

UNIVERSIDAD POPULAR AUTÓNOMA DEL ESTADO DE PUEBLA
FACULTAD DE INGENIERÍA MECATRONICA, ELECTRONICA, BIÓNICA
Y AEROESPACIAL



MONOGRAFÍA PARA OPTAR EL TÍTULO EN LICENCIATURA EN
INGENIERÍA MECATRÓNICA

IMPORTANCIA DEL CONOCIMIENTO ACERCA DE LOS DISTINTOS
SOFTWARES Y LENGUAJES DE PROGRAMACIÓN PARA LA
AUTOMATIZACIÓN DE PROCESOS EN LA INDUSTRIA.

Asesor:

Edson Olmedo Urrutia

Estudiantes:

Adolfo Fernández Cuspinera

Jorge Cisneros Ortega

Rodrigo Muñoz Villanueva

Puebla, 2022



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyla**

Tesis Digitales Restricciones de uso:

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de textos, imágenes, gráficas, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Introducción

La automatización de procesos empleando controladores lógicos programables (PLC's), normalmente se basa en el uso de bloques definidos, por lo cual en algunos casos dada su complejidad y lógica, no existe metodología alguna para repetir el proceso, esto ocasiona cierta dificultad en los diseños, dado que algunos procesos son exclusivos y poco entendibles. Al punto en el que únicamente el diseñador puede ser la única persona que lo comprenda en su totalidad.

Antes que nada tenemos que comprender que es la automatización, la cual se entiende que es un sistema en el cual se transfieren ciertas tareas de producción, las cuales pueden ser frecuentemente realizadas por operadores humanos con ayuda de un conjunto de elementos tecnológicos, un sistema de este tipo consta de dos partes principales.

1. Parte Operativa
2. Parte de mando

Parte Operativa

Se caracteriza por ser la parte donde se actúa directamente sobre la máquina. Esta parte comprende todos los elementos necesarios para que la máquina se mueva y realice la operación deseada. Podemos concluir que los elementos que forman esta parte operativa son los accionadores de las máquinas, los cuales pueden ser componentes como motores, cilindros, compresores e inclusive los captadores como fotodiodos y transductores.

Así como los humanos necesitamos nuestros sentidos para poder percibir todo lo que se encuentra alrededor de nuestro entorno. Cuando hablamos de sistemas automatizados en lugar de nuestros sentidos ocupamos de los transductores para poder obtener la información que necesitamos en nuestro proceso. gracias a los transductores podemos obtener mucha información tanto de la variación de ciertas magnitudes en el sistema como el estado físico de sus componentes.

Los transductores los podemos clasificar en base al tipo de señal que transmiten, en general existen tres grupos principales:

- **Los transductores todo o nada:** se encargan de suministrar una señal binaria que puede ser claramente diferenciada. Los ENDSTOP - finales de carrera son transductores de este tipo.
- **Los transductores numéricos:** se encargan de transmitir valores numéricos en forma de combinaciones binarias un ejemplo de transductores de este tipo son los encoders.

- **Los transductores analógicos:** suministran una señal continua la cual proyecta el reflejo fiel de la variación de la magnitud física medida en cuestión.

Objetivos principales de la automatización

- Mejorar la calidad y mantener calidad
- Producir cantidades necesarias en el momento preciso
- Mejorar productividad y reducir costes
- Hacer flexible producción

Una automatización es un sistema o máquina que puede responder automáticamente a los cambios que se producen en su interior. El objetivo de la automatización es controlar equipos sin necesidad de intervención del operador. Este último actúa sobre las variables de control que el automatismo se encarga de actuar sobre la salida.

Nivel de automatización

1. Nivel de máquina: máquinas (prensas) que realizan trabajo productivo
2. Nivel de Celda (Grupo): Control automático del grupo (2Robots)
3. Nivel de fábrica: control automático de toda la planta de producción (barco) trabajando en conjunto para lograr los objetivos de producción
4. Nivel de empresa: todos los barcos establecidos

Sistema de automatización

En el Nivel 2, puede existir una red de comunicación local entre los elementos de la celda mediante sensores, actuadores, etc. Las señales de entrada y salida pueden ser de cualquier tipo, pero los conceptos de automatización tradicionales se utilizan para sistemas de eventos discretos que requieren señales binarias. Es decir, toma solo dos valores (activo o inactivo).

Ventajas de automatizar

- Reducción de costos.
- En las secciones de fabricación automatizada se requieren menos operarios.
- Se puede fabricar las 24 hrs.

- Las máquinas cometen menos errores que los humanos y los productos consiguen un nivel alto de calidad.
- Tiempos de procesos menores.
- Reducción de trabajos monótonos pesados y peligrosos.

Desventajas

- Eliminación de puestos de trabajo
- La automatización de procesos implica que los operarios tomen decisiones específicas
- Los costos originarios por un sistema automático tienen mayor consecuencia que los individuos asumen una mayor responsabilidad de éxito de la empresa
- Para configurar procesos automáticos modernos se necesitan 3 componentes:
 1. Sensores para captar los estados
 2. Actuadores para emitir comandos de control
 3. Unidades de control para la ejecución del programa y toma de decisiones

Sensores industriales

- Final de carrera
- Detector inductivo
- Detector capacitivo
- Detector ultrasónico
- Detector fotoeléctrico
- Detector de presión

Actuadores Industriales

- Accionamiento Eléctrico (Motores DC, AC, Servomotor)
- Accionamiento Neumático (Cilindros, Motores)
- Accionamiento Hidráulico (Cilindro, Motores)

Parte operativa: Formada principalmente por el conjunto de dispositivos máquinas y sub procesos diseñados para realizar procesos.

Parte de Control: Formada de elementos de procesamiento, mando, interfaz de comunicación y de diálogo con el hombre.

La parte de control envía acciones de mando a través de los preaccionadores que son elementos que permiten el manejo de grandes potencias a partir de señales de baja potencia.

Automatismos Programados

Se implementan por medio de un programa que se ejecuta en un microprocesador, las instrucciones de este programa determinan la función lógica que relaciona la entrada con las salidas.

- Autómata programable industrial
- Ordenador
- Microcontrolador

Sistema de evento discreto

Es aquel que evoluciona entre un número finito de estados y cuyo cambio de estado depende del valor de determinadas variables binarias.

Grafcet Reglas

1. Generar situación de reposo con condiciones iniciales
2. Una transición está en disposición de ser válida
3. Franquear una transición implica la activación de todas las etapas siguientes inmediatas y la desactivación de las inmediatas
4. Transiciones en paralelo se activan de forma simultánea
5. Una o varias acciones se asocian a cada etapa y solo se activan cuando la etapa está activa

Introducción a los PLC

Los autómatas programables industriales son dispositivos electrónicos programables diseñados para la programación en tiempo real. Utiliza un lenguaje especial diseñado específicamente para facilitar la generación de programas que implementen los algoritmos. Utilizamos programas robustos que pueden soportar condiciones de trabajo duras, como entornos industriales.

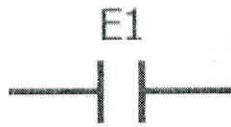
Diagrama de escalera

Los planes de contacto (también conocidos como escaleras o diagramas de escalera) son un lenguaje de programación gráfico muy popular dentro de los controladores lógicos programables porque se basan en esquemas de control tradicionales. De esta forma, con los conocimientos que todo ingeniero o ingeniero eléctrico tiene, es muy fácil acostumbrarse a programar en este tipo de lenguaje. Se llama "KOP" en STEP7.

Ladder es uno de una variedad de lenguajes de programación de controladores lógicos programables (PLC) estandarizados en IEC 61131-3. En una escalera, la energía se mueve de izquierda a derecha en lugar de arriba hacia abajo como en los esquemas eléctricos. En un circuito típico, los contactos se muestran a la izquierda y las bobinas a la derecha. La lógica de control representada por este circuito puede pensarse como un sistema que asume una lógica de contacto y concluye con una bobina.

La programación de un autómatas en escalera requiere no solo familiaridad con las convenciones de los circuitos (también conocidas como lógica de contacto), sino también el conocimiento de todos los elementos que componen este lenguaje. Los más comunes se describen generalmente a continuación.

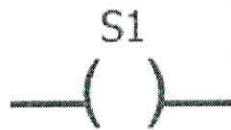
- Contacto normalmente abierto (E1): si la variable asociada E1 vale '0', el contacto permanece abierto, y si vale '1' se cierra.
- Contacto normalmente cerrado (E2): si la variable asociada E1 vale '1', el contacto permanece abierto, y si vale '0' se cierra.
- Salida, bobina o relé (S1): la variable asociada S1 tomará el valor de la variable (o combinación de variables) que esté a su entrada (punto de conexión del lado izquierdo). También se puede enclavar o desenclavar, indicándose con una S o R como se indica en los casos de S2 y S3.



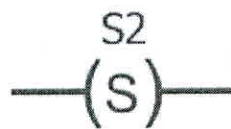
Entrada: contacto normalmente abierto



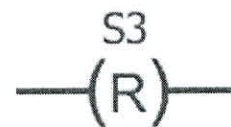
Entrada: contacto normalmente cerrado



Salida, bobina o relé

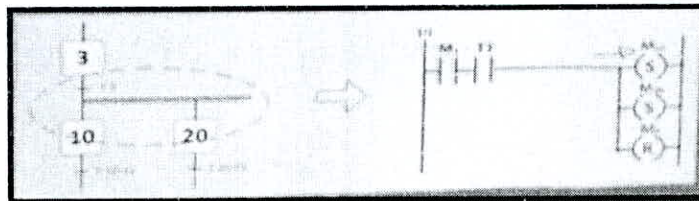


Activar salida, bobina o relé

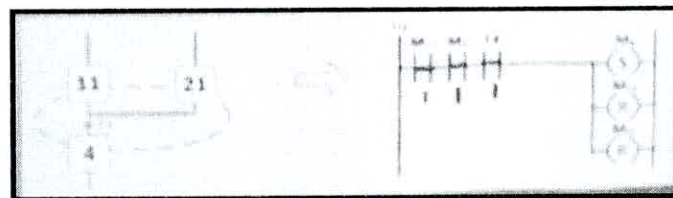


Desactivar salida, bobina o relé puesto '0'

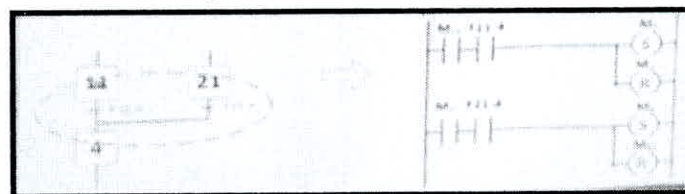
Secuencias simultáneas divergencia AND.



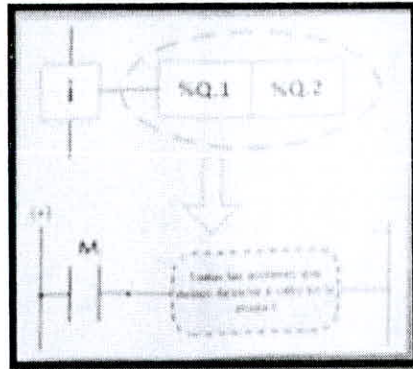
Secuencia Convergencia AND.



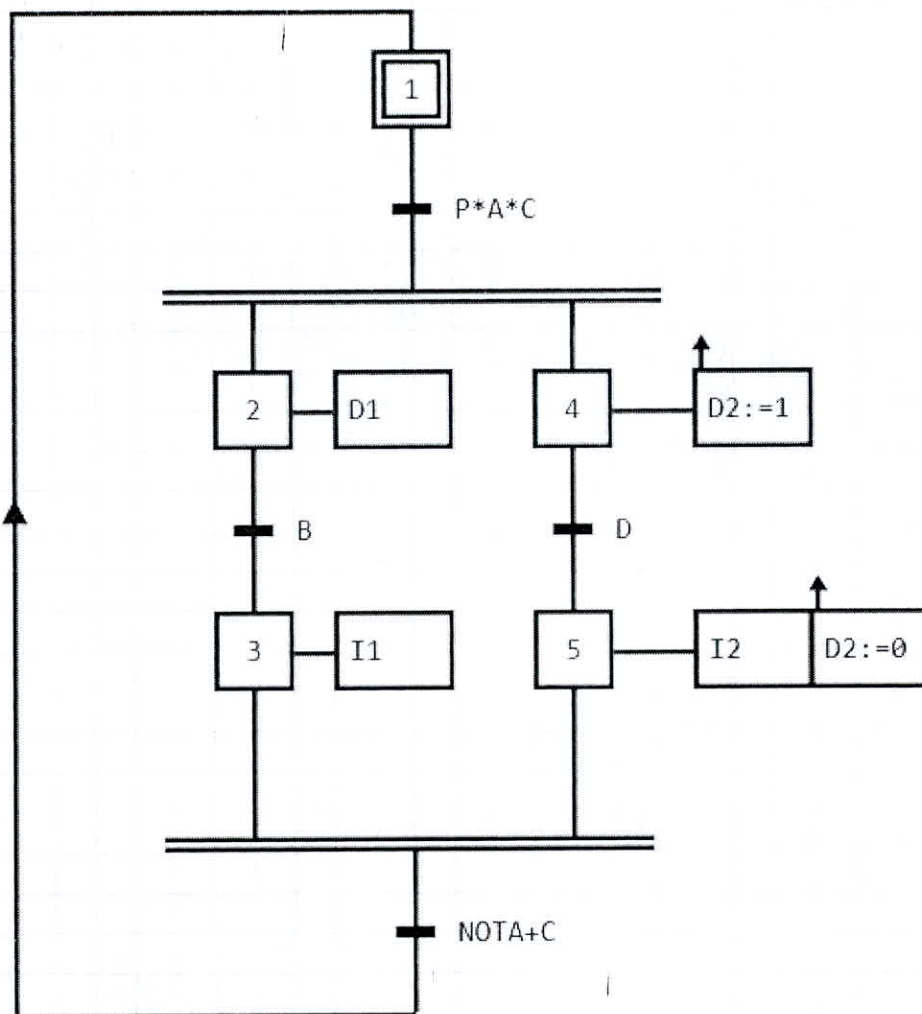
Secuencia alternativa Convergencia OR.



El programa ladder deberá contemplar las acciones asociadas a cada etapa, considerando las marcas internas activas que corresponden a cada una

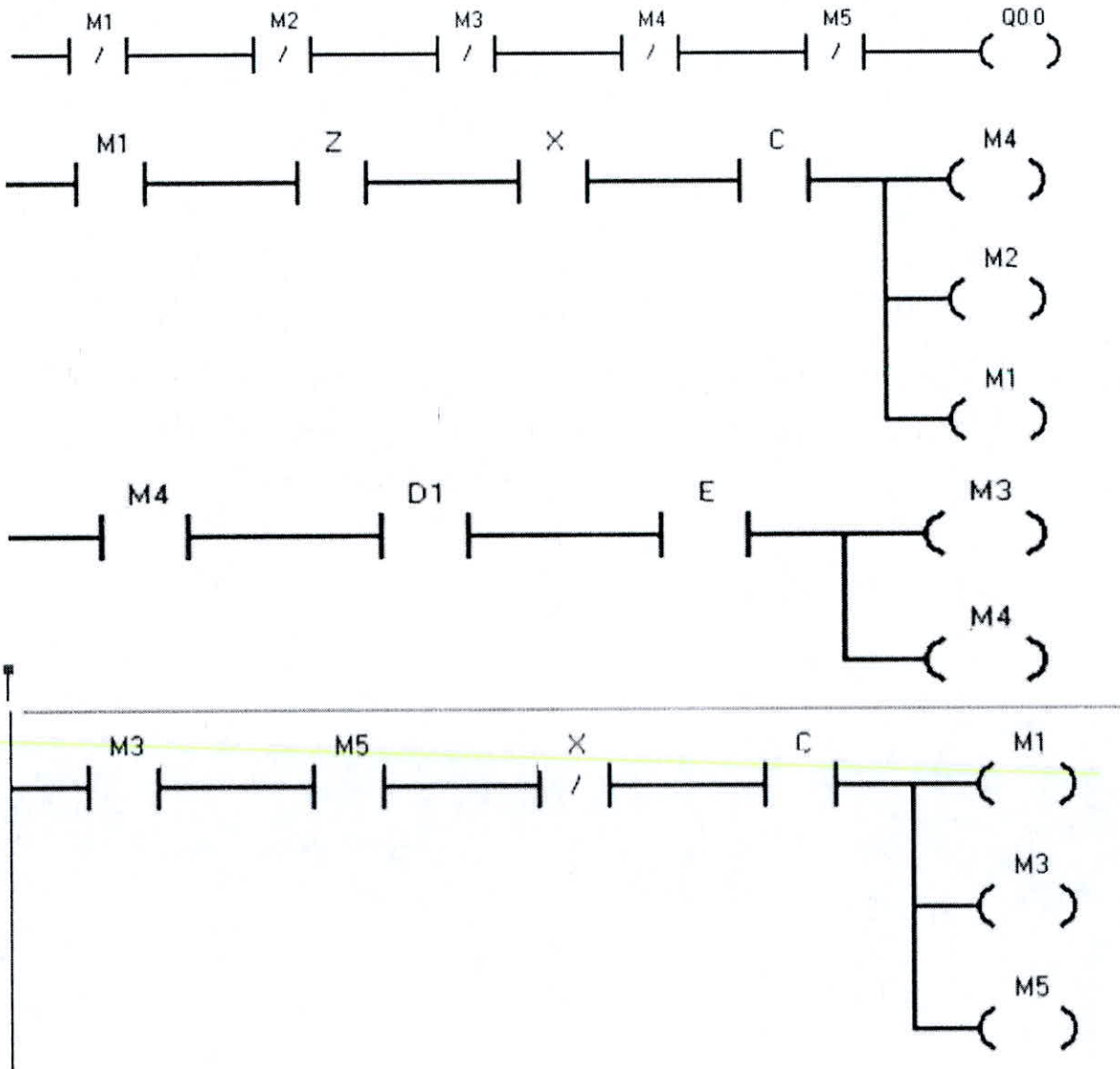


Ejemplo realizado en clase:

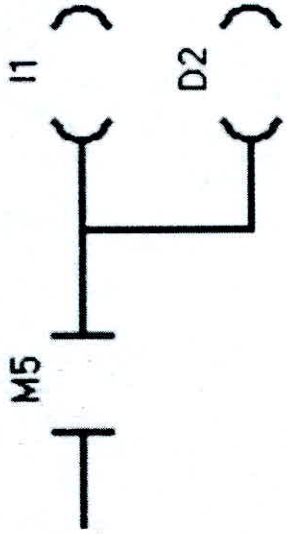
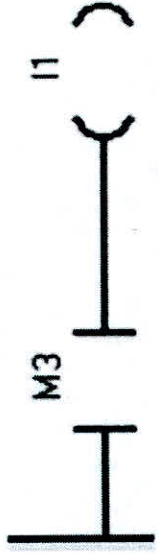
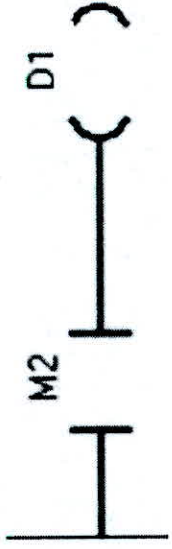
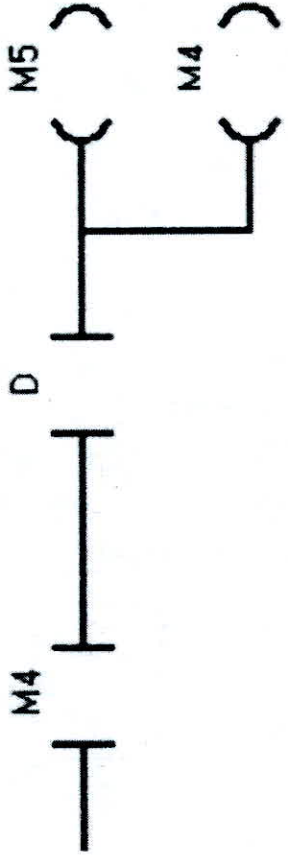


Tomando como ejemplo este ejercicio propuesto debemos transformar el lenguaje grafcet al lenguaje de escalera tomando las notas de la sección de arriba podemos empezar a construir nuestro programa, cambiando cada etapa por una respectiva

entrada sirviendo las condicionales como nuestros puntos clave para desarrollar el ejercicio



Podemos observar que esta es la parte principal del programa en el cual inicia una vez que nuestros sensores o pulsadores se encuentren activos, dando luz verde a que el proceso de divergencia se realice.



Una vez que el proceso llega a la etapa de convergencia como podemos observar en el ejemplo una etapa no depende para poderla llevar a cabo, la única dependencia se encuentra al final del ejemplo ya que requiere que cada etapa esté concluida para poder completar un ciclo y así lograr en este caso el ejercicio que se plantea en un inicio.

Qué es PLC

Se trata de un ordenador industrial que utiliza esta tecnología para automatizar procesos, cuyo objetivo es desarrollar de forma eficaz todos los sistemas que componen la máquina. Gracias a estas ventajas, el PLC se ha convertido en una herramienta fundamental para el desarrollo tecnológico industrial y de todo el entorno social.

¿Cómo funcionan los PLC?

El funcionamiento del PLC se basa en procesos regulares y continuos. Las secuencias para estos dispositivos se describen a continuación.

- Autodiagnóstico: Es una revisión de todos los circuitos. El dispositivo mostrará una señal si hay un problema.
- Lea y registre entradas: evalúe cada entrada para diagnosticar si está activada o desactivada, y registre estos procesos en la memoria para crear una imagen. Carga y ejecución de programas: la computadora ejecuta programas instruidos por el usuario en función de las imágenes en la memoria.
- Registrar y actualizar salidas: En este paso se restauran todas las salidas al mismo tiempo.

Algunas características de los PLC

- Controlan las entradas y salidas de manera segura
- Poseen una programación compatible con distintos lenguajes
- Interfaz amigable que facilita la comunicación con el usuario
- Conexión a sistemas de supervisión
- Ejecutan la programación de forma continuada
- Memorias divididas en dos partes

El conjunto de estos pasos permite diagnosticar las distintas señales dentro de un proceso, arrojando resultados compatibles con la programación. Es importante destacar que dicha programación puede ser reconfigurada, en caso de requerirlo.

¿En qué sector de la industria se usa el PLC?

Como lo comentamos anteriormente, en el mercado hay disponibilidad de PLC de acuerdo a la necesidad de automatización que requiera la industria.

Facilitamos una lista referencial de sectores empresariales que han utilizado estos autómatas, multiplicando la eficiencia en sus sistemas y ahorrando costos.

- Empresas dedicadas al procesamiento de cemento y arenas
- Elaboradores de plástico
- Para la realización de maderas y puertas
- Empresas de seguridad y transporte
- Instalación de aires acondicionados y calefacción
- Industrias de automóviles y neumáticos
- Corporaciones petroleras y químicas
- Compañías energéticas
- Tráfico
- Urbanizaciones residenciales, entre otros.

PROFINET

PROFINET es compatible con proveedores de tecnología de control (vendedores). Tan importante como los datos de rendimiento técnico del sistema de comunicación es el acceso al sistema de control más importante.

Muchos fabricantes de control, como Phoenix Contact y Siemens, ya han implementado interfaces PROFINET en sus dispositivos, ofreciendo un amplio soporte de protocolos y acceso a muchas soluciones de automatización en todos los sectores.

El establecimiento de un proceso de certificación probado garantiza la alta calidad de los productos PROFINET y su interoperabilidad dentro del sistema.

Productividad incrementada.

Para poder producir económicamente, una planta de producción debe lograr un retorno de la inversión (ROI). Esto requiere una reducción significativa en los costos de desarrollo e ingeniería, que representan una gran parte del costo total, y una conversión simple de las soluciones existentes a nuevos conceptos.

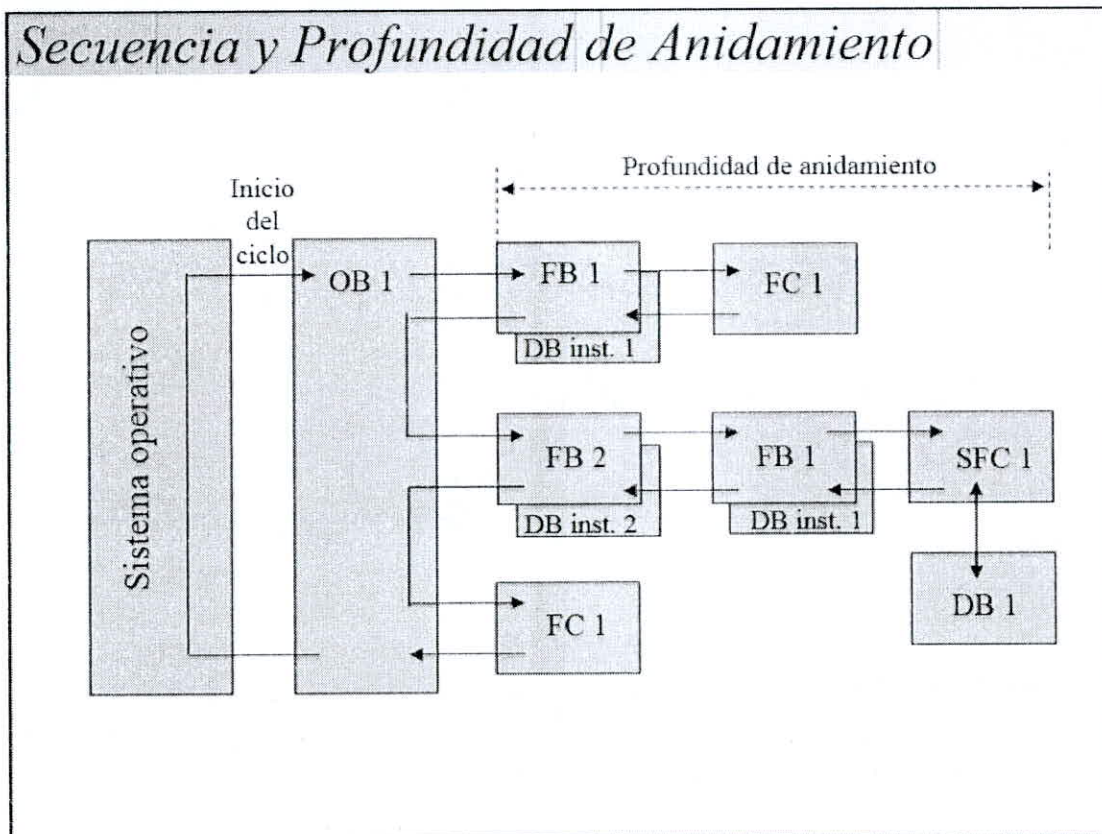
Por lo tanto, la estructura de comunicación dentro de una empresa tiene una influencia decisiva en el desarrollo de nuevas oportunidades de productividad.

Para los fabricantes, el uso de PROFINET minimiza los costos de instalación, ingeniería y puesta en marcha. Al mismo tiempo, los componentes de la planta que funcionan de forma autónoma y los bajos costos de mantenimiento facilitan la expansión de las plantas de producción y aseguran una alta disponibilidad de la planta. Características clave de PROFINET

La comunicación integrada basada en Ethernet (todo en un cable) satisface una amplia gama de requisitos, desde la parametrización de datos intensivos hasta la transmisión ultrarrápida de datos de E/S. De esta forma, PROFINET permite la automatización en tiempo real. Además, proporciona una interfaz directa al nivel de TI.

Diagrama de bloques funcional

Es una representación gráfica de procesos funcionales con bloques y diagramas de fácil comprensión e interpretación. FBD ayuda a determinar la función entre las variables de entrada y salida a través de una serie de bloques y diagramas básicos conectados por flechas llamadas "conexiones".



El diagrama de bloques funcionales también es conocido como diagrama de flujo funcional. Como su nombre indica, es una representación paso a paso de un flujo que ayuda a simplificar los procesos de trabajo y crear una mejor comprensión de los mismos. La idea fue dada por *Frank Gilbreth en 1921*, precedida por otros ingenieros y científicos que desarrollaron un modelo de proceso de varios niveles para simplificar múltiples funciones y las relaciones entre ellas.

En la actualidad el diagrama de bloques funcionales ocupa una posición ventajosa siendo ampliamente utilizado en varios campos de:

- Rediseño de procesos de negocio
- Gestión de procesos de negocio
- Ingeniería de sistemas informáticos
- Ingeniería de Sistemas

Un Diagrama de Bloques Funcionales puede ayudarnos a crear relaciones entre una o más variables ayuda a crear relaciones entre una o más variables para establecer una comprensión de los procesos funcionales alineados en un sistema.

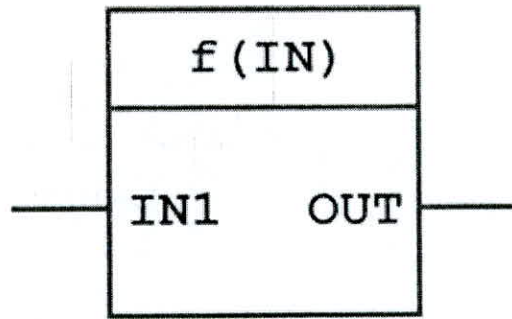
Estos diagramas son de utilidad para entender las funciones y las relaciones entre dos o más variables utilizadas en:

- Ingeniería de Software
- Ingeniería de Sistemas

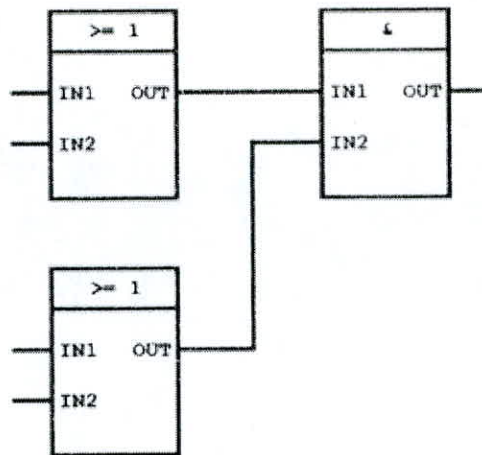
Para los ingenieros y programadores, el FBD es una herramienta esencial que les ayuda a entender y crear correlaciones entre dos o más variables conectándolas con una flecha de conexión.

Funciones

Las funciones que se ponen en un bloque funcional que se demuestra con una caja. Una caja cuadrada es un símbolo de una función, como se muestra en la siguiente imagen.



Un bloque funcional puede tener dos o más entradas y salidas. Las entradas y salidas pueden conectarse con otras entradas y salidas del otro bloque funcional, estableciendo así una relación entre la salida de una función y la entrada de otra, como se observa en el siguiente diagrama.



Tipos de Diagramas de Bloques Funcionales

Bloques de función biestables

Son conocidos por ser la forma más simple de memoria. Depende de si quieres restablecer o establecer una salida. La salida aprenderá y recordará el último punto de la entrada establecida.

Detección de Bordos

Este tipo de bloque de función es muy útil y se utiliza en la programación de PLC y en la electrónica. Su nombre se debe a que cuando la entrada detecta un borde

progresivo, la salida se activa. Y se detecta porque la salida desarrolla un pulso cuando se detecta un borde positivo.

Bloques de Función de Temporizador

Se utilizan a gran escala en la ingeniería de PLC. Existen tres tipos de bloques de temporizador; estos tipos de bloques incluyen un temporizador de retardo, un temporizador de pulsos.

Bloques de Función de Contador

En este bloque se toman entradas y salidas, y otro tipo de datos. Existen tres tipos de bloques de función: contador, contador ascendente, contador descendente y contador ascendente-descendente.

GRAFCET

En el mundo de la automatización se conocen varios métodos para programar y desarrollar procesos lógicos. Algunos de estos métodos eran completamente intuitivos y luego dieron paso a varios lenguajes de programación. En los primeros días de la automatización industrial, la programación estaba reservada para expertos en la materia. Sin embargo, en estos días hay otra forma de implementar el mismo diagrama lógico. Un método muy común y fácil es GRAFCET. A diferencia de otros lenguajes de programación, GRAFCET es un sistema gráfico con una sintaxis sencilla. Se pueden usar algunos comandos simples para especificar procesos automatizados a través de un flujo lógico. El propio nombre indica el sistema utilizado. Panel de funciones para pasos y transiciones (GRAFCET). El nacimiento de la industria responde a la necesidad de una forma sencilla para que los operadores demuestren cómo hacer su trabajo sin especialización.

Cómo programar en GRAFCET

GRAFCET consta de varios elementos en un flujo de trabajo lógico. A través de una serie de procesos, estos elementos se distribuyen en etapas, acciones asociadas y transiciones.

Los sistemas que utilizan GRAFCET siempre deben comenzar en el nivel inicial. Luego puede agregar etapas regulares o fijas. Estas llamadas etapas se refieren al estado del proceso de la máquina, habilitado o deshabilitado, y las acciones asociadas. Se requiere un director para esa etapa, donde se realizan las acciones correspondientes.

Las acciones asociadas son aquellas que deben realizarse en la etapa pre asignada. Es decir, la acción a realizar. En los procesos automatizados, esto puede variar desde el arranque de bobinas hasta el encendido y apagado de motores. Al mismo tiempo, también existen acciones asociadas a condiciones. Las condiciones también están marcadas por el nivel. Tales acciones pueden retrasarse, detenerse condicionalmente o restringirse.

Finalmente, una transición es un paso evolutivo de una etapa a la siguiente. Una transición es un segmento de condiciones que el sistema debe superar para llegar a la siguiente etapa. Estas condiciones, también llamadas sensibilidades, van desde acciones simples hasta funciones booleanas. Finalmente, las líneas de conexión se utilizan para crear un flujo o diagrama completo. Estos pueden ser verticales u horizontales y dependen directamente del elemento anterior.

Este método de programación abre posibilidades para trabajar con diagramas secuenciales y paralelos y muchas otras características.

Ventajas de los conjuntos de gráficos

- Fácil programación con sistema visual simple
- Idiomas alternativos para diferentes niveles de operadores y técnicos
- Método de flujo para una fácil detección de errores

La programación de GRAFCET es fácil de entender y fácil de implementar, lo que la convierte en un recurso valioso para los operadores y los recursos técnicos en la industria de la automatización.

Simulación de la industria

La simulación de procesos industriales permite la reproducción virtual de un proceso para estudiar su comportamiento, el análisis de los efectos de diferentes variables que pueden intervenir en el proceso y el estudio de varios diseños sin una gran experimentación. Es una herramienta que permite comparar propuestas. . Esto es de gran ayuda en la mitigación de riesgos y la optimización del proceso de toma de decisiones, así como en la planificación, el análisis y la mejora de los procesos comerciales.

¿Por qué es útil la simulación de procesos?

A través de la simulación, los procesos pueden medirse o esbozar mediante la creación de modelos que contengan sistemas de producción de fábrica en un entorno virtual. Con los procesos virtuales, se pueden corregir errores e ineficiencias sin afectar la planta de producción y se pueden predecir los resultados. A través de la

simulación se puede analizar cualquier tipo, cambio o propuesta antes de su ejecución sin coste adicional, de forma rápida, precisa y sin riesgo.

Al simular y analizar el proceso, puede verificar el diseño y la posición de los elementos que componen el proceso, optimizar el tiempo de ciclo y verificar y optimizar el acceso, la manipulación y la ausencia de colisiones.

Algunas de las ventajas encontradas en la simulación son:

- Explorar diferentes alternativas.
- Optimizar los tiempos de producción.
- Aumentar la calidad y fiabilidad del diseño.
- Análisis de puntos críticos del proceso.
 - Ergonomía
 - Producción
 - Mantenimiento.
 - Logística.
 - Layout (disposición de medios)
- Análisis de la capacidad máxima de producción.
- Evaluar el diseño de instalaciones para adaptarse a la fabricación de nuevos modelos.
- Evitar costes extra al simular el proceso antes de instaurarlo.
- Visualización del proceso antes de la implantación.
- Facilidad de revisión, modificación y optimización de diseños en tiempo real.
- Reducir tiempos de implantación.
- Mayor impacto visual.

Además, como beneficios intangibles, encontramos que la simulación posibilita que los clientes entiendan y comprendan los procesos de una forma completa, permitiendo identificar los problemas de una forma clara y concisa.

Ejemplos de simuladores ocupados en el diplomado

STEP 7

Es el software estándar para configurar y programar los sistemas de automatización SIMATIC. STEP 7 forma parte del software industrial SIMATIC. Funciones del software estándar El software estándar le asiste en todas las fases de creación de soluciones de automatización, tales como

- crear y gestionar proyectos
- configurar y parametrizar el hardware y la comunicación
- gestionar símbolos

- crear programas, p.ej. para sistemas de destino S7
- cargar programas en sistemas de destino
- comprobar el sistema automatizado
- diagnosticar fallos de la instalación

El interfaz de usuario del software STEP 7 ha sido diseñado siguiendo los criterios ergonómicos más avanzados, lo que permite conocer rápidamente sus funciones.

PC Simu

Este es un simulador que le permite simular gráficamente procesos automatizados intercambiando entradas y salidas sin activar los interruptores de entrada ni mostrar los LED de salida en el PLC.

Funciona de dos formas: a través de un simulador (S7-200) oa través de un cable PC-PPI para comunicarse con un autómeta real.

Los elementos simulados incluyen interruptores, botones, detectores, teclados, interruptores selectores, potenciómetros, LED, pantallas, barras de progreso, texto, motores, convertidores de frecuencia, cintas transportadoras, puertas de garaje, actuadores lineales neumáticos, sin varilla, giratorios. Es una fórmula, un lechón. , deposición de sólidos y líquidos, activación de imágenes en formato BMP e incluso elevadores.

Roboguide

FANUC ROBOGUIDE es un simulador de robot que simula los propios movimientos del robot y las instrucciones del programa. Esto reduce significativamente el tiempo que se tarda en crear nuevos ajustes de movimiento. Las celdas se pueden diseñar, probar y modificar fuera de línea para minimizar el impacto en la producción. Para ahorrar tiempo en el modelado 3D, los modelos de piezas se pueden importar desde su PC como datos CAD. Una amplia biblioteca de software de simulación de robots también permite a los usuarios seleccionar y modificar piezas y dimensiones según sea necesario. Debido a su diseño intuitivo y facilidad de uso, ROBOGUIDE requiere poca capacitación. También hay disponibles herramientas especiales para aplicaciones específicas.

Factory IO: Simulación 3D de fábrica

Factory I0 es un software de simulación 3D, creado por Real Games, el cual nos permite construir y controlar procesos industriales en tiempo real.

Este tipo de simulación es totalmente interactiva e incluye gráficas de alta calidad y sonido, proporcionando un entorno realista industrial. A su vez utiliza tecnología innovadora, la cual permite la creación fácil y rápida de los sistemas automatizados en 3D, los cuales pueden ser controlados en tiempo real mediante la conexión de Factory I0 y equipos externos (PLCs o microprocesadores).

Las características principales que vuelven a Factory I0 un gran software para el aprendizaje de tareas de control realista, mediante PLC, son: 20 escenarios inspirados en las aplicaciones industriales más frecuentes, usando una librería con más de 80 componentes industriales, con los que se puede crear desde un escenario personalizado, hasta una fábrica virtual; la capacidad de crear estrategias para el diagnóstico de averías, ya sea por enchufe abierto o corto circuito; la seguridad y eficiencia del software, ya que nos muestra aplicaciones reales de la industria, evitando los problemas de costos, daño a personas y/o a equipos.

TIA PORTAL

La principal utilidad que TIA Portal ofrece es la posibilidad de integrar distintas aplicaciones de software industrial para procesos de producción en un mismo interfaz lo que facilita enormemente el aprendizaje, la interconexión y la operación. No importa si se trata de la programación de un controlador, de la configuración de una pantalla HMI o de la parametrización de los accionamientos: con esta nueva arquitectura de software tanto los usuarios nuevos como los expertos trabajan de una forma intuitiva y efectiva ya que no necesitan operar una amplia variedad de sistemas de diferentes orígenes.

EJEMPLO GRAFCET

En este caso vamos intentar automatizar una mezcladora, a continuación se mostrarán las instrucciones del proceso así como actuadores y sensores necesarios para realizar este trabajo.

Ejercicio: MEZCLADORA

Se quieren mezclar 2 productos con agua. Se llena el depósito de agua abriendo la válvula V1. La dosificación de los dos productos se realiza con una tolva acumulativa, se vierte el producto A sobre la tolva hasta que se alcanza un peso SP1 y a continuación se añade el producto B para conseguir el peso total de los 2 productos, SP2. Se abre la válvula de la tolva durante 10 segundos para dejar caer el contenido. Se realiza el proceso de mezclado durante 30 segundos accionando el agitador y se vacía el depósito para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor P.

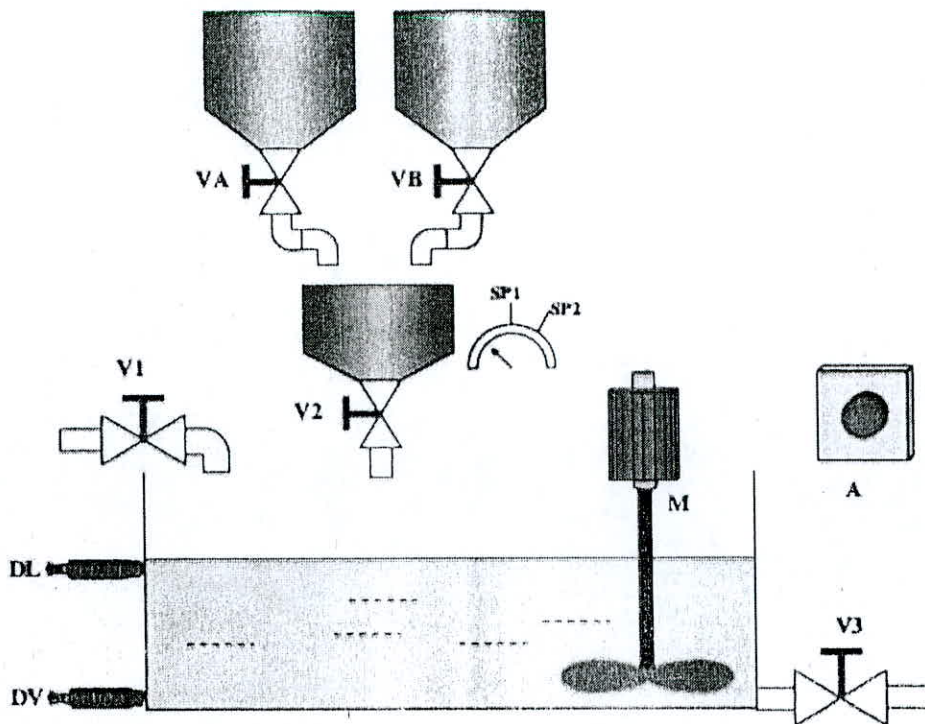


Ilustración 1: Elementos que componen la mezcladora

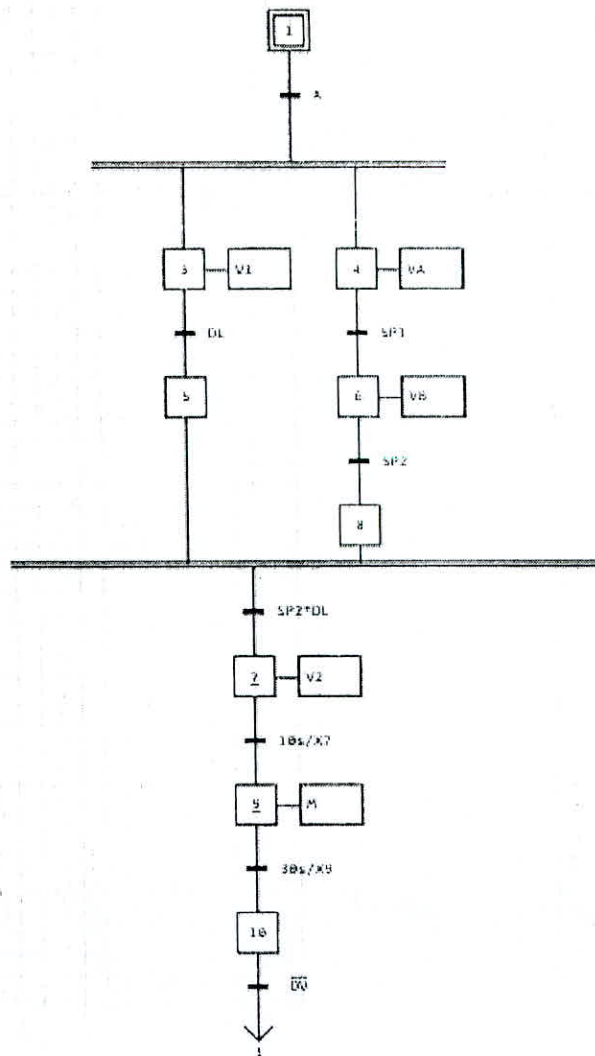


Ilustración 2: Diagrama realizado en grafcet estudio, listo para poderlo traducir a lenguaje LADDER de ser necesario, se aprecian las transiciones y las etapas en el proceso.

Proceso en lenguaje LADDER

Para este caso consideramos los datos como se muestran a continuación

ENTRADAS DEL SISTEMA AUTÓMATA

P	%I1.0
DL	%I1.1
DV	%I1.2
SP1	%I1.3
SP2	%I1.4

SALIDAS DEL SISTEMA AUTÓMATA

V1	%Q2.0
V2	%Q2.1
V3	%Q2.2
VA	%Q2.3
VB	%Q2.4
M	%Q2.5

ESTADOS

ESTADO INICIAL	%M0
LLENAR A	%M1
DOSIFICAR A	%M2
DOSIFICAR B	%M3
LLENAR B	%M4
LLENAR	%M5
VERTER	%M6
MEZCLAR	%M7
VACIAR	%M8

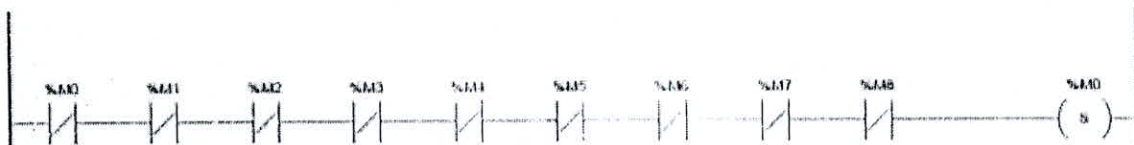
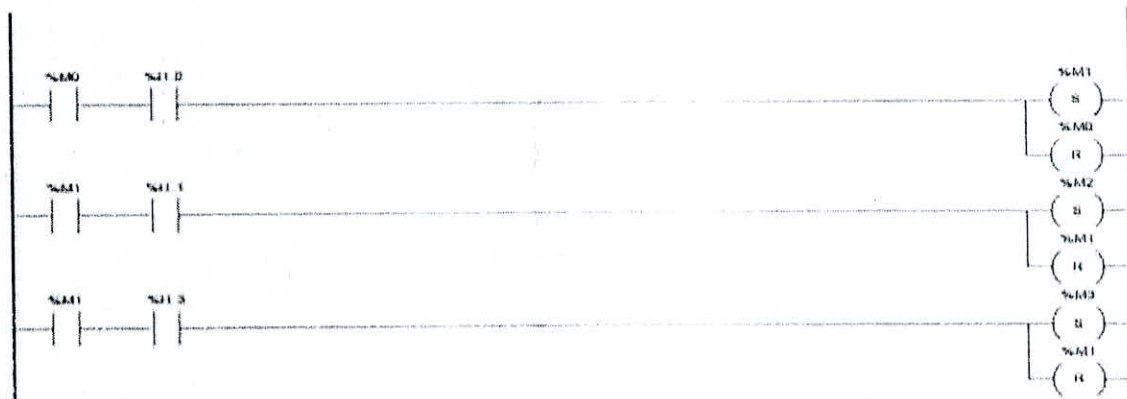


Ilustración 3: Primera parte, Inicialización.



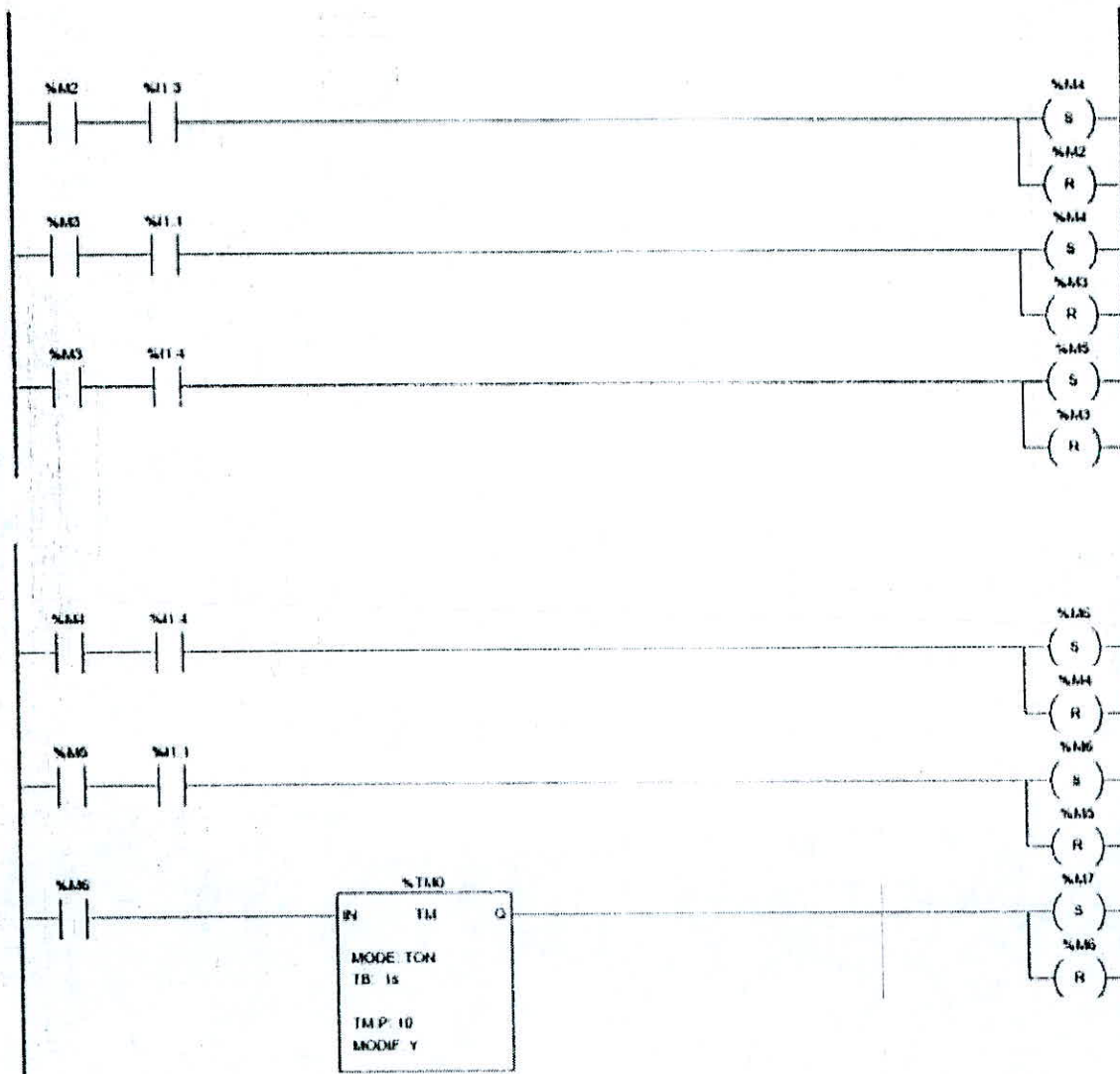


Ilustración 4: Programación de la secuencia de estados

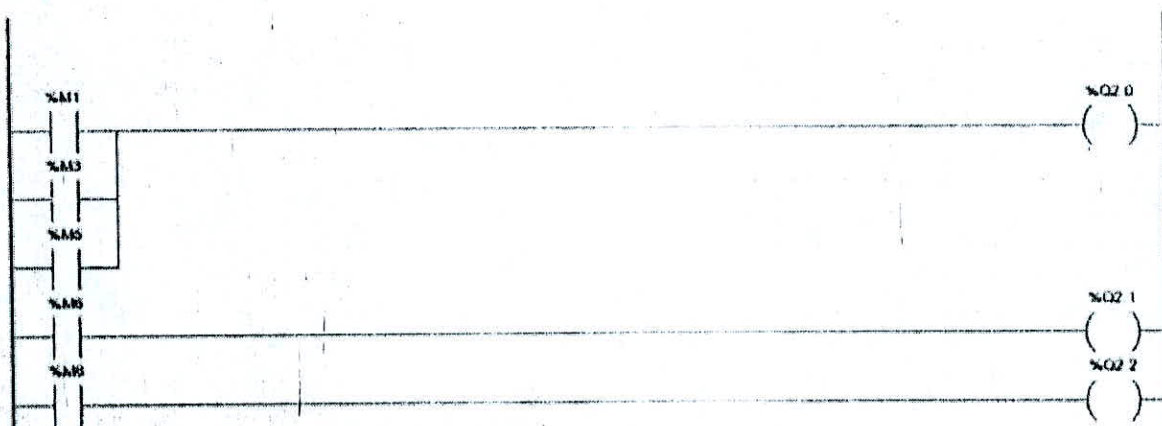


Ilustración 5: Activación de las salidas

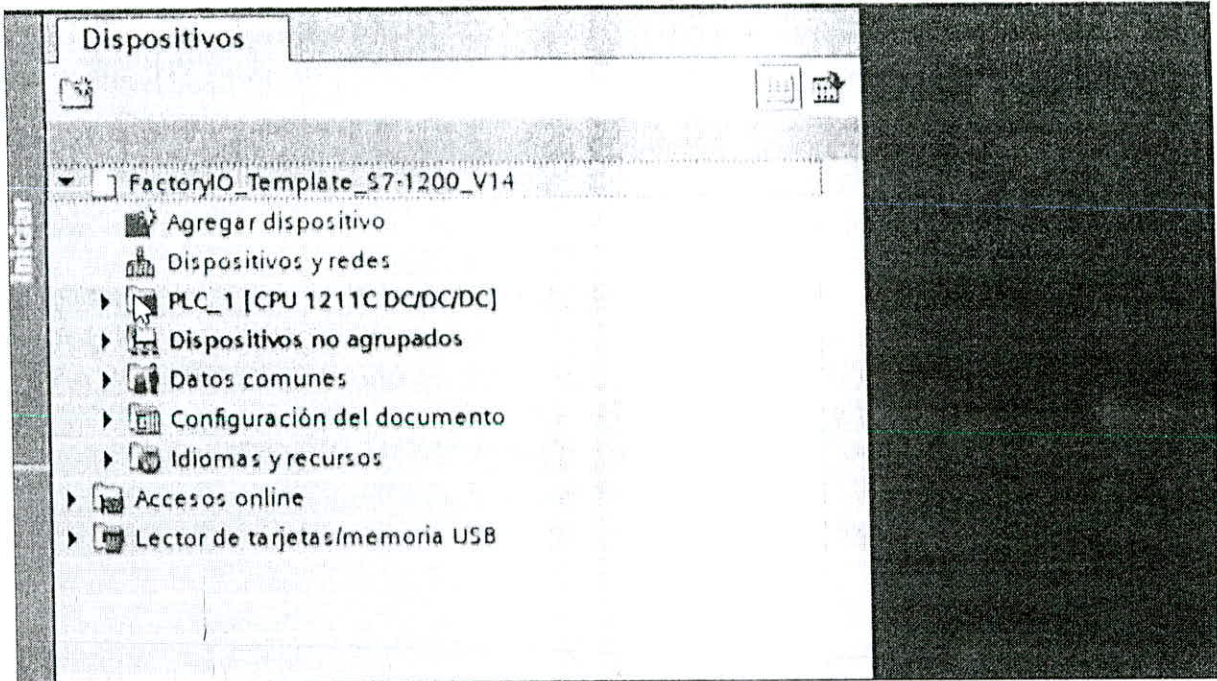


Ilustración 8:

- Se tiene que abrir el bloque de programas
- Seguido se abre el Main, en el lenguaje KOP
- En proyecto, guardamos como, se guarda para poder utilizarlo después
- Una vez guardado se puede iniciar simulación

OBI : "Main Program Sweep (Cycle)"

Comentario:

□ ~~Sum 1~~: Titulo:

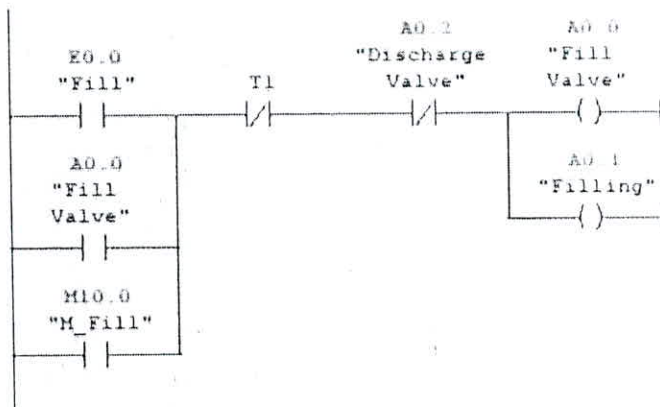


Ilustración 9 : Comenzamos con nuestro código ladder con nuestras variables principales en este caso valvula de llenado el temporizador (T1) , válvula de descarga y nuestros actuadores.

☐ Segm. 2 : Título:

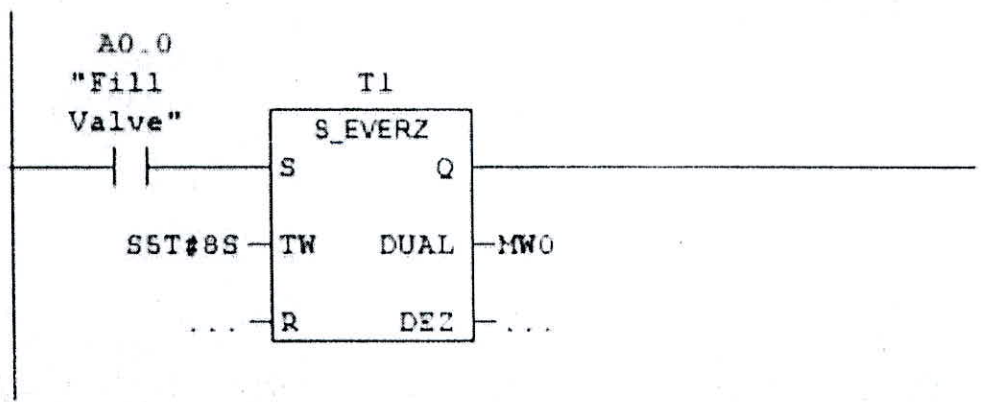


Ilustración 10: Utilizamos un temporizador de 8s con nuestra válvula de llenado ya que en este caso , solo llenaremos nuestro tanque por el tiempo determinado

☐ Segm. 3 : Título:

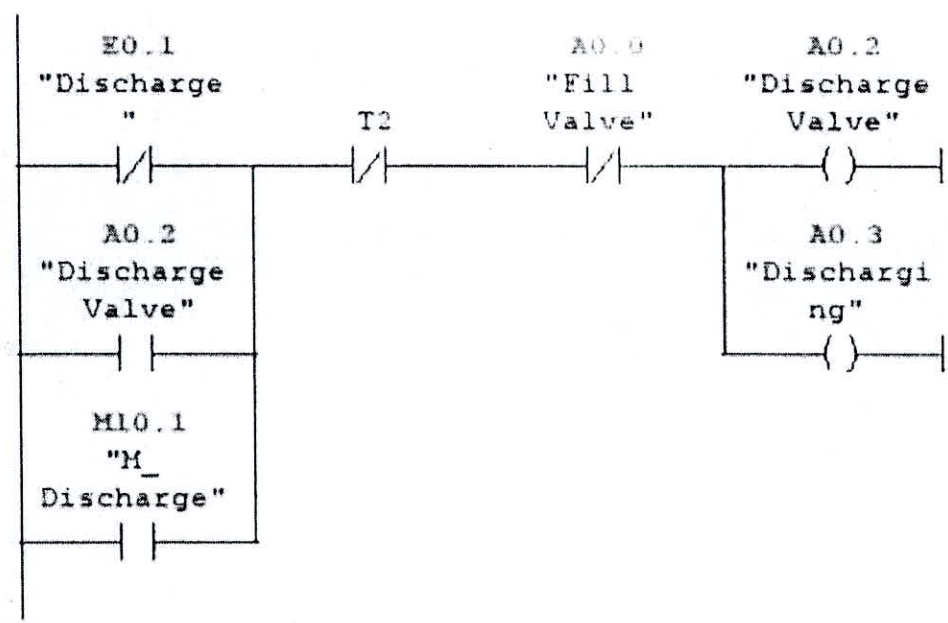


Ilustración 11: Una vez lleno utilizaremos nuestra válvula de descarga la cual será activada por un timer (T2) una vez lleno el tanque procederemos a vaciarlo activando nuestra válvula de vaciado conforme al tiempo determinado con nuestro timer

☐ Segm. 4 : Título:

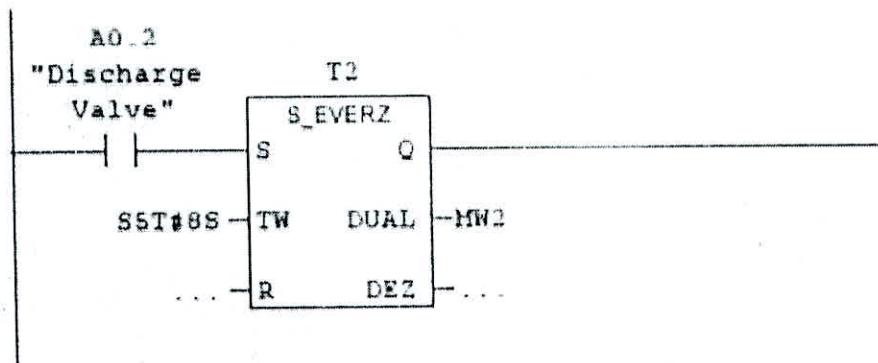


Ilustración 12: Timer 2 con este activaremos nuestra válvula de descarga la cual será activada de igual manera durante 8 segundos con esto nos aseguramos que tenga la misma cantidad de vaciado.

☐ Segm. 5 : Título:

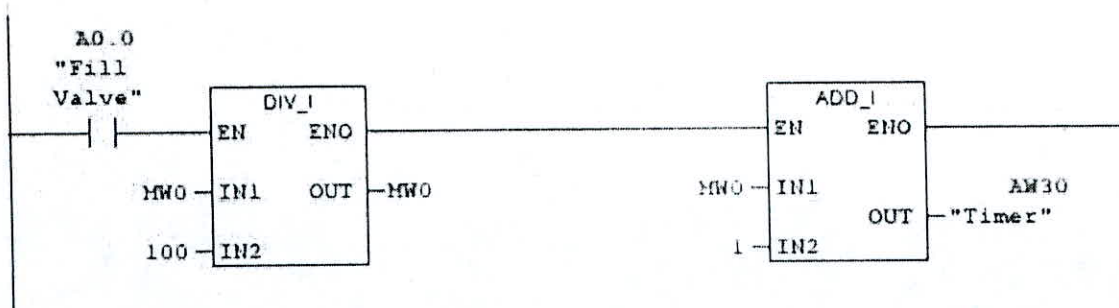


Ilustración 13: Válvula de llenado activada por un timer la función ADD_i sumar enteros

☐ Segm. 6 : Título:

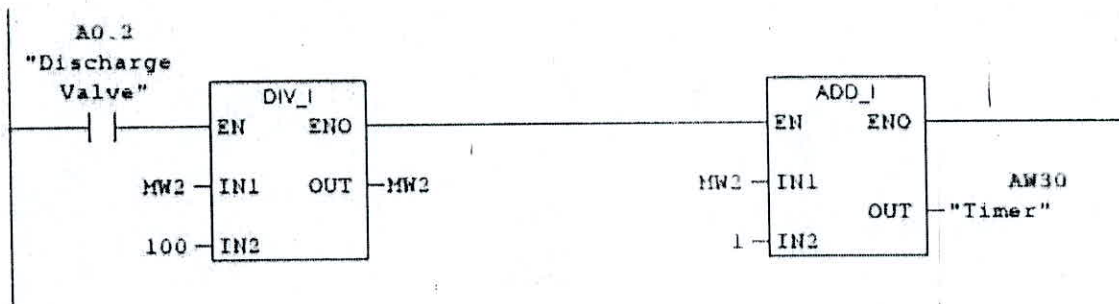


Ilustración 14: Válvula de descarga activada por nuestro timer

Segm. 7 : Titulo:

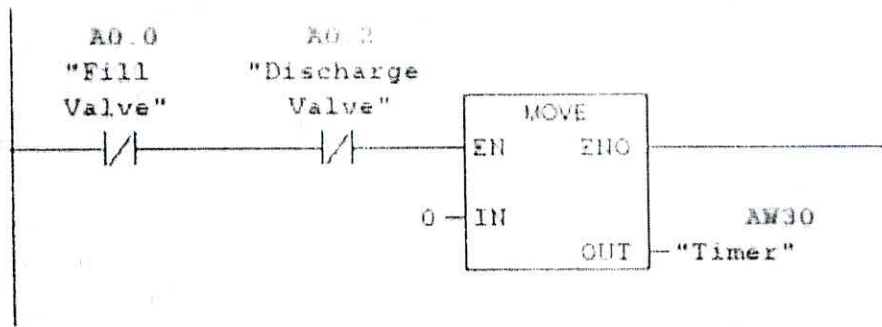


Ilustración 15: Utilizamos la función move la cual asigna un valor que es activada por la entrada de habilitación el valor indicado por la entrada in se copia en la dirección de salida out.

	Estado	Símbolo	Dirección	Tipo de dato
1		Discharge	E 0.1	BOOL
2		Discharge Valve	A 0.2	BOOL
3		Discharging	A 0.3	BOOL
4		Fill	E 0.0	BOOL
5		Fill Valve	A 0.0	BOOL
6		Filling	A 0.1	BOOL
7		M_Discharge	M 10.1	BOOL
8		M_Fill	M 10.0	BOOL
9		Tanque Digital	OB 1	OB 1
10		Timer	AW 30	WORD

Ilustración 16: Tabla de Variables utilizadas en nuestro programa válvulas de llenado y vaciado siendo sensores y actuadores

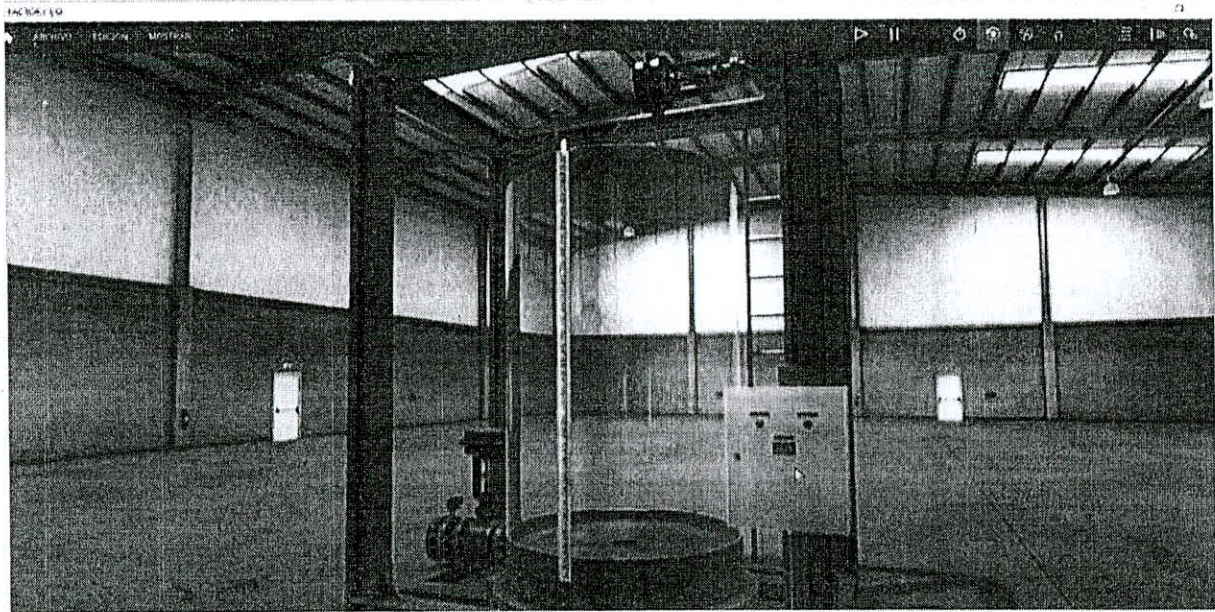


Ilustración 17: Tanque de simulación para llenado y vaciado

- Al iniciar abre en automático el PLC
- Así como la opción para cargar el PLC que se requiere
- El PLC ya está conectado y simulado al proyecto en específico
- Se abre la escena, en este ejemplo es tanque con temporizadores

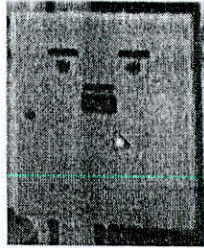


Ilustración 18: Tablero que utilizaremos para llenado y vaciado

- En el tablero se observan dos botones, uno para carga y uno para descarga y un display para ver el tiempo transcurrido
- En la opción mostrar, debemos fijar la tabla de etiquetas: sensores y actuadores

Etiquetas:

- DISCHARGE, botón para descargar (sensor) botón de contacto normal cerrado
- DISCHARGE VALVE, válvula de descarga (actuador)
- DISCHARGING, el led de boton rojo
- FILL, botón para cargar (sensor) contacto normal abierto
- FILL VALVE, actuador para válvula de descarga
- FILLING, led de boton verde
- TIMER, display con dato numerico

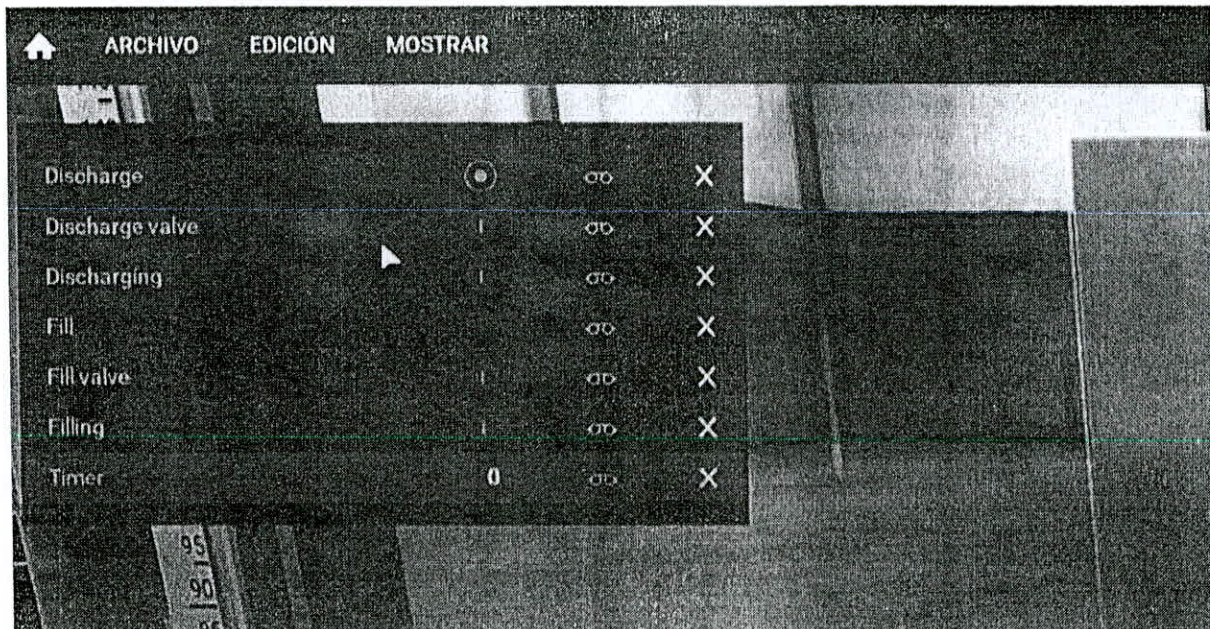


Ilustración 19: Observamos que tenemos nuestras variables determinadas las cuales podemos activar manualmente desde esta pantalla

- En los drivers se elige el PLC
- En configuración de damos por defecto y queden entradas y salidas analógicas
- Se conecta en la primera entrada FILL
- En la segunda entrada DISCHARGE
- Del lado de actuadores se van a conectar el
- Timer en QD30 (salida)
- Fill vale en la primera entrada
- Discharge valve en la segunda entrada
- El filling en la tercera y el último actuador que es discharging
- Se conecta sin problema si todo está bien
- Se puede iniciar la simulación y ya se puede programar
- Para programar en la tabla de variables solo se copian las variables, las entradas y los actuadores respectivamente

Se agregan dos variables auxiliares:

- Tiempo de carga y tiempo de descarga
- Se vuelve a descargar el PLC ya con la programación

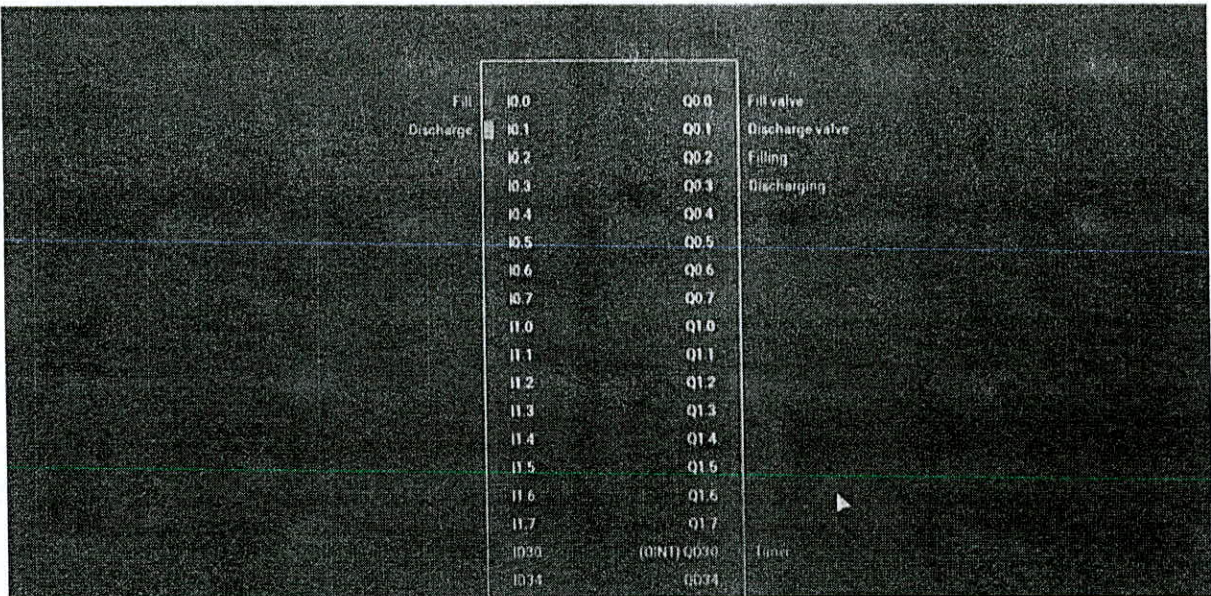


Ilustración 20: En la simulación se colocan las variables de entrada y salida que son nuestros sensores y actuadores se colocan del lado izquierdo nuestros sensores y del lado derecho todos los actuadores.

EJEMPLO STEP7 MicroWIN

Implementación de un semáforo con ayuda del simulador de STEP7 utilizando lenguaje de escalera, para que funcionara nuestro semáforo utilizamos temporizadores para cada color en cada una de sus transiciones.

SALIDAS

VERDE	Q0.0
AMARILLO	Q0.1
ROJO	Q0.2

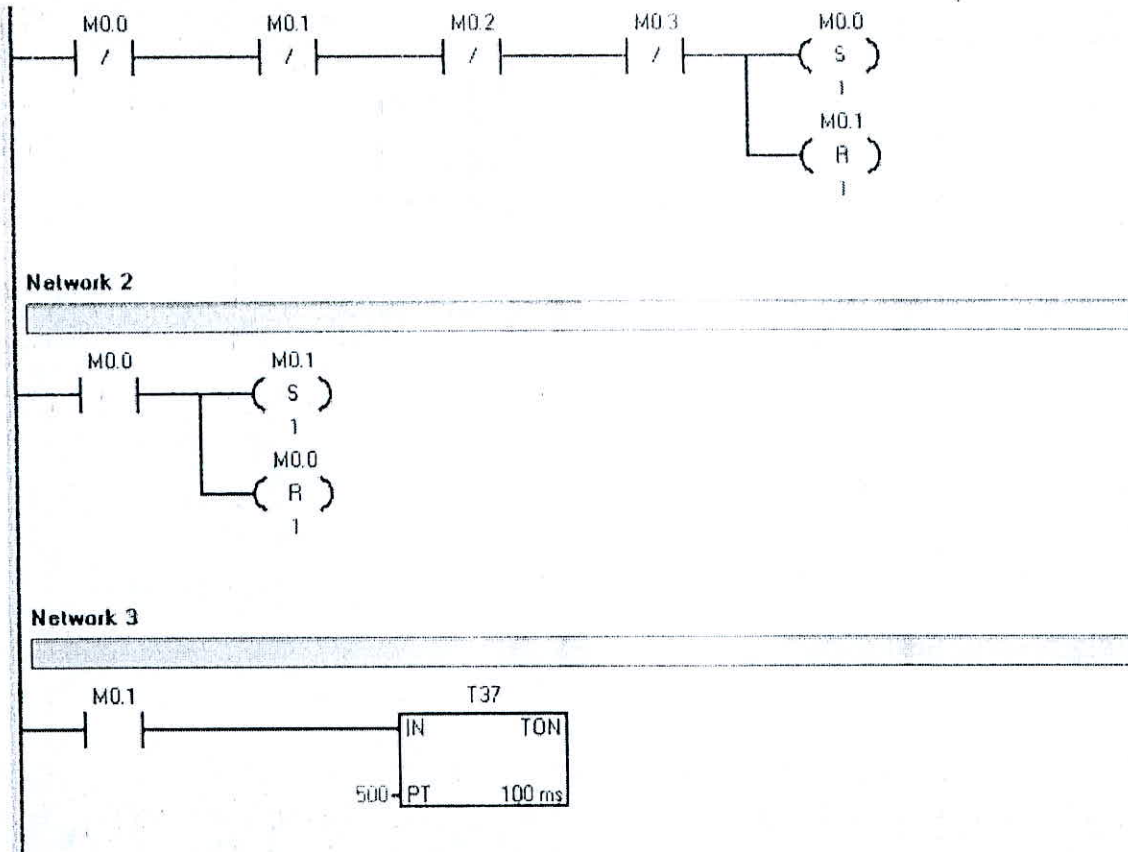


Ilustración 22: Secuencia de inicio del color verde de nuestro semáforo.

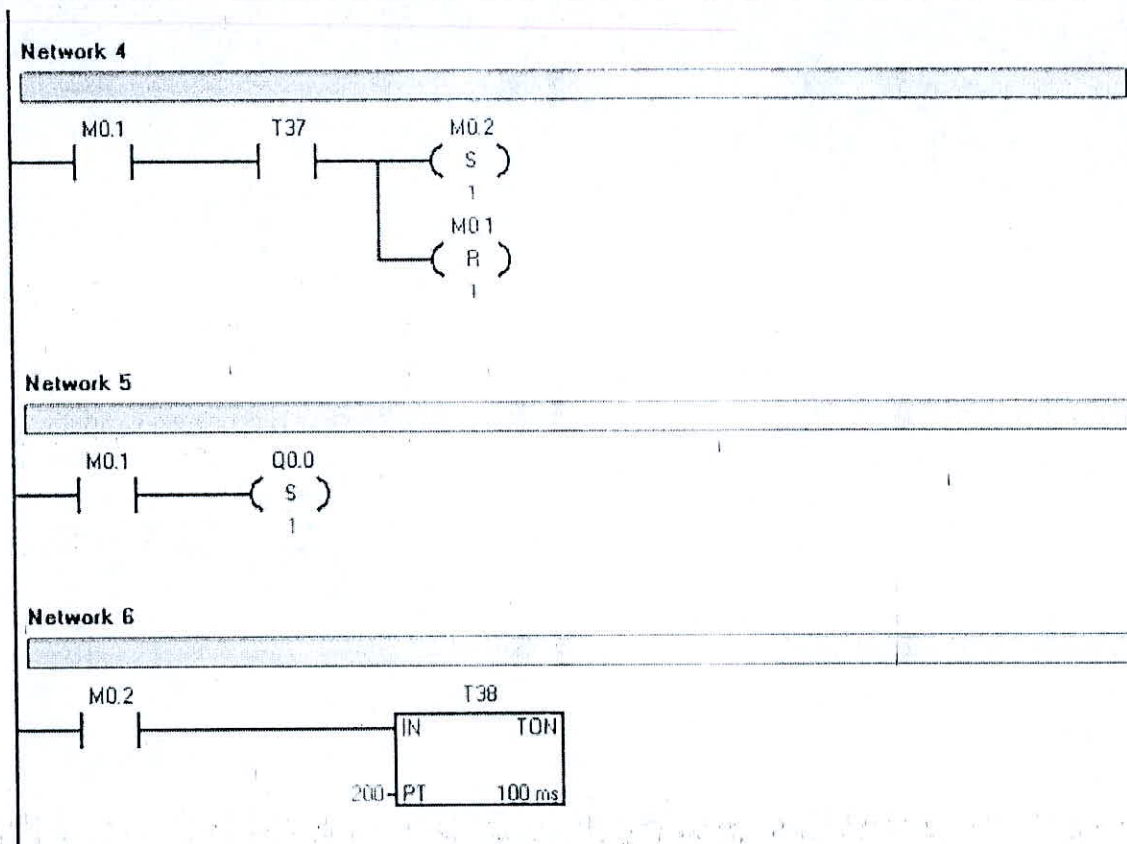


Ilustración 23: Secuencia de inicio del color amarillo de nuestro semáforo. Tiempo de temporización más corto.

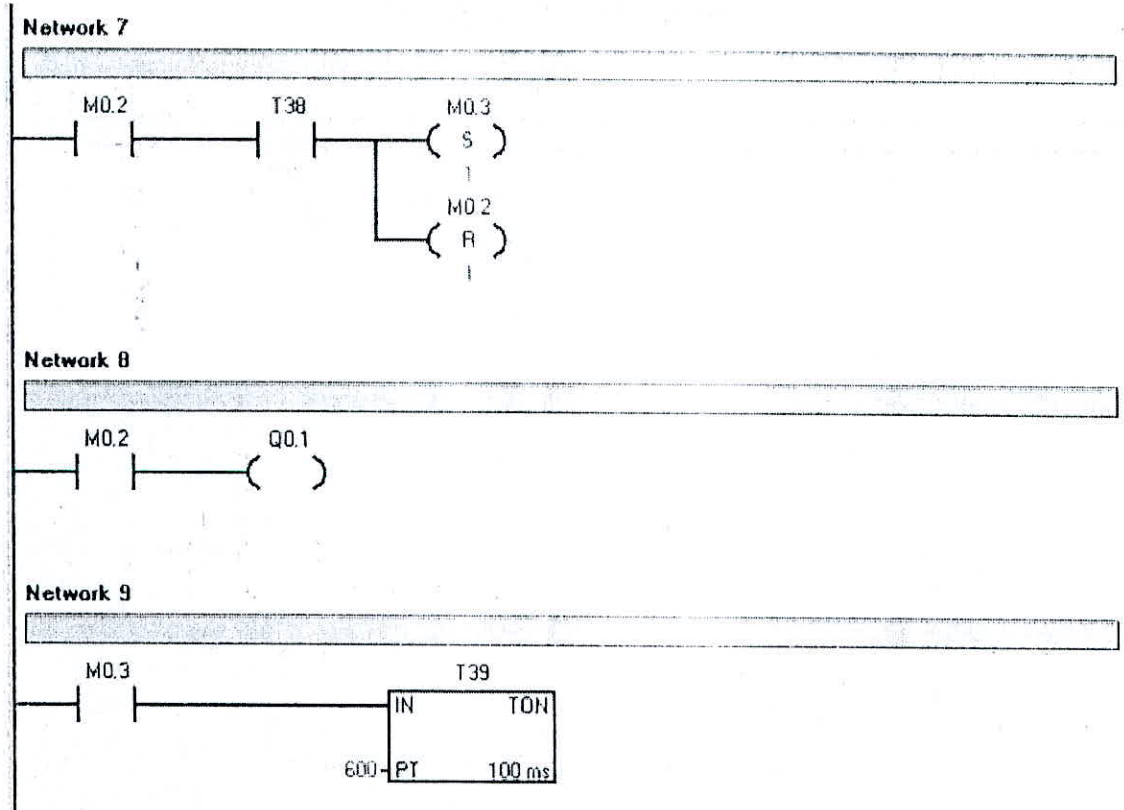


Ilustración 24: Secuencia de inicio del color rojo de nuestro semáforo. Tiempo de temporización más largo.

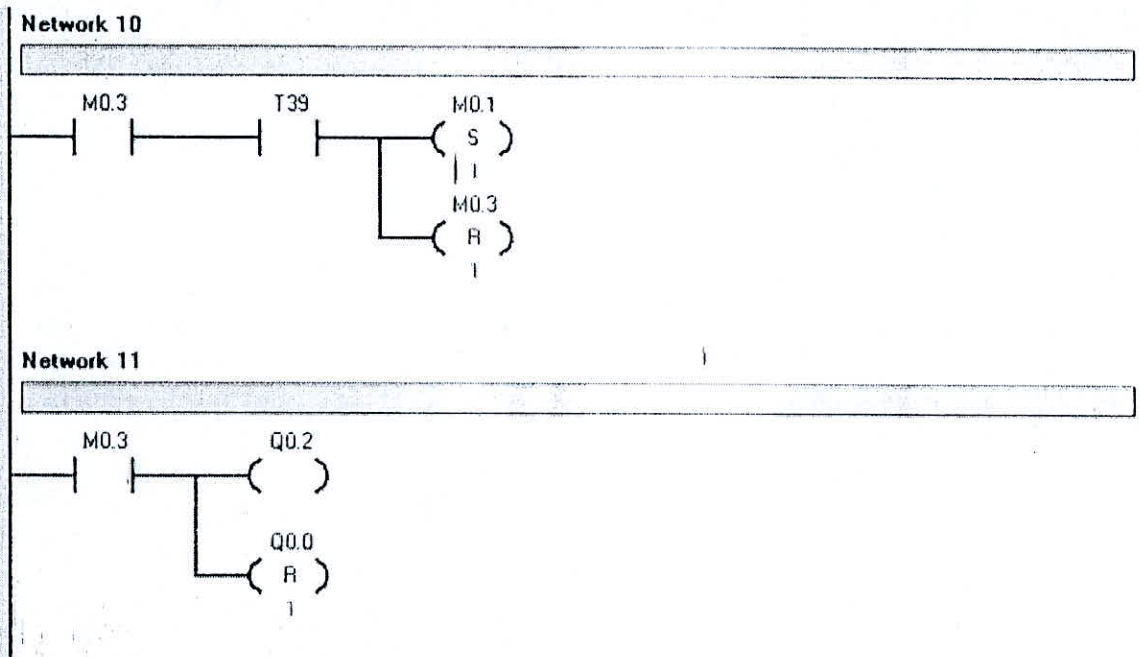


Ilustración 25: Reinicio de la secuencia

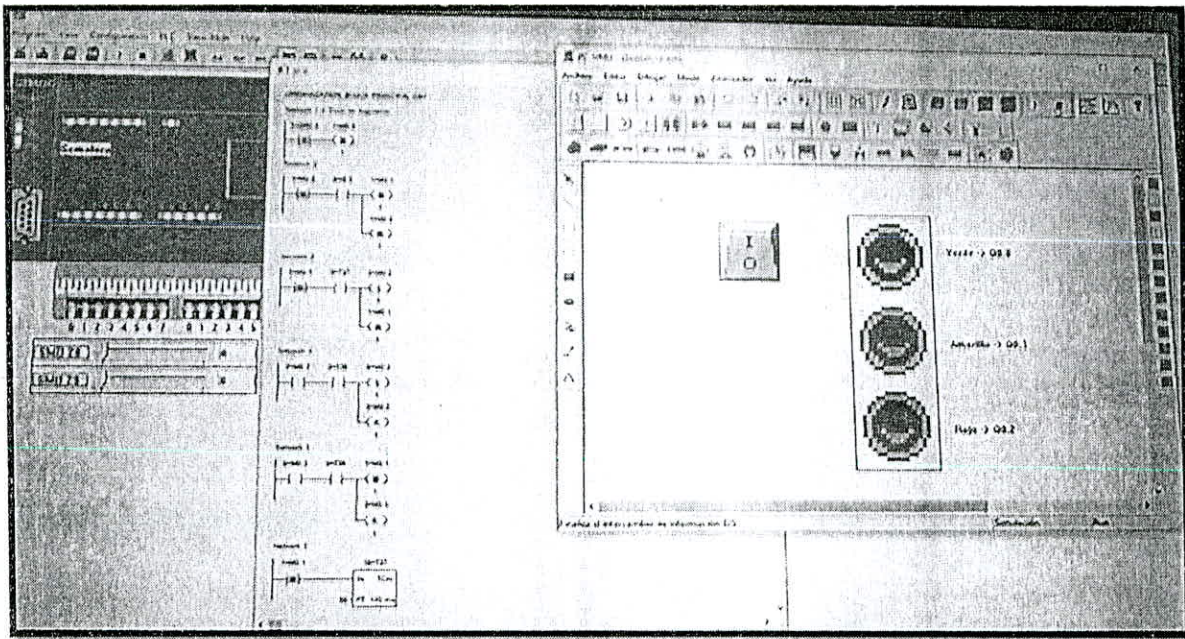


Ilustración 26: Simulación e interfaz de nuestro semáforo

EJEMPLO INTERFAZ HMI TIA PORTAL, SEMAFORO INTELIGENTE

El PLC empleado es el Siemens SIMATIC S7-1200, el cual está compuesto de una CPU, una fuente de alimentación y varios módulos de entrada y salida

El controlador modular SIMATIC S7-1200 es el corazón de la línea de productos Siemens para procesos automáticos elementales, y de mucha precisión, potencia y flexibilidad para el control de una amplia variedad de dispositivos de apoyo a los requerimientos de automatización. El compacto diseño, flexible configuración y potente conjunto de instrucciones se combinan para que el S7-1200 sea la solución ideal para el control de una amplia gama de aplicaciones.

Características

El Simatic S7-1200 ofrece una gran variedad de propiedades técnicas, de las que se pueden destacar las siguientes:

Alta capacidad de procesamiento. Cálculo de 64 bits.

Interfaz Ethernet / PROFINET integrado.

Entradas analógicas integradas.

Bloques de función para control de ejes conforme a PLCopen.

Programación mediante el software STEP 7 Basic v13 para la programación y configuración tanto del S7-1200, así como de los paneles de la gama Simatic Basic Panels.

- A continuación, se describen los diferentes circuitos que se realizaron en TIA Portal
- Comenzaremos viendo las variables que se usaron en el PLC.
- En total se ocuparon 41 variables, de diferentes tipos de datos "Bool e Int",

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...
1	Start	Bool	%M0 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Clock_Byte	Byte	%MB100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Clock_10Hz	Bool	%M100 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Clock_5Hz	Bool	%M100 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Clock_2.5Hz	Bool	%M100 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Clock_2Hz	Bool	%M100 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Clock_1.25Hz	Bool	%M100 4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Clock_1Hz	Bool	%M100 5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Clock_0.625Hz	Bool	%M100 6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Clock_0.5Hz	Bool	%M100 7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Semaforo 1	Int	%MW2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Sec-Cambio	Bool	%M0 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Semaforo 2	Int	%MW4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Sec-A2	Bool	%M0 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Run	Bool	%M0 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
16	Reset	Bool	%M0 4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
17	Sem1 On	Bool	%M0 5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
18	Sem2 On	Bool	%M0 6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
19	Sem1-Listo	Bool	%M0 7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
20	Inicio S1	Bool	%M1 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
21	Inicio S2	Bool	%M1 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
22	Sec-V2	Bool	%M1 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
23	V2	Bool	%M1 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
24	A2	Bool	%M1 4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
25	A1	Bool	%M1 5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
26	V1	Bool	%M1 6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
27	A3	Bool	%M1 7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
28	V3	Bool	%M6 0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
29	Sec-A1	Bool	%M6 1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
30	Sec-V1	Bool	%M6 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 27: Tabla de Datos 1

31		Semaforo 3	Byte	%MB7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
32		Inicio S3	Bool	%MB 3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
33		Sec-A3	Bool	%MB 4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
34		Sec-V3	Bool	%MB 5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
35		varA	Bool	%MB 6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
36		varB	Bool	%MB 7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
37		varC	Bool	%MB 0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
38		Rutina Inicio	Bool	%MB 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
39		Max A	Int	%MW10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
40		Max B	Int	%MW12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
41		Max C	Int	%MW14	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
42		-Agregar-			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 28: Tabla de Datos 2

Seguiremos en el Main, y las diferentes secciones, en total se ocuparon un total de 7 segmentos, en esa forma divididos.

- ▶ **Segmento 1:** Inicio
- ▶ **Segmento 2:** Logica
- ▶ **Segmento 3:** Semaforo 1
- ▶ **Segmento 4:** Semaforo 2
- ▶ **Segmento 5:** Semaforo 3
- ▶ **Segmento 6:** Resta de Carros
- ▶ **Segmento 7:** Reset

Ilustración 29: Total de segmentos

Segmento Inicio

Diagrama de la rutina de inicio en lenguaje de escalera parte 1

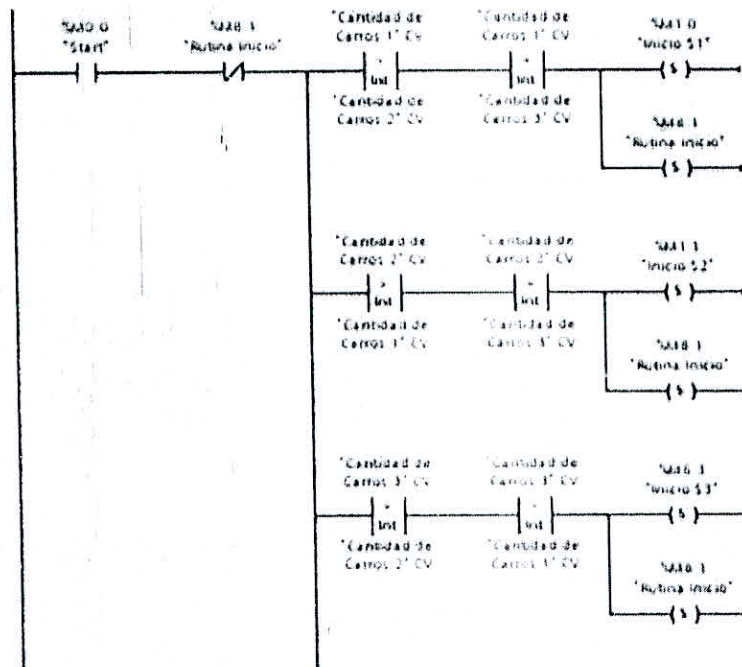


Ilustración 30: Segmento Inicio con los comparadores

Parte 2 rutina de inicio

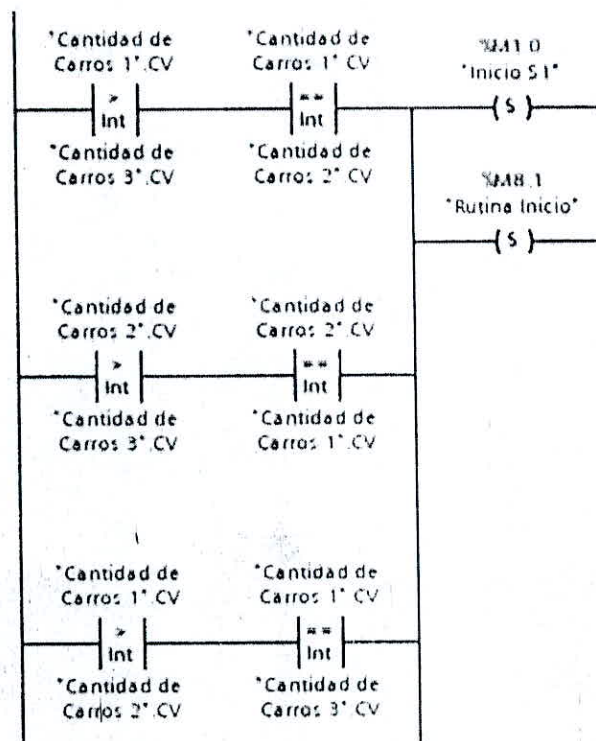


Ilustración 31: Parte de comparación entre semáforos para el inicio correcto.

Parte 3 rutina de inicio

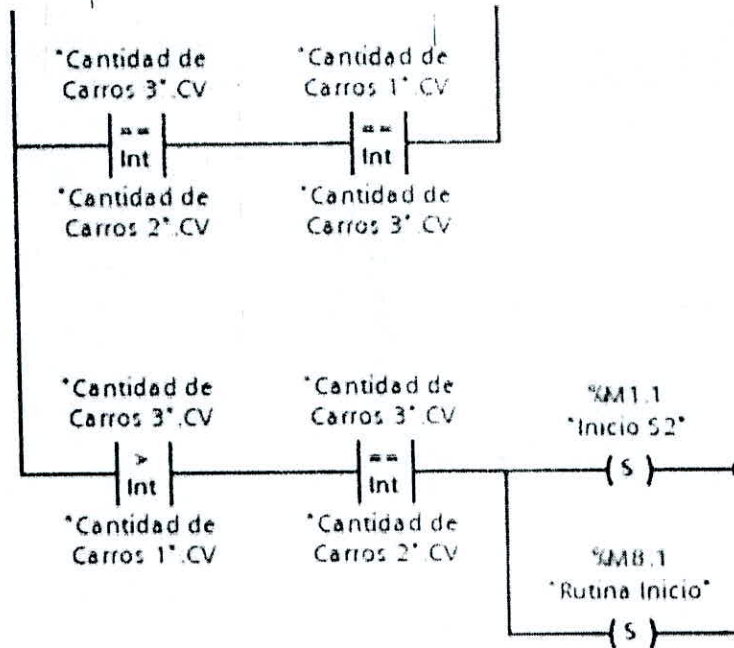


Ilustración 32: Parte de comparación y reset de los semáforos

Segmento Lógica

Lógica semáforo parte 1

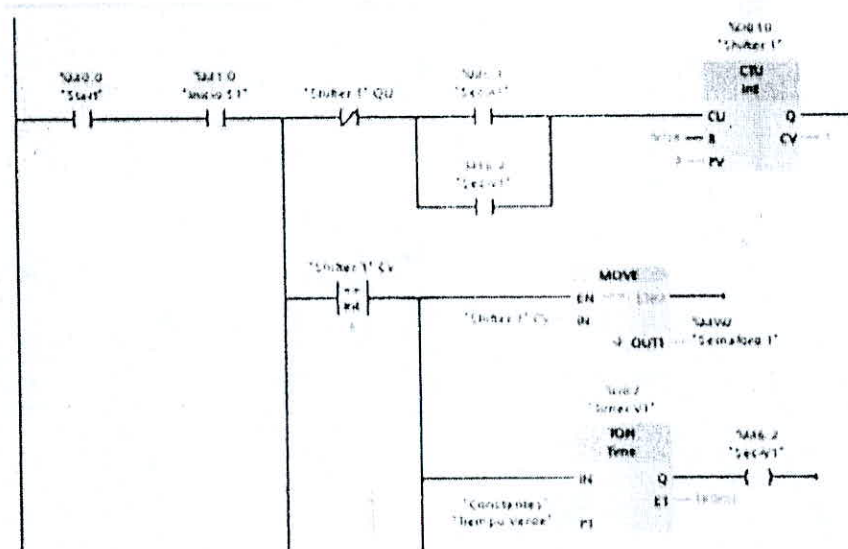


Ilustración 33: Parte inteligente del PLC para la comparación de semáforos dependiendo de los autos.

Lógica semáforo parte 2

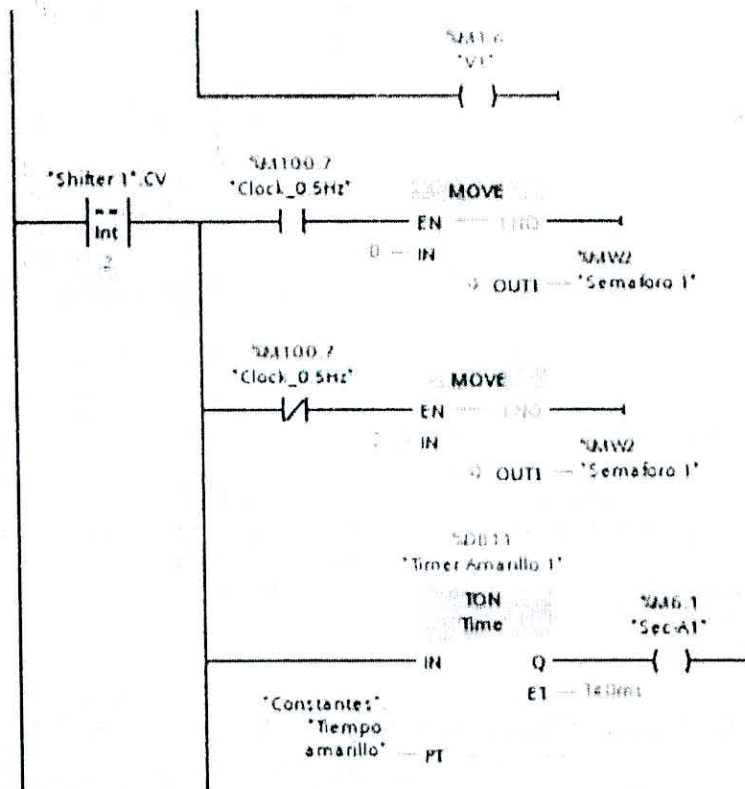


Ilustración 34: Parte inteligente del PLC para la comparación de semáforos dependiendo de autos PARTE B

Lógica semáforo parte 3

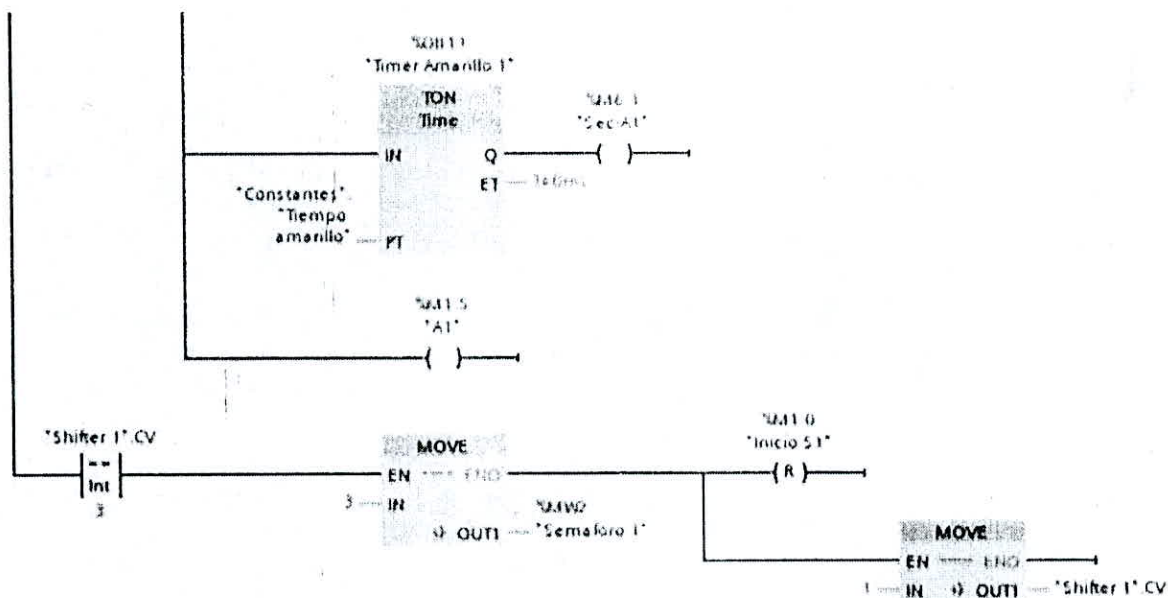


Ilustración 35: Parte inteligente del PLC para la comparación de semáforos dependiendo de autos PARTE C

Lógica de la resta de los carros parte 1

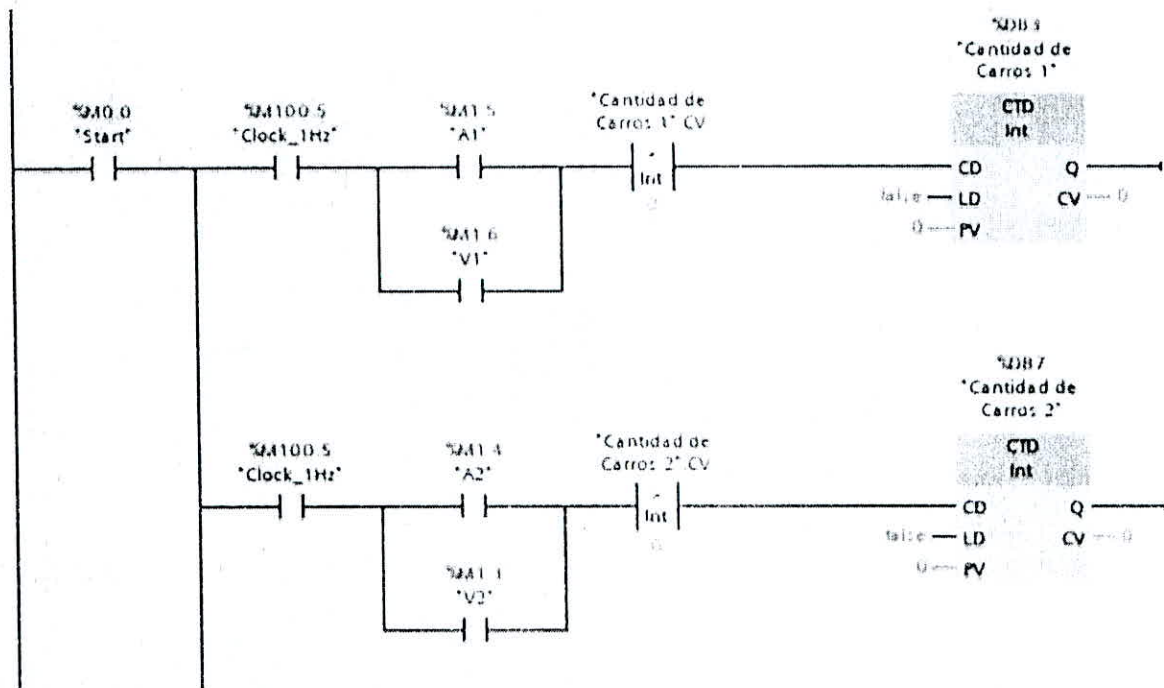


Ilustración 36: Resta de autos que pasen por semáforos con ayuda del reloj del plc.

Lógica de la resta de los carros parte 2

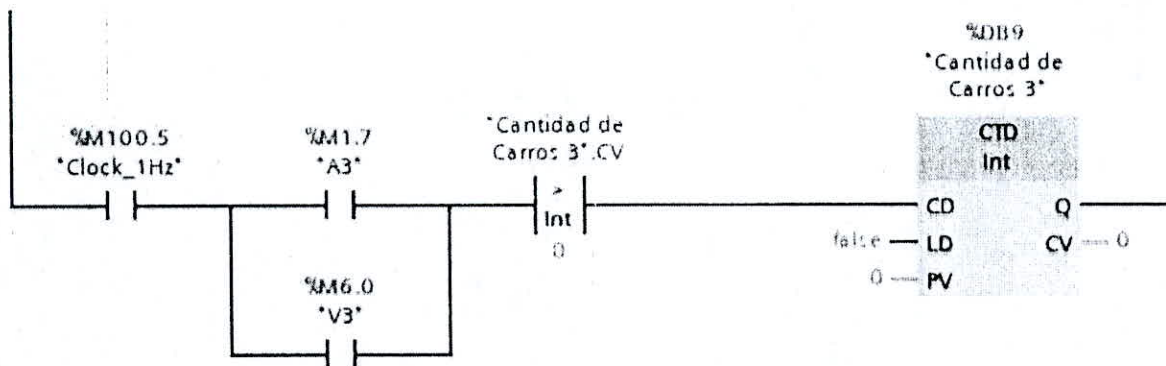


Ilustración 37: Resta de autos que pasen por semáforos PARTE B

Lógica para el reinicio de la secuencia (reset)

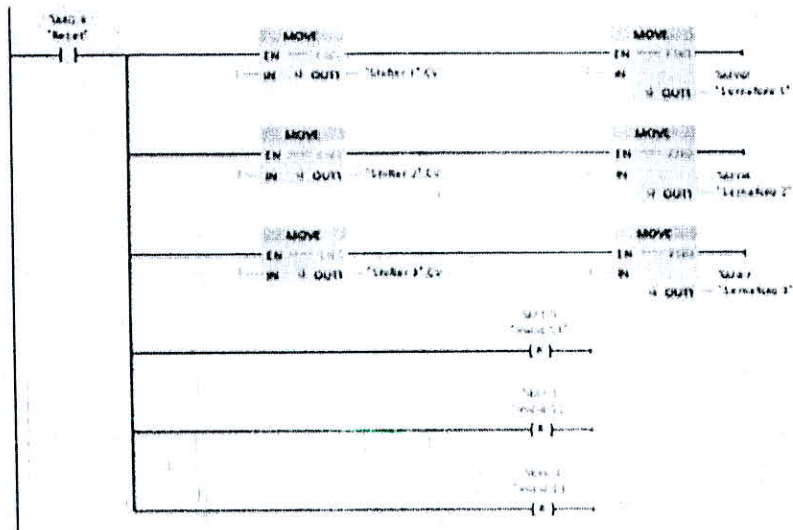


Ilustración 38: Reset, se mandan todos los semáforos en rojo asignando el valor de 3.

Lógica inteligencia semáforo parte 1

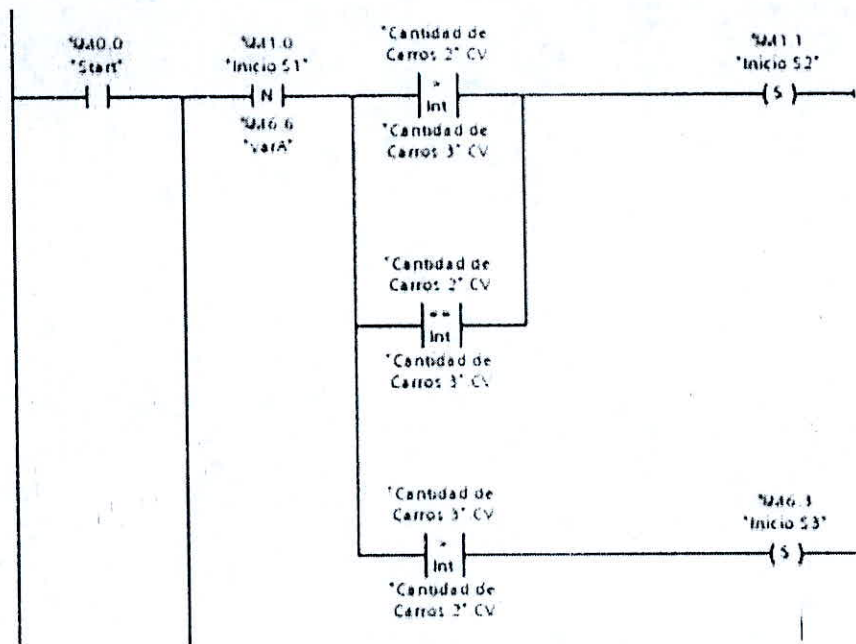


Ilustración 39: Hace evaluación de los contadores de los coches que hay en cada semáforo.

Lógica inteligencia semáforo parte 2

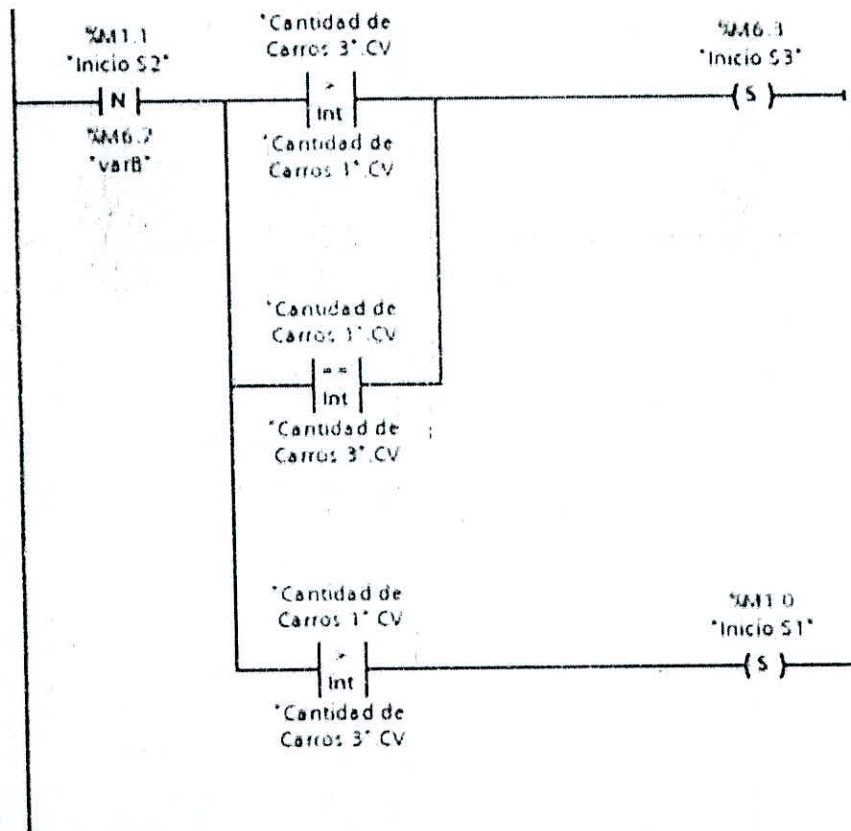


Ilustración 40: Hace evaluación de los contadores de los coches que hay en cada semáforo PARTE B

Lógica inteligencia semáforo parte 3

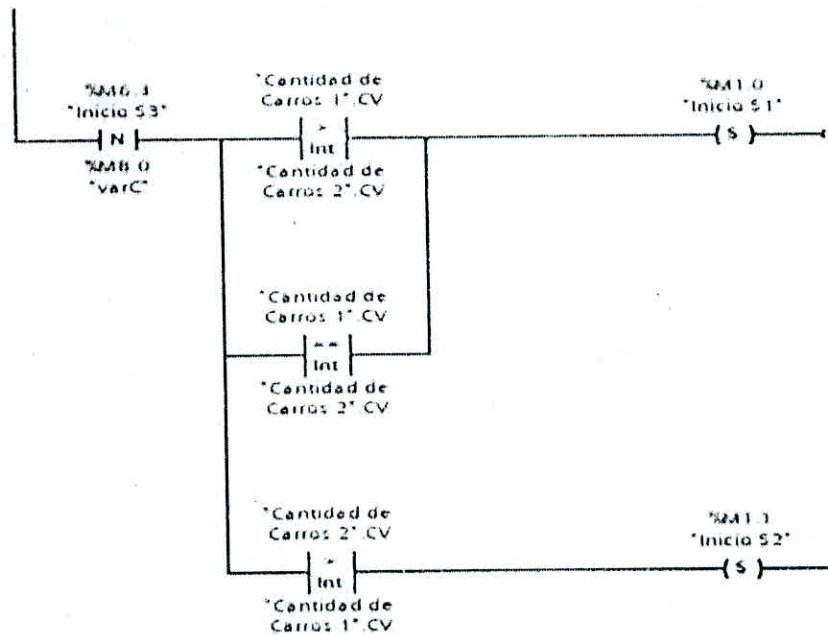


Ilustración 41: Hace evaluación de los contadores de los coches que hay en cada semáforo PARTE C

RESULTADO FINAL

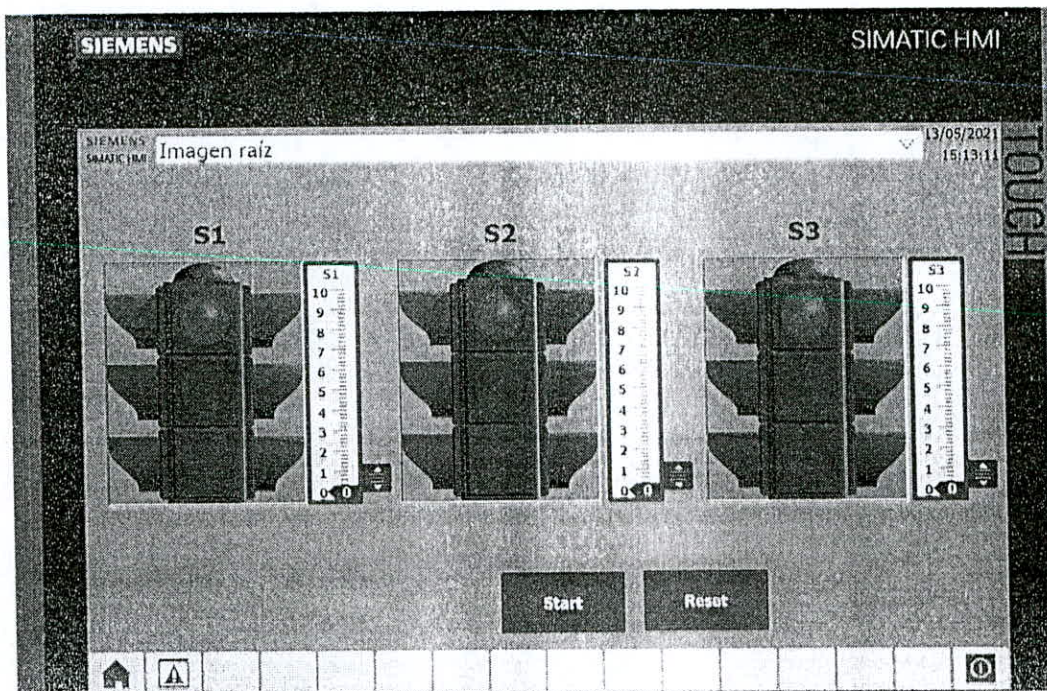


Ilustración 42: Interfaz final de semáforos con su respectivo contador del lado derecho.

FUENTES

Barrientos, A. and Gambao, E., *Sistemas de Producción Automatizados*, Universidad Politécnica de Madrid, 2014.

Cabañero, C., Giménez, C., and Lusa, A., *Gestión de La Producción*, Universitat Oberta de Catalunya, 2013.

Mandado, E., Marcos, J., Fernández, C. and Anesto, J. I., *AUTÓMATAS Programables Y Sistemas de Automatización*, 2009.

Platero, C., *Apuntes de Regulación Automática*, 2012.

Siemens, *Esquema de Contactos (Kop) Para S7-300 Y S7-400*, Siemens, 2004a.

Siemens, *Lista de Instrucciones (AwI) Para S7-300 Y S7-400*, Siemens, 2004b.