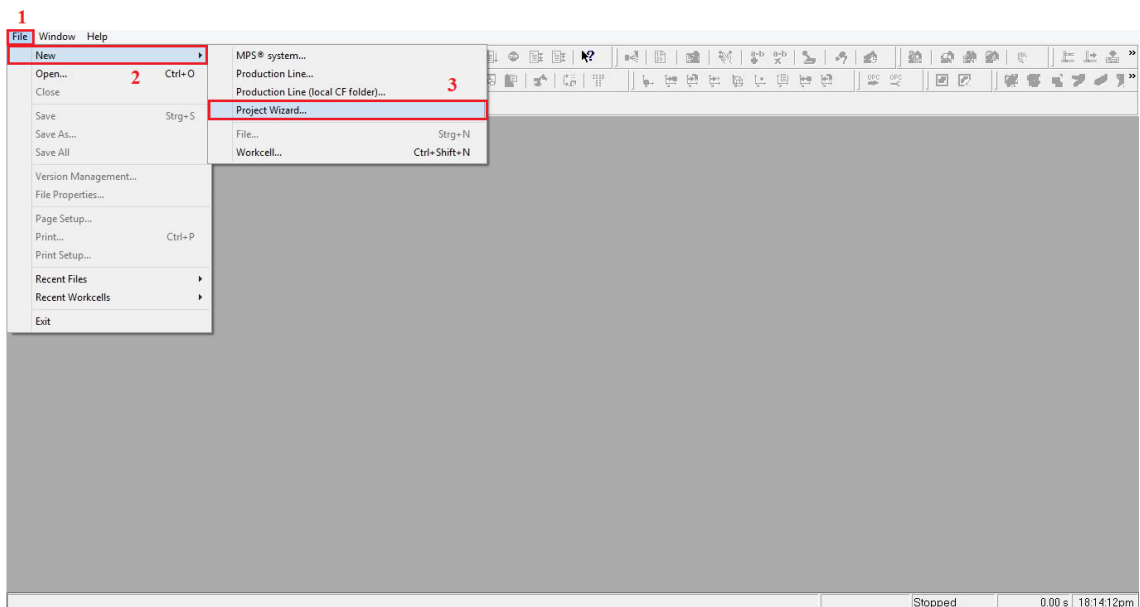


Figura 3-9. Nuevo proyecto

La segunda ventana está orientada a la selección del robot que se utilizará, se debe buscar en el listado "Robot Type", en este caso corresponde al RV-2SDB. Seleccionar lenguaje de programación MELFA BASIC V y luego presionar "Finish".

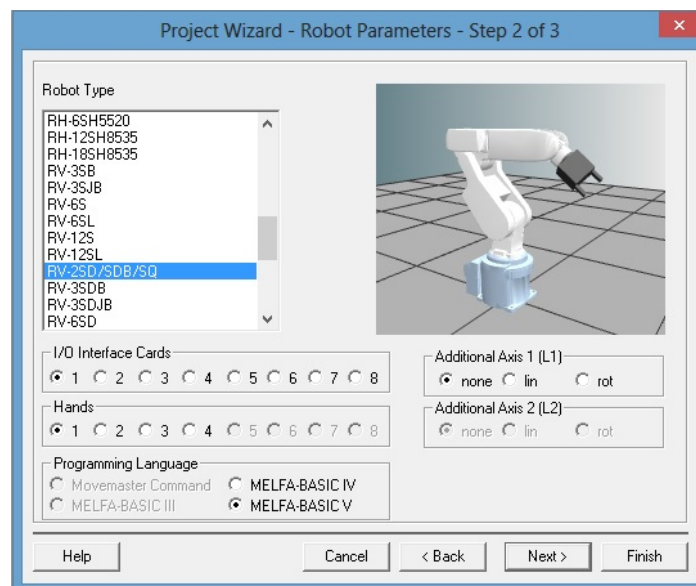


Figura 3-10. Modelo de robot

3.4.2 Interfaz de programación

La interfaz de programación se compone de cuatro ventanas principales:

- La primera es la ventana de mensajes, donde aparecerán los posibles problemas de compilación, carga de programas, etc.
- La segunda corresponde a la ventana de programación, es aquí donde se debe ingresar el código de programa.
- La tercera ventana es la de posiciones, es aquí donde se guardan las coordenadas de los distintos puntos de referencia que utilizará el brazo para realizar sus movimientos
- La cuarta ventana es la interfaz gráfica, muestra los movimientos en línea del robot.

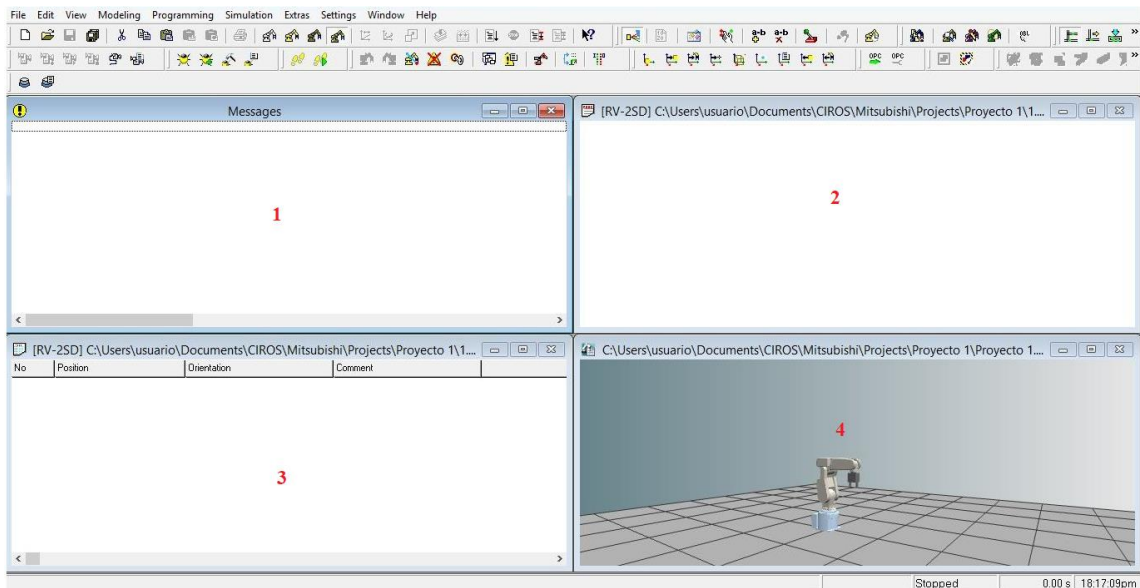


Figura 3-11. Interfaz de programación

3.4.3 Programación y resultados

Debido a la falta de conocimientos en el manejo de la estación CNC, se utilizó un sensor del tipo laser perteneciente a la estación del brazo robótico para simular que se ha terminado el proceso de fabricación de una pieza, por ende, el robot debe trasladarla hacia la estación de adquisición. Se ha utilizado el lenguaje de programación MELFA BASIC V y se ha consultado a la referencia incorporada para conocer la nomenclatura de los ciclos de trabajo. A continuación, en la figura 3-11 se muestra un extracto de la programación.

```
[RV-2SD] C:\USERS\CELDA1\DOCUMENTS\CIROS\MITSUBISHI\PROJECTS\CMH2\1.MB5
Def Io S1 = Bit,28
HOPEN 1
If S1=1 Then
Ovrd 50
Mov P1
Mov P3
Ovrd 30
Mov P6
Ovrd 5
Mov P7
Mov P8
Dly 1
HClose 1
Dly 1
Mov P9
Ovrd 60
Mov P10
```

Figura 3-12. Extracto de programación MELFA BASIC

Una vez terminado el programa secuencial, es necesario asignar las coordenadas correspondientes a cada punto programado. Esto se puede hacer desde el "Teaching Pendant" o bien desde el software. Para esto se debe presionar F8, tras lo cual se desplegará la ventana de la figura 3-13 que se utiliza para posicionar el brazo de forma manual y también para asignar coordenadas a puntos nuevos o guardados.

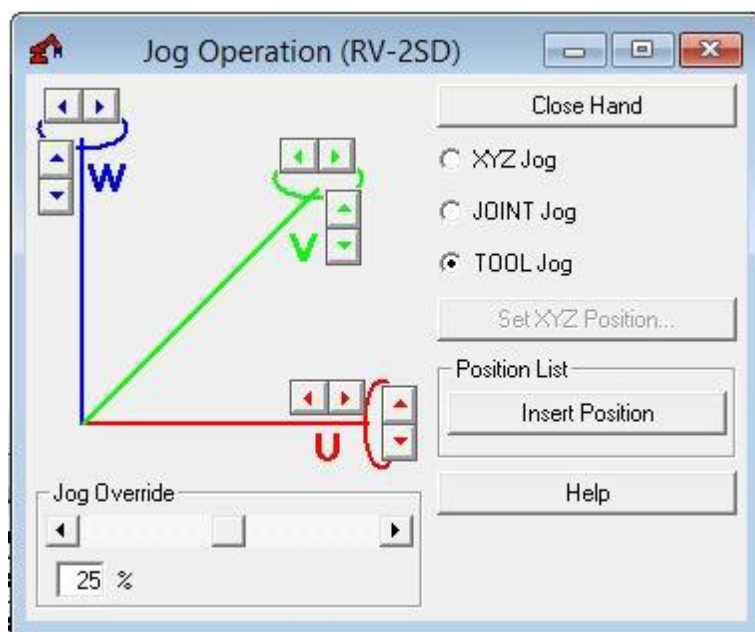


Figura 3-13. Referenciación de puntos

CAPITULO 4: CONCLUSIONES Y REFERENCIAS

4.1 RESULTADOS

Tras realizar la programación independiente de cada una de las estaciones de la celda flexible de manufactura que se propuso controlar, se realizó un trabajo sincronizado obteniendo los siguientes resultados:

- Si bien el brazo robótico debía tomar piezas desde la CNC, se utilizó un sensor de la misma estación para simular el término de fabricación de una pieza, cuyo comportamiento es similar al proceso real, pero con una localización distinta.
- Las estaciones de adquisición y distribución trabajaron de la forma esperada, es decir, de forma sincronizadas entre sí, y comandadas desde la estación de transporte.
- La estación de transporte trabajó correctamente, siendo adicionalmente el controlador central, encargado de sincronizar los distintos procesos.
- La estación de visión o inspección trabajó correctamente la mayor parte del tiempo, sin embargo, en momentos cuando la luz solar varía rápidamente se producían errores aleatorios. Es por esto que se recomienda utilizar esta estación bajo condiciones de iluminación estables.
- Finalmente, el almacén se desempeñó de buena manera, el sistema pudo realizar un almacenamiento y despacho selectivo desde y hacia los carros respectivamente. Sin embargo, el despacho no se pudo completar, ya que no se cuenta con una estación de despacho después del almacén.

4.2 CONCLUSIONES

Como conclusión se puede destacar que dependiendo del proceso que se desea programar, es importante seleccionar el lenguaje apropiado, con el fin de desarrollar códigos más pequeños y fáciles de entender a la hora de buscar una falla.

Por otro lado, también es recomendable que en procesos más complejos donde debe trabajar con grandes cantidades de datos, se recomienda utilizar subrutinas para cada operación o grupo de ellas, ya que de esta forma se es más simple la solución de problemas.

Respecto a los sistemas de visión artificial es importante destacar que, si bien son de mucha ayuda y bastante más precisos que el ojo humano, son muy propensos a los factores externos como la luz ambiente, lo que en algunas ocasiones produce un mal desempeño debido a su variabilidad durante el día.

4.3 BIBLIOGRAFÍA

FESTO. 2019. Festo Didactic. [En línea] 2019. [Citado el: 3 de 1 de 2019.]
<https://www.festo-didactic.com/int-es/learning-systems/software-e-learning/ciros/ciros-education-uso-de-entornos-virtuales-de-aprendizaje.htm?fbid=aW50LmVzLjU1Ny4xNC4xOC4xMTEwLjgxODc>.

M. Alonzo, Freddy y E. Brravo, Miguel. 2014. Diseño, construcción y control de un brazo robótico. *Diseño, construcción y control de un brazo robótico*. Quito : Universidad San Francisco de Quito, 2014.

Mery, Domingo. 2002. *Inspección Visual Automática*. Santiago de Chile : Universidad de Santiago de Chile, 2002. C2_1.

MICRO, automatización. 2018. REDES AS-i. [En línea] 2018.
<http://www.microautomacion.com/capacitacion/Manual071RedesASi.pdf>.

Salazar Serna, César Augusto y Correa Ortiz, Luis Carlos. 2011. [En línea] 10 de Diciembre de 2011. [Citado el: 4 de Diciembre de 2018.]
<http://revistasum.umanizales.edu.co/ojs/index.php/ventanainformatica/article/view/File/126/184>.

SCHOBER. 2010. FESTO DIDACTIC. [En línea] 2010. [Citado el: 2 de ENERO de 2019.] http://jami.dale.people.cpsc.edu/REF%20-%20Festo%20Documentation/REF%20-%20MPS500/S3%20-%20Vision/MPS500_Manual_Vision_A004.pdf.

Schober. 2010. MPS500 Manual Vision. [aut. libro] FESTO. *MPS500 Manual Vision*. 2010, pág. 53.

Wolfgang Eckart, ADIRO. 2002. <https://www.festo.com/didactic>.
<https://www.festo.com/didactic>. [En línea] 2002. [Citado el: 2 de Enero de 2019.]
http://jami.dale.people.cpsc.edu/REF%20-%20Festo%20Documentation/REF%20-%20MPS500/S5%20-%20Auto%20Store%20&%20Retriv/Technical%20Documentation/English/Teachware/161_ASRS_Instructor_RelC.pdf.