Capítulo 3

3.1. Automatización con PLC

El controlador lógico programable entablara la relación de los periféricos de entrada con la unidad lógica para generar una salida y dar respuesta a los periféricos de salida mientras esta no genera una salida el circuito se retroalimentara con respecto al tiempo. En el siguiente diagrama de bloques se muestra la construcción de este circuito y cuáles serán los periféricos de entrada y los de salida.

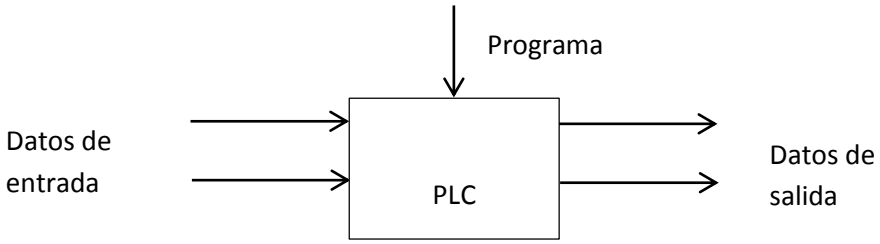


Imagen No 8 *Controlador Lógico Programable*

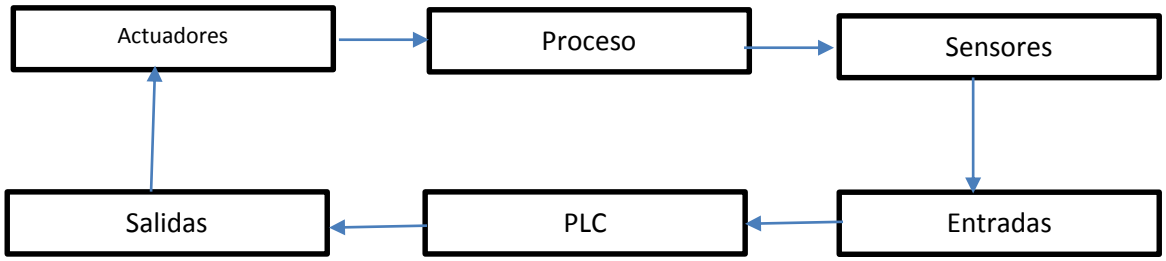


Imagen No 9 Configuración del sistema PLC

Descripción de las áreas funcionales

Área funcional	Equipos correspondientes
Área funcional A	= Barrena = Sensor de presión
Área funcional B	= Bomba de lodo de perforación = Sensor de caudal entrante
Área funcional C	= Bomba de lodo de perforación 2 = Sensor de caudal saliente

Tabla No 5 Áreas funcionales

3.1.1. Módulos del PLC

1.- Modulo: PS 307 5A — Referencia: 6ES7 307-1EA00-0AA0 Fuente de alimentación carga AC 120/230V/5ª

2.- Modulo: CPUC — Memoria de trabajo 32KB; 0,2ms/1000 instrucciones; DI10/DO6 integradas; 2 salidas de impulso (2,5kHz); 2 canales de contaje y medición con encoders incrementales 24V (10kHz); interfaz MPI; configuración en una fila hasta 8 módulos; firmware V2.6; también disponible como módulo SIPLUS con la referencia 6ES7 312-5BE03-0AB0

3.- Modulo de señales — Los modulos de señales adaptan los distintos niveles de las señales de proceso al S7-312C

4.- Cable de bus PROFIBUS

5.- Cable para conectar una programadora

La imagen siguiente muestra el orden y la configuración de los módulos del sistema PLC.

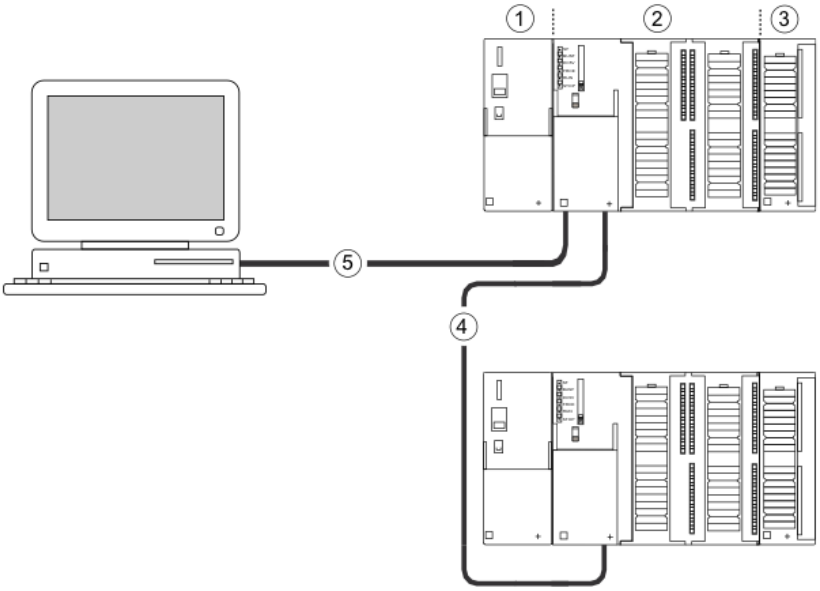


Imagen No 10 Orden y configuración de los módulos del sistema

3.1.2. Variables de control

1. Sensores de entrada: un sensor de caudal y un sensor de presión.
2. Los sensores arrojan la información del proceso.
3. PIC realiza los procesos para general las salidas correspondientes.
4. Teniendo las salidas, se trasmite la señal de control a los actuadores que son los preventores RAM.
5. Por último se realiza el proceso.

Al obtener los sensores de entrada se obtuvieron las variables y constantes para poder generar el código DL que se muestra más adelante.

Las variables y constantes que se pretenden manejar manejan para el sensor de caudal son:

1. Variable: Volumen-1 = Volumen del lodo de perforación de entrada
2. Variable: Volumen-2 = Volumen del lodo de perforación de salida
3. Variable: Volumen-3 = Volumen del flujo invasor
4. Constante: Volumen-4 = Volumen de flujo invasor máximo aceptable

Las variables que se manejan para el sensor de presión son:

1. Variable: Presión-1 = Presión que se genera a la hora de perforar el pozo
2. Constante: Presión-2 = Presión máxima aceptable en perforación de pozos

Las señales de proceso activarán los siguientes preventores

1. Preventor-1 = Preventor anular o de ariete para tubería
2. Preventor-2 = Preventor de ariete ciego
3. Preventor-3 = Preventor de corte

Las variables que se manejan dependiendo de qué giro tengan si son de entrada, salida o marcas tiene un bite, esto quiere decir que tiene 8 posiciones de entrada, salida o marca, el número de bit va de 0 a 7 y cuando se llega al 7 la dirección aumenta de uno en uno.

Bloques de organización

OB = Bloque de organización

DB = Bloque de datos

FB = Bloque de funciones

FC = Función

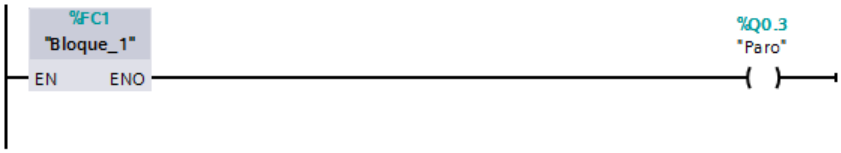


Imagen No 11 Bloque organizacional

3.1.2.1. Periféricos de entrada

Para poder hacer el proceso de control del sistema preventor de reventones por medio del PLC se tomarán en cuenta dos variables que estarán proporcionadas por los periféricos de entrada, los cuales consisten en un sensor de caudal y un sensor de presión para la activación pertinente del sistema BOP.

3.1.2.1.1. Presión

Para la obtención de la variable de presión, utilizaremos el medidor barométrico que se encuentra en la cabina de perforación, esta presión es la registrada en el arreglo compuesto por la tubería de perforación y la barrena. Se tomarán los valores del ejemplo de RAM de corte ya que este sensor será la condición de este ariete, de esta manera el escalamiento será de 1:4 para simulación, el valor máximo a registrar será $Presion_{Maxima} = 10.19\ bar$, de esta manera se condicionara la presión registrada cada $t = 60\ seg$.

Imagen No 12 Sensor de presión FESTO

Para poder desarrollar este proceso de adquisición de datos por medio de un sensor de presión, se recreara el circuito en escala y se simulara el cierre del preventor utilizando dos actuadores de doble efecto y con las conversiones correspondientes para este proceso de periférico de entrada

3.1.2.2. Velocidad de fluido

Otra de las variables a controlar es la velocidad del fluido o caudal con la cual registraremos el volumen del fluido, para hacer una retroalimentación a lazo cerrado y poder comparar la cantidad de fluido; esto deberá ir cambiando conforme va cambiando la profundidad de la perforación.

De la presa de lodo de perforación pasa al pozo con una fluidez de $1.8334\frac{bbl}{mi} * \frac{158.987294928\ l}{bbl} = 291.492\frac{l}{mi}$. Esta es la fluidez más utilizada en la perforación de pozos petroleros; la escalaremos de 1 a 50 para poderla simular con parámetros de caudal para una fluidez de $5.829\frac{l}{mi}$ y el sensor que utilizado será el caudalímetro YF-S402.

Imagen No 13 Caudalímetro

La ganancia de fluido invasor permisible es $0.4\frac{bbl}{mi} * \frac{158.987294928\ l}{bbl} = 63.5949\frac{l}{mi}$ con el escalamiento de 1:50 $1.2719\frac{l}{mi}$ que se puede registrar con el caudalímetro; con estos parámetros se condicionan el RAM ciego y de tubería.

3.2. Creación del Proyecto Automatización BOP

Se llevó acabo la creación del proyecto Automatización BOP en dos entornos diferentes, CoDeSys y TIA para poder comparar la optimización de cada uno y obtener los mejores resultados.

3.2.1. Simulación CoDeSys

La creación del proyecto en el entorno de CoDeSys se generó con las siguientes variables y condiciones para poder ser manipulado. La variable PRESION se activa cuando la $P_{max} > P_T$; para activar esta variable se manda un 1 lógico al PLC ya que es una toma de decisión digital.

Nota: Se eligió el lenguaje LD o Ladder ya que es un lenguaje genérico que todos los PLCs reconocen sin importar tamaño ni marca.

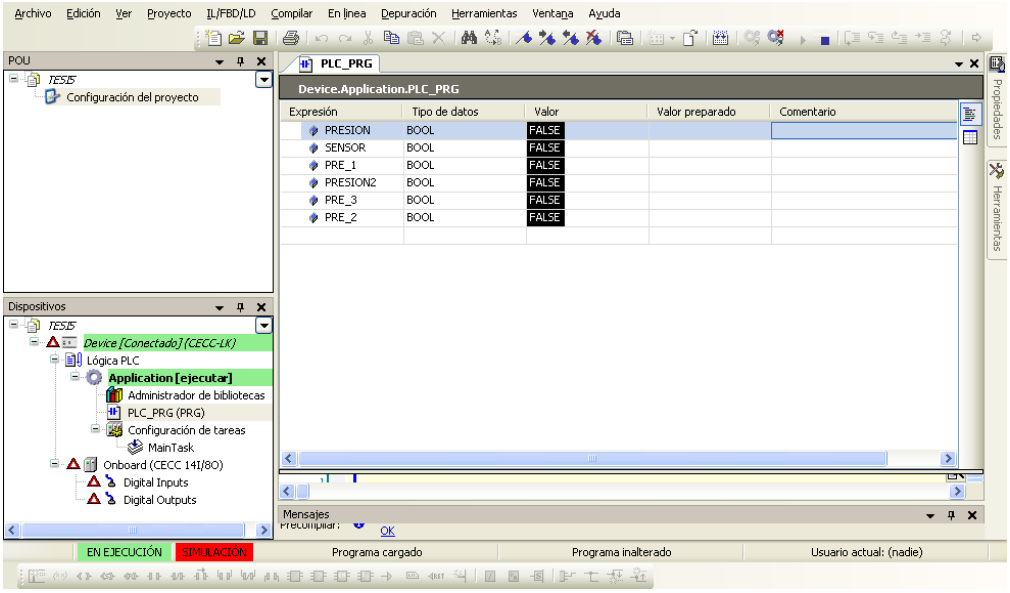


Imagen No 14 Variables del programa

El programa se inicializa con todas sus variables en 0. En el diagrama LD se muestra en el circuito la parte iluminada en azul que indica que está en activación la

condición del preventor uno; si el sensor está inactivo, da como respuesta 1 ya que así se declaró esta variable en este escalón.

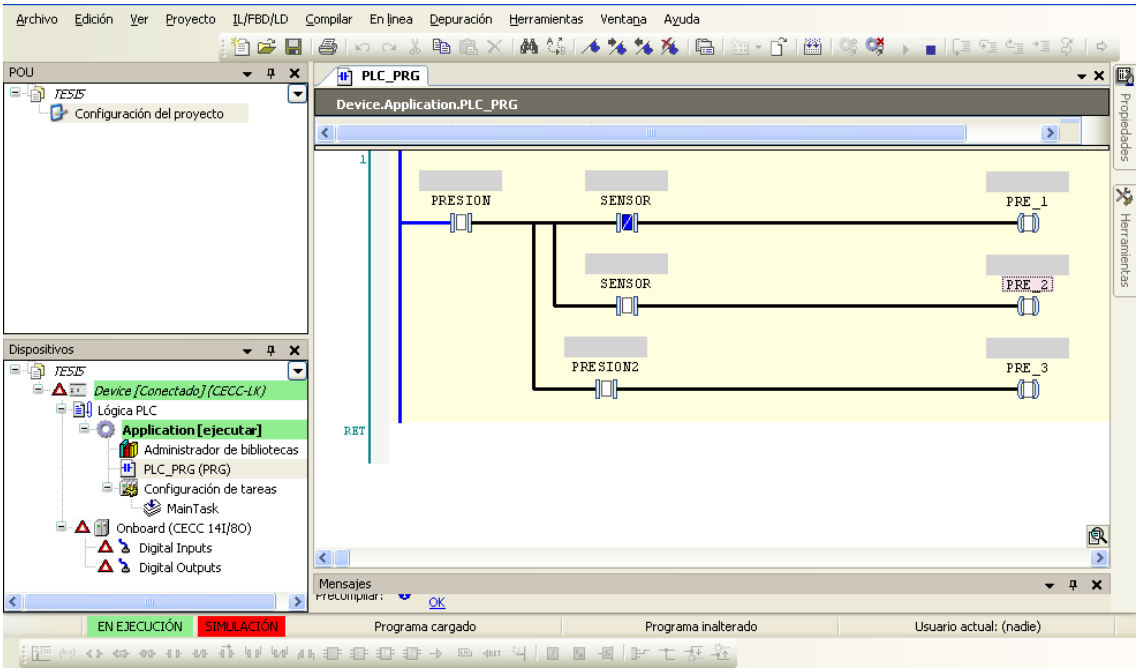


Imagen No 15 Código LD

Cuando se lleva a cabo la activación de la variable de entrada PRESION y como la variable SENSOR está desactivada, se activa la variable de salida PRE_1, la cual activa al actuador de cilindro de doble efecto.

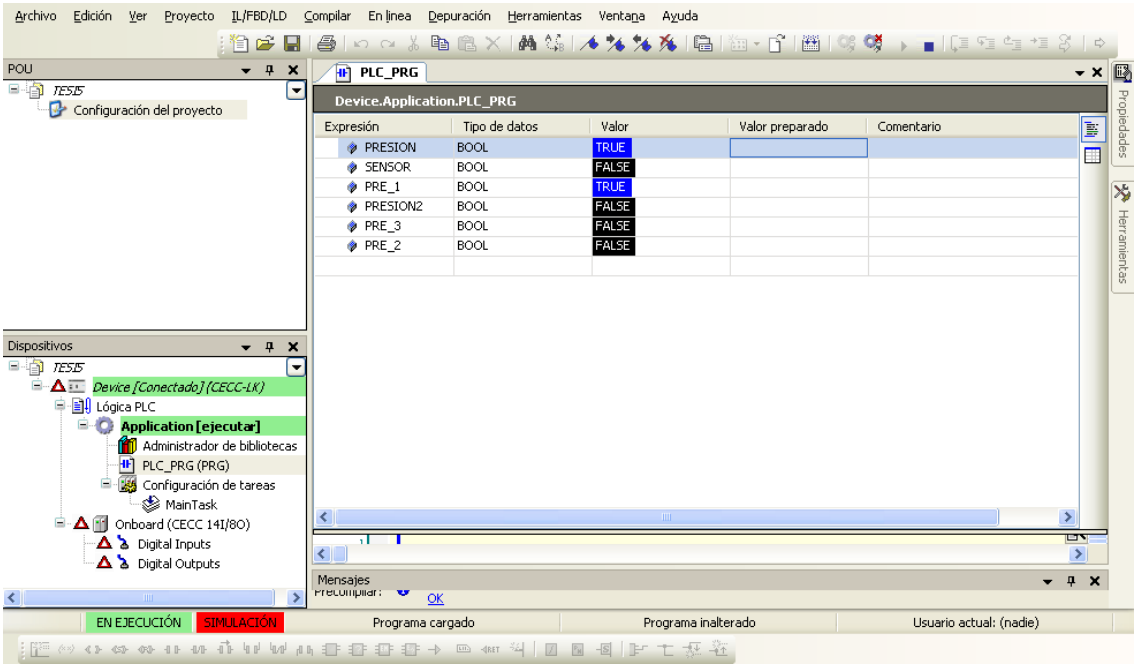


Imagen No 16 Activación variable PRESION

La activación de la variable PRESION genera una reacción en el programa y se puede notar mejor en el lenguaje LD; la activación se puede apreciar con la línea azul remarcada que indica el acceso del fluido activando el pistón que simula al primer ariete del arreglo ya que dicho arreglo consta de 3 pares de arietes.

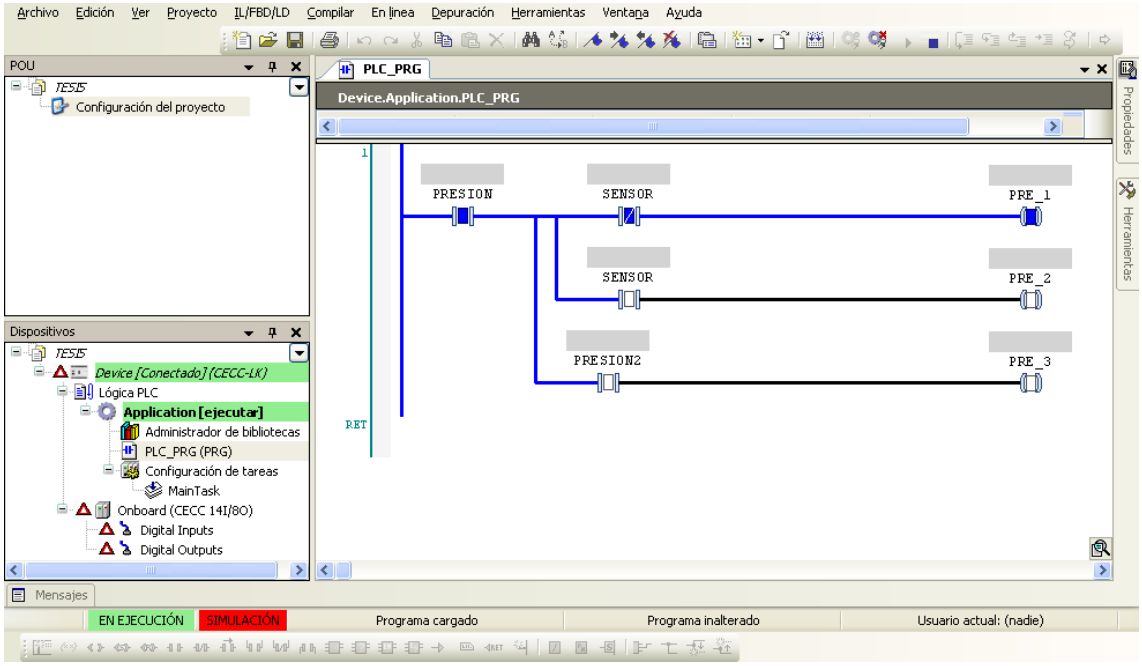
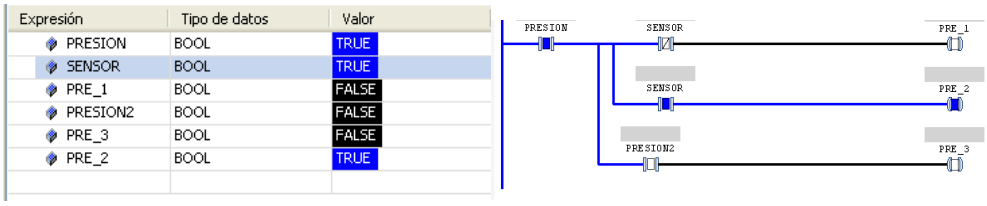


Imagen No 17 Prueba

Al realizar la simulación en este entorno se encontraron limitaciones para poder simular cómo realizar un entorno HIM y una conexión OPC, por esos motivos esta simulación solo se tomara como guía para hacer el circuito de electroneumática; en la siguientes imágenes se puede ver el proceso.



Expresión	Tipo de datos	Valor
PRESION	BOOL	TRUE
SENSOR	BOOL	TRUE
PRE_1	BOOL	FALSE
PRESION2	BOOL	FALSE
PRE_3	BOOL	FALSE
PRE_2	BOOL	TRUE

Imagen No 18 Proceso

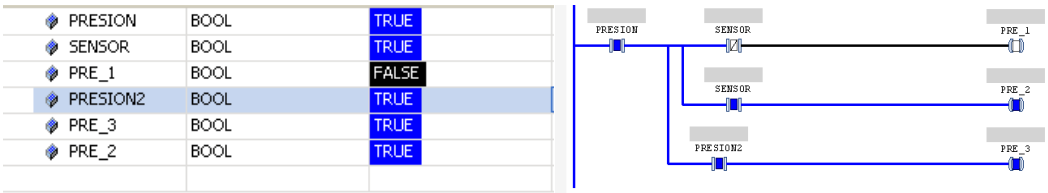


Imagen No 19 Proceso completo

Con esto se da la terminación de esta simulación en CoDeSys; dados las limitante que se presentaron, esto se documenta con el propósito de actualizar los programas que se utilizan en las aulas.

3.2.2. Simulacion Electroneumatica

El procedimiento de simulacion que se lleva acabo está basado en un circuito electroneumatico, para poder construir el sistema PLC y obtener las variables a controlar, de igual forma se obtendrán las marcas que se manerajan en el programa de escalera LD; las variables de entrada se difiniran posteriormente. El software utilizado para la representacion de los RAMs que conforman el BOP es FluidSIM neumática version 4.2, este metodo de simulacion se tomó de igual manera por su enlazamiento por medio de OPC para entrelazar los programas y tener una respuesta simulada con actuadores.

EL circuito se divide en tres segmentos para poder tener un mejor control de los sensores que fungen de perifericos de entrada y las respuestas arrojadas a los perifericos de salida, se muestra a continuacion los segmentos y los componentes de cada uno.

1.- RAM de ariete: cuando hay precencia de tuberia.

Marca	Denominación de componentes
	Fuente de tensión (0V)
	Fuente de tensión (24V)
K1	Obturador
1Y	Solenoide de válvula
K1	Relé
SENSOR_1	Pulsador (Obturador)
	Válvula de 2/n vías
	Válvula de 2/n vías
	Fuente de aire comprimido
1A1	Cilindro de simple efecto
1A1	Cilindro de simple efecto

Tabla No 6 Componentes activacion RAM de ariete

2.- RAM ciego: cuando no hay tuberia ni herramienta dentro del pozo.

Marca	Denominación de componentes
K2	Obturador
2Y	Solenoide de válvula
K2	Relé
SENSOR_2	Pulsador (Obturador)
	Fuente de tensión (0V)
	Fuente de tensión (24V)
	Válvula de 2/n vías
	Válvula de 2/n vías
1A1	Cilindro de simple efecto
1A1	Cilindro de simple efecto

Tabla No 7 Componentes activacion RAM de ciego

3.- RAM de corte: no importa si hay herramienta o tubería, la presión que se registra es muy alta y el pozo se cierra completamente.

Marca	Denominación de componentes
	Fuente de tensión (0V)
	Fuente de tensión (24V)
K3	Obturador
3Y	Solenoide de válvula
K3	Relé
SENSOR_3	Pulsador (Obturador)
1A1	Cilindro de simple efecto
1A1	Cilindro de simple efecto
	Válvula de 2/n vías
	Válvula de 2/n vías
	Fuente de aire comprimido

Tabla No 8 Componentes activacion RAM de ciego

En la imagen 20 se muestra la configuración completa del BOP y sus tres pares de arietes que lo componenen, como la unión de todos sus complementos, pero es visible que los preventores están alimentados por la misma bomba, pero accionados completamente en formas diferentes con el fin de controlarlos de la manera mas óptima; de igual manera se implementa el PID para corregir los errores.

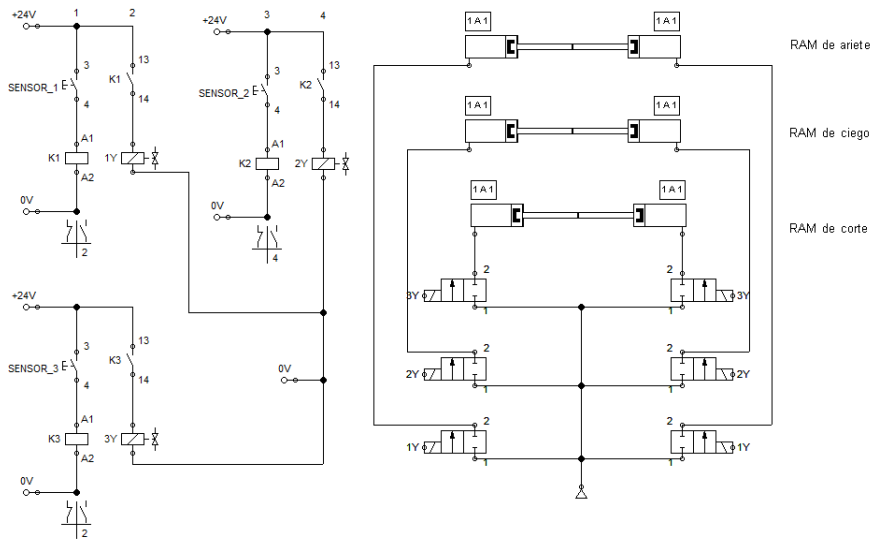


Imagen No 20 Circuito BOP accionado electro-neumáticamente

En la imagen 21 se muestra el primer segmento activado que consiste en el RAM de ariete y se activa cuando el volumen registrado por el caudalímetro de salida es mayor que el volumen de entrada, teniendo en cuenta una constante que se resta en el volumen de salida, esta constante consiste en el valor de extracción del material a la hora de la perforación y está condicionada por el diámetro de la barrena, el tiempo de perforación y la velocidad.

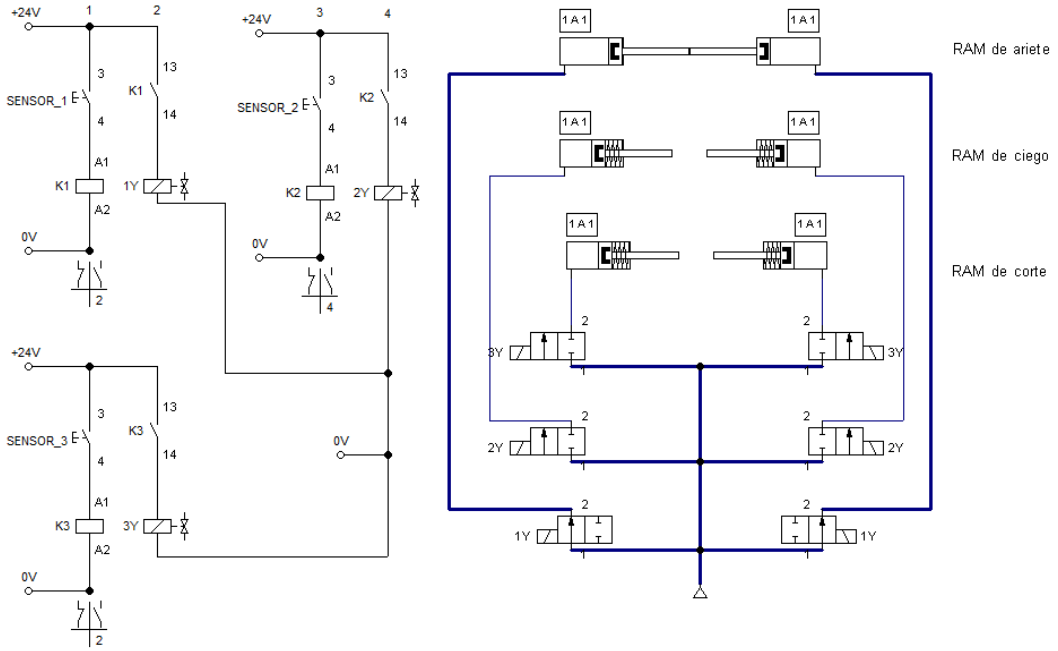


Imagen No 21 Activación del RAM de ariete

En la imagen 22 se activa el ariete de ciego si es que el sensor de tubería está en su estado 0, indicando que no hay tubería ni herramienta y las condiciones que rigen volumétricamente también rigen a este segmento.

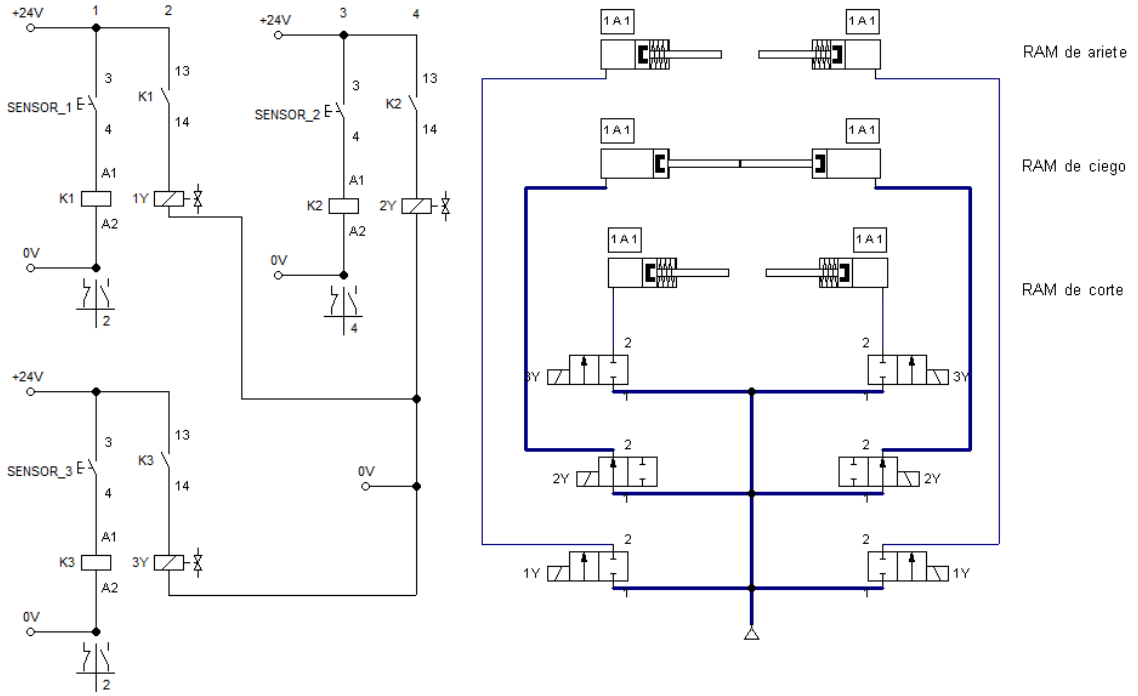


Imagen No 22 Activación del RAM ciego

El último segmento que se puede ver en la imagen 23 es activado si existe una presión contraria mayor a la que puede resistir el pozo; dado esta circunstancia se tiene que cortar la tubería o la herramienta que esté en el interior del pozo, ya que el riesgo es mayor a la de los segmentos controlados volumétricamente.

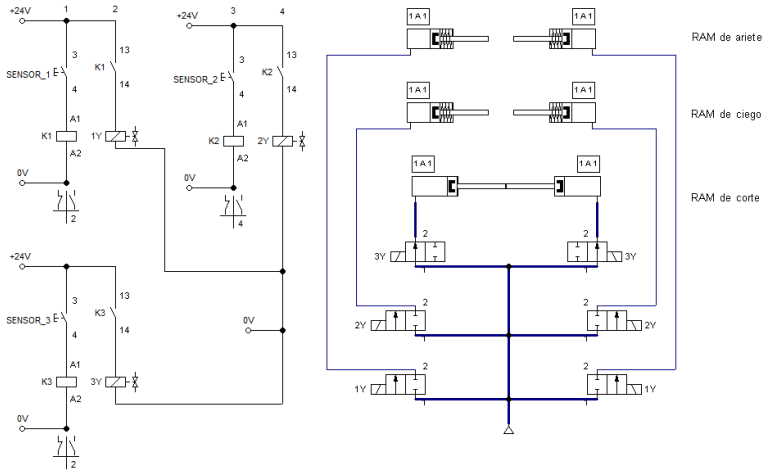


Imagen No 23 Activación del RAM de corte

Obtenido el circuito controlado electro-neumáticamente se obtienen las variables globales que van a regir a el programa como periféricos de entrada o de salida o marcas que funjan como memorias o reset, éstas se explicarán conforme se vaya viendo el código y su escalamiento en un caso en el área laboral. Este proceso es de simulación, por ello las direcciones que se manejan comenzaran todas de 0.0 hasta 0.7, ya sean datos de entrada, de salida o marcas.

Nombre	Tipo de dato	Dirección	Comentario
Encender	BOOL	Entrada 0.0	Activación del BOP
Sensor_1	BOOL	Entrada 0.1	Volumen de lodo de perforación que entra al pozo
Sensor_2	BOOL	Entrada 0.2	Volumen de lodo de perforación que sale del pozo
Sensor_3	BOOL	Entrada 0.3	Presión registrada en la barrena
Marca_k1	BOOL	Marca 0.0	Símbolo del relé K1
Marca_k2	BOOL	Marca 0.1	Símbolo del relé K2
Marca_k3	BOOL	Marca 0.2	Símbolo del relé K3
MarcaACT_HMI	BOOL	Marca 0.3	Marca usada para iniciar el proceso a través de la HMI
VisualizarHMI	BOOL	Marca 0.4	Marca usada para visualizar el proceso en la HMI
MarcadorContador	BOOL	Marca 0.5	Inicia el incremento
MarcaTimer	BOOL	Marca 0.6	Ciclo cada 60 segundo
Solenoide 1Y	BOOL	Salida 0.0	Salida para la activación del solenoide 1Y
Solenoide 2Y	BOOL	Salida 0.1	Salida para la activación del solenoide 2Y
Solenoide 3Y	BOOL	Salida 0.3	Salida para la activación del solenoide 3Y

Tabla No 9 Datos de variables de entrada, salida y marcas

3.2.3. Creación del proyecto en TIA

TIA (Totally Integrated Automation) es una estrategia en la automatización tecnológica, la cual fue desarrollada en 1996 por Siemens Automation and Drives. Esta estrategia define la interacción de componentes singulares extensivos, herramientas software (SW) y servicios para alcanzar una solución automatizada.

En la imagen 24 se crea el proyecto con el nombre de Automatización BOP; este proyecto se desarrolló en un lenguaje escalera o LD ya que para su simulación es más fácil seguir las guías de activación y ver el proceso que lleva.

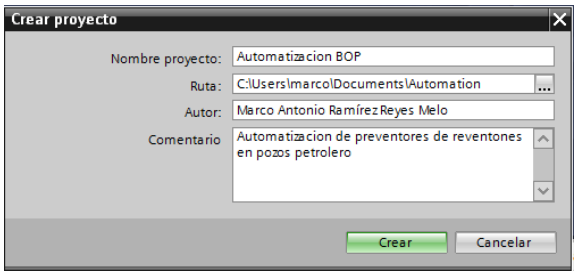


Imagen No 24 Creación de proyecto

Para la simulación en TIA V13 se utilizó el CPU 1516-3 PN/DP como se ve en la imagen 25, ya que la compatibilidad de esta versión solo abarca los CPU 1200 a 1500 y que los dispositivos que se encuentran en el laboratorio ya no son aptos para su simulación de TIA versión 13.

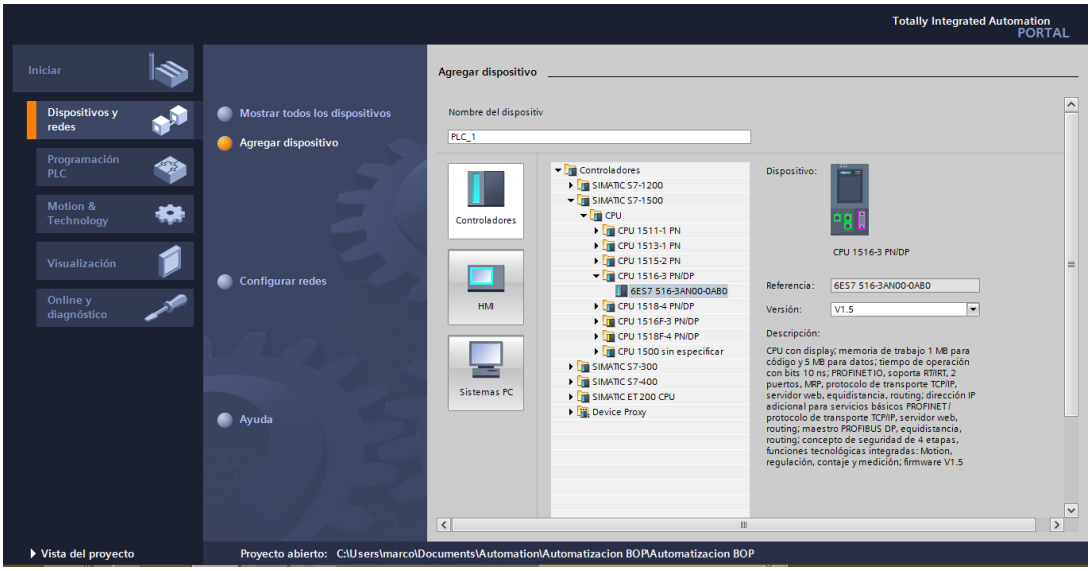


Imagen 25 CPU 1516-3 PN/DP

Descripción técnica del uso de TIA: CPU con display; memoria de trabajo 1 MB para código y 5 MB para datos; tiempo de operación con bits 10 ns; PROFINET IO, soporta RT/IRT, 2 puertos, MRP, protocolo de transporte TCP/IP, servidor web, equidistancia, routing; dirección IP adicional para servicios básicos PROFINET / protocolo de transporte TCP/IP, servidor web, routing; maestro PROFIBUS DP, equidistancia, routing; concepto de seguridad de 4 etapas, funciones tecnológicas integradas: motion, regulación, contaje y medición; firmware V1.5

Nota: Estas CPUs no tienen E/S integradas en la CPU, por lo que se deben agregar módulos con ellas, se eligieron los siguientes modulos por su simplicidad y semejanza a los que contiene CPU 312.

Módulo de entradas: DI 16x24VDC BA

Descripción: Módulo de entradas digitales DI16 x 24V DC; en grupos de 16; retardo a la entrada 3,2ms; tipo de entrada 3 (IEC 61131).

Módulo de salidas: DQ 8x24VDC/2A HF

Descripción de salidas: Módulo de salidas digitales DQ8 x 24V DC / 2A; en grupos de 4; 8A por grupo; diagnóstico parametrizable; valor sustitutivo parametrizable para salida.

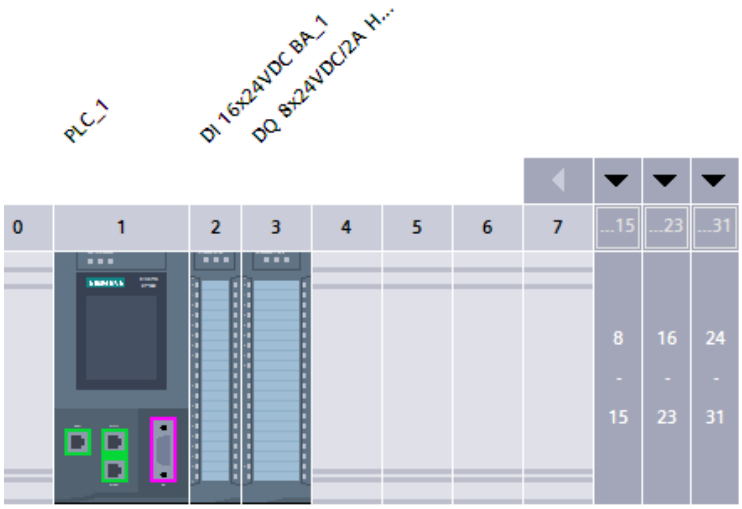


Imagen No 26 Módulos

Una vez descritas las características del CPU, se declararon las variables, como se puede ver en la imagen 27, que se describieron anteriormente en la tabla de datos de variables de entrada, salida y marcas. Todas las variables son booleanas y están definidas en la dirección con una I si son entradas, M si son marcas y Q si son salidas.

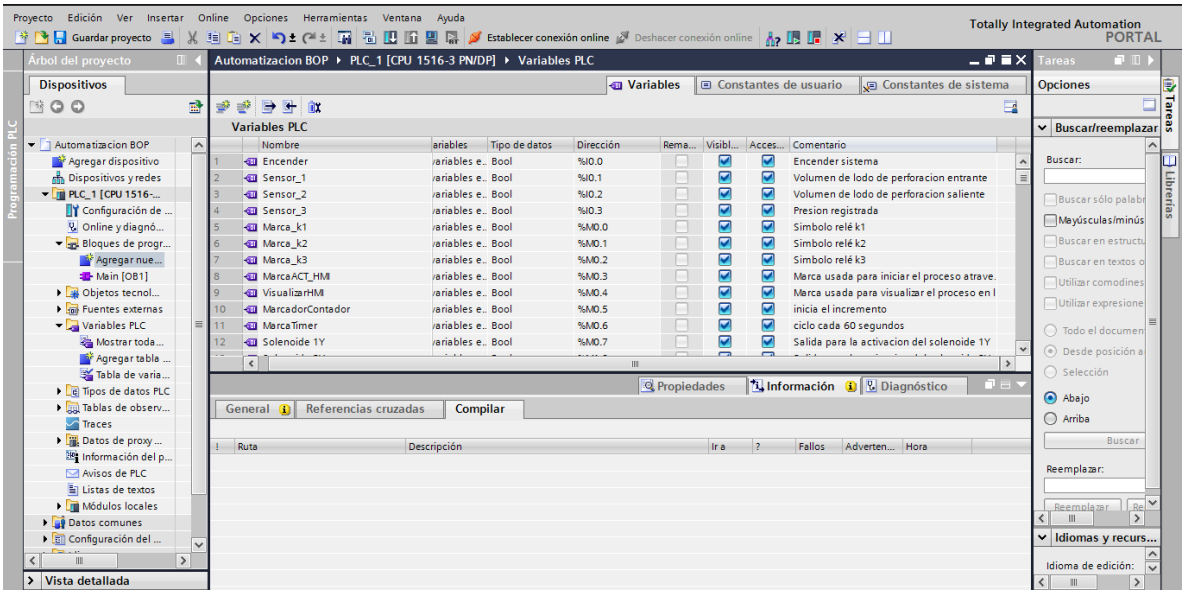


Imagen No 27 Ingreso de las variables

Teniendo las variables, se creó un bloque de funciones con el nombre de Bloque_1(FC1) con el fin de realizar el programa y después mandarlo al bloque organizacional, para que de esta manera se convierta en un programa hereditario y ser mejorado posteriormente.

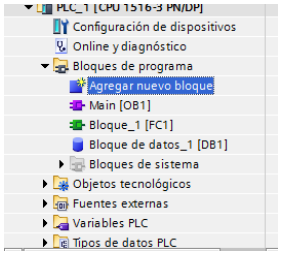


Imagen No 28 Bloques del PLC (CPU1516-3)

Programa Automatización BOP en lenguaje KOP

El programa se divide en 5 segmentos, estos están ordenados con respecto a las entradas de los periféricos y su dirección es asignada en el PLC de igual manera para los actuadores de salida con el fin de controlar el pozo; de esta manera el riesgo que se llegara

a presentar en el pozo se controlaría a la brevedad dando como resultado un porcentaje menor al descontrol y cierre total de éste.

El segmento 1 consiste en la visualización del proceso; la visualización se hará por medio de un dispositivo HMI de igual manera simulado que estará conformado por una marca llamada MarcaACT_HMI, que como su nombre lo dice, será la activación de la visualización. Las demás marcas Encender, Sensor_1, Sensor_2 y Sensor3 son las variables de activación de los periféricos de entrada, excepto Encender, ya que éste será un interruptor físico que activará el proceso del PLC.

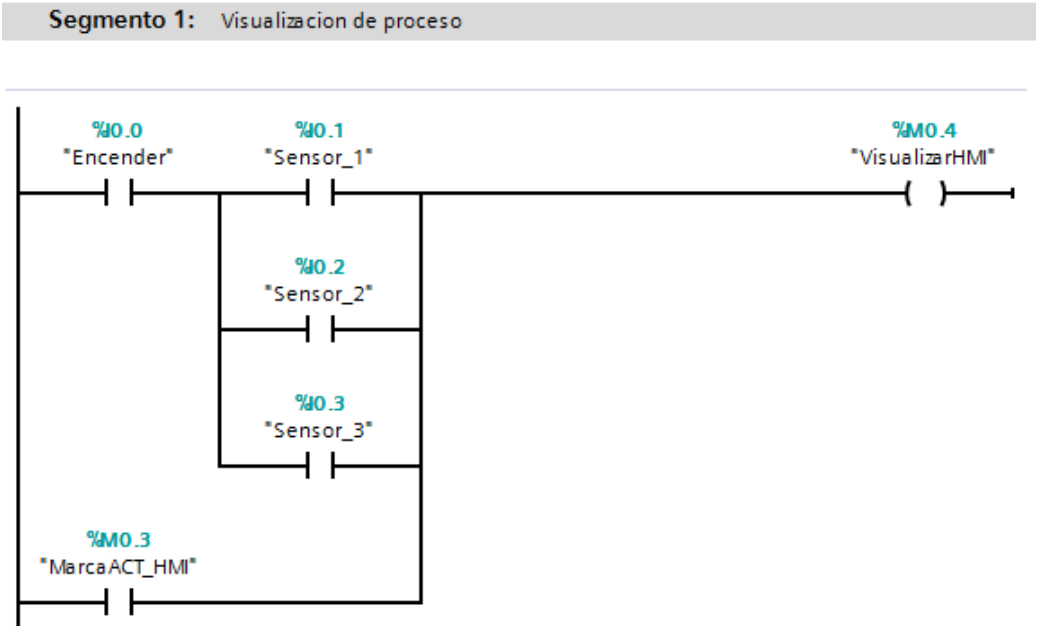


Imagen No 29 Segmento 1

Para los siguientes segmentos se tuvo que realizar la creación de un bloque de datos con el nombre Bloque de datos_1[DB1] que se utilizaron para la simulación. La creación del bloque es para poder generar valores que se pudieran registrar en los periféricos de entrada. De esta manera, se crean los sensores de entrada y no se daña la tabla de valores globales al cambiar valores o borrarlos posteriormente.

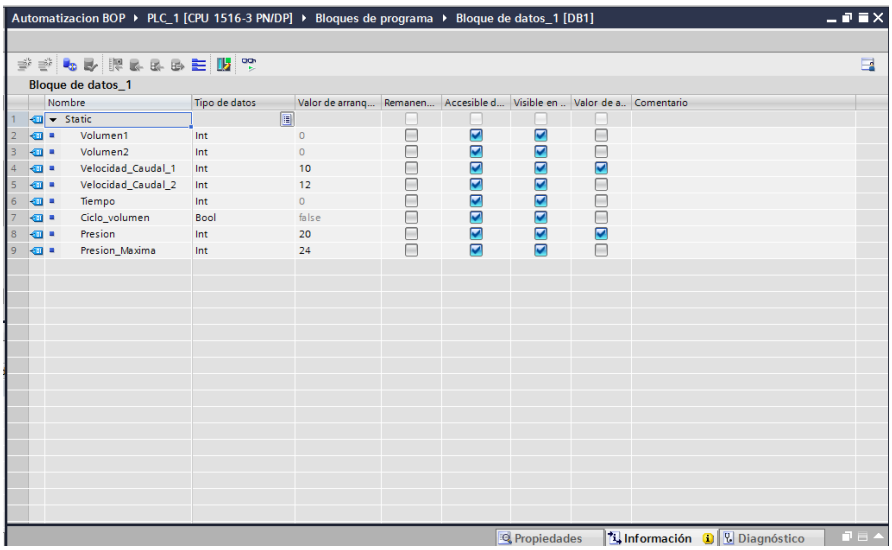


Imagen No 30 Bloque de datos_1[DB1]

Ya declaradas las variables en el bloque, se genera el segmento 2 que consiste en controlar al pozo en caso de un fluido externo al lodo de perforación; este proceso se lleva a cabo por medio de dos caudalímetros que generan los valores de Volumen1 y Volumen2.

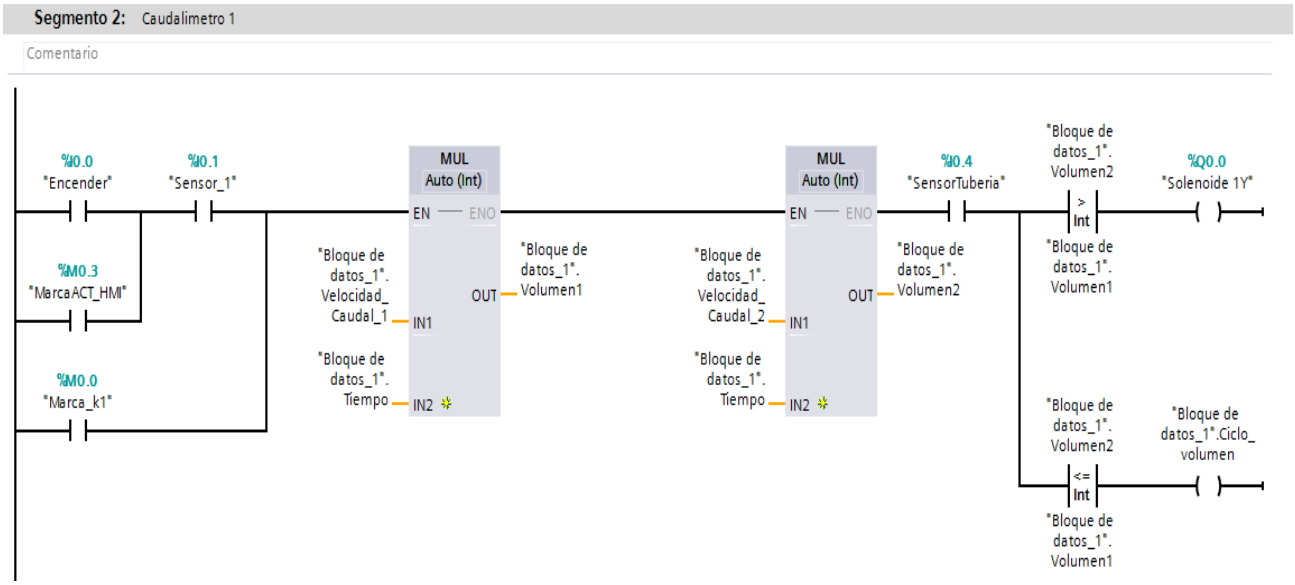


Imagen No 31 Segmento 2

El segmento 2 consiste en dos marcas conectadas en paralelo, las cuales son Encender y MarcaACT_HMI, seguidas por el Sensor_1, el cual indica el paso del fluido por el caudalímetro 1; este sensor está conectado en paralelo con la marca K1 que funciona

como memoria; posteriormente entra a una función matemática que multiplica $Caudal_1 * Tiempo = Volumen1$ seguida por la misma operación, pero con diferentes entradas $Caudal_2 * Tiempo = Volumen2$. Después de las operaciones matemáticas, se encuentra el SensorTuberia que sensor indica si hay tubería de perforación o herramienta dentro del pozo; si el sensor está activado pasa a la comparación Volumen2>Volumen1 donde se activa el Solenoide 1 que da como respuesta la activación del actuador RAM de ariete del sistema BOB. Si no se cumple la primera comparación, se activará la bobina de volumen para su registro.

Las comparaciones del segmento dos son importantes, ya que una vez activándose el actuador, este proceso debe de terminar para poder controlar el pozo por completo, ya que el tiempo de control dependerá del grado de complejidad.

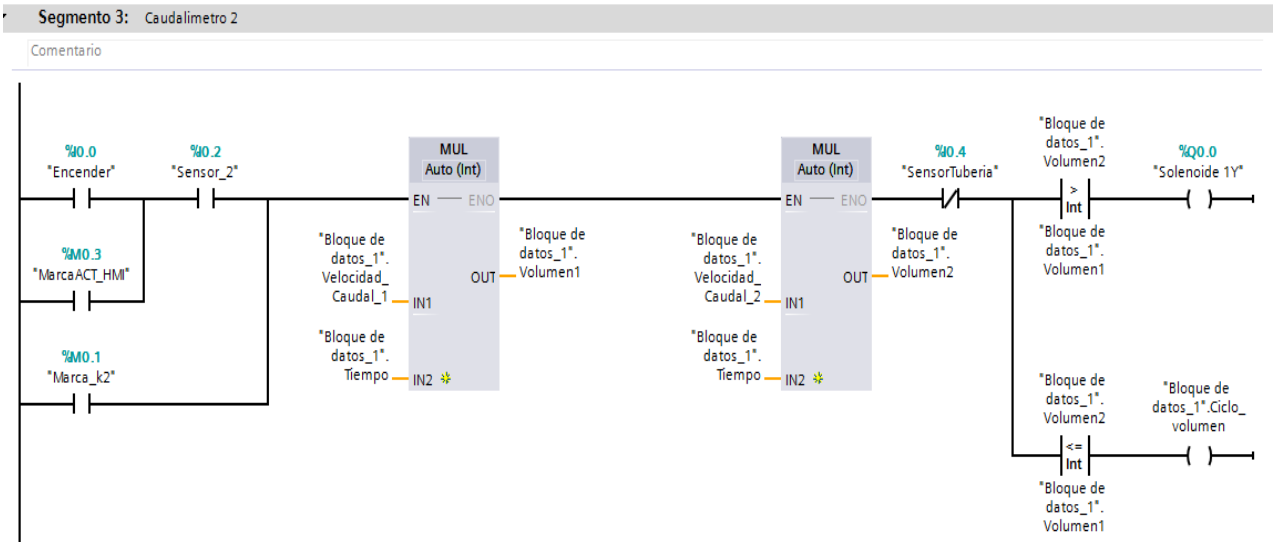


Imagen No 32 Segmento 3

El segmento 3 es parecido ya que solo cambian las variables para estar con respecto al sensor 2 que indica la saida del lodo de perforacion del pozo (la marca de memoria por Marca_k2). Al final de las operaciones matematicas, el SensorTubería debe de estar desactivado para poder pasar a las comparaciones, esto significa que no hay tuberia ni herramientas dentro del pozo y si se cumple la condicion de Volumen2>Volumen1, se acivará el Selenoide 2Y que activa el RAM ciego.

En el segmento 4 se encuentra un temporizador de retardo a conexión para llegar al incremento del tiempo y activar las marcas de los caudalímetros Marca_k1 y Marca_k2.

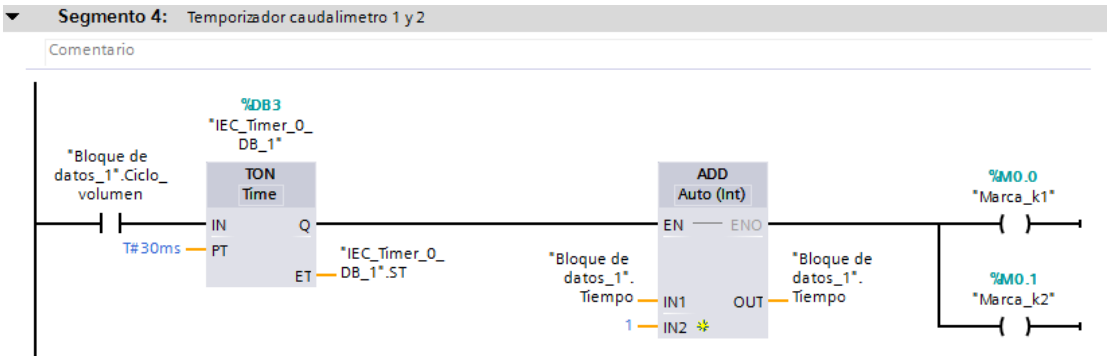


Imagen No 33 Segmento 4

El ultimo segmento controla la variable Sensor_3 que es para comparar la presión registrada en el pozo y compararla con la presión máxima que se puede registrar. Si la presión registrada en el pozo es mayor a la máxima, se activa el Solenoide 3Y el cual acciona el actuador de RAM de corte; cuando se activa esta parte del BOP, quiere decir que el pozo se cerraró por tiempo indefinido ya que el peligro de seguir con la perforacion del pozo es muy grande.

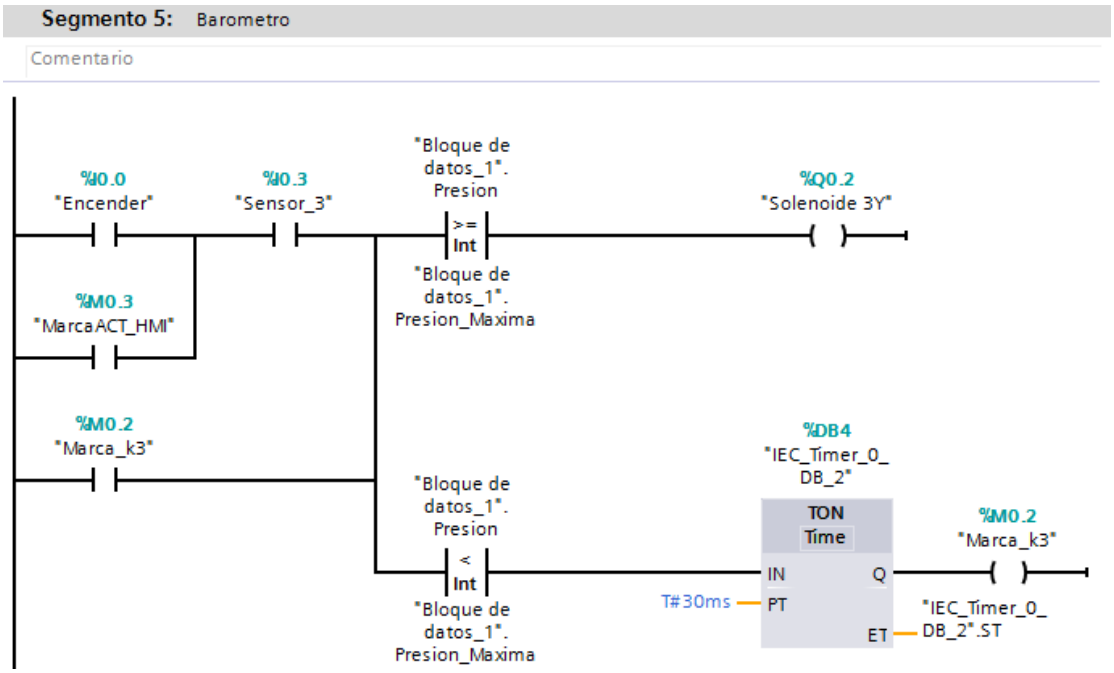


Imagen No 34 Segmento 5

Los 5 segmentos del bloque de funciones se convierten en una línea del bloque de organización para poder compilarlo y simularlo y comprobar el correcto funcionamiento del programa en lenguaje KOP.

Imagen No 35 Bloque de organización

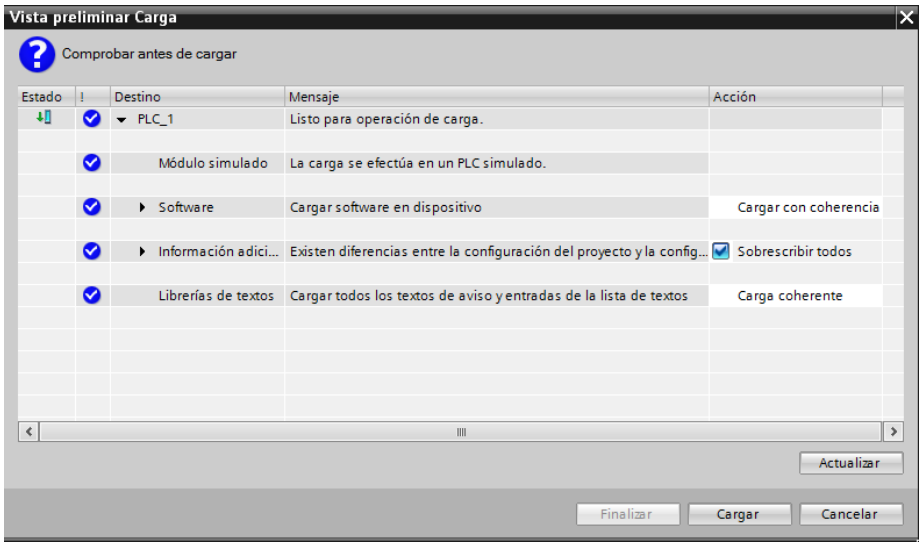


Imagen No 36 Comprobación de código

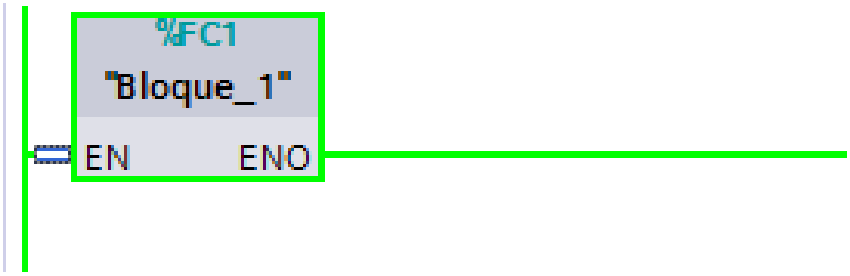


Imagen No 37 Bloque de organización compilado y visualizado

Simulación con S7_PLCSIM

Para generar la simulación se creó un archivo con el nombre Simulación Automatización BOP en otro de los programas que conforman a Siemens; en la imagen se muestra la ventana de simulación en la cual se descarga el programa realizado en TIA.

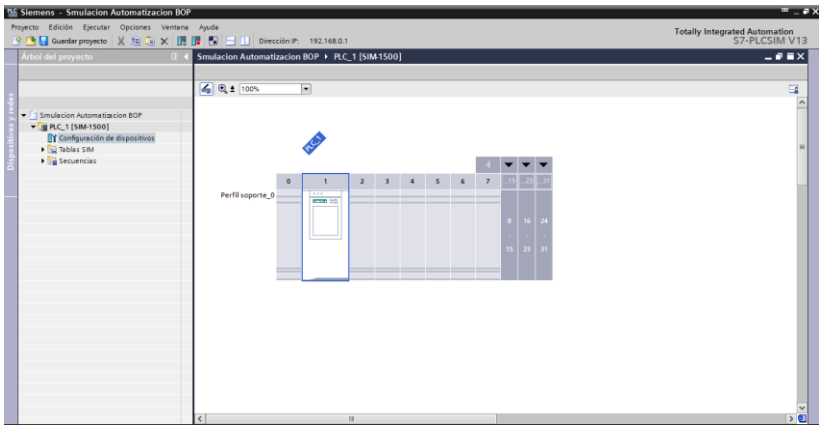


Imagen No 38 Ventana de simulación

El programa se carga con el botón “Carga en dispositivo” y una vez cargado, declaramos las variables que se van a manipular para simular este proceso mediante una lista Sim.

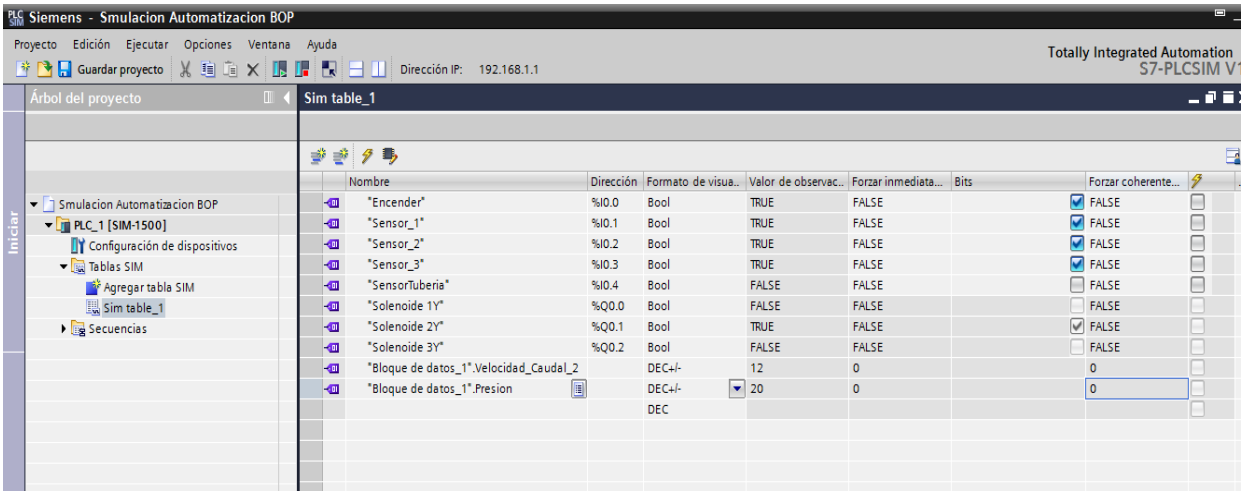


Imagen No 39 Declaración de variables en S7-PLCSIM

3.2.3.1. Simulación en S7-PLCSIM

Para crear la simulación en S7-PLCSIM se consiguió primero la licencia, ya que sin ella no es posible la creación de un nuevo proyecto, una vez que se abrió el programa, se crea una simulación con el nombre Simulación Automatización BOP.

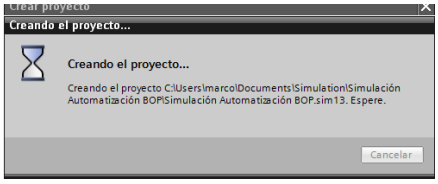
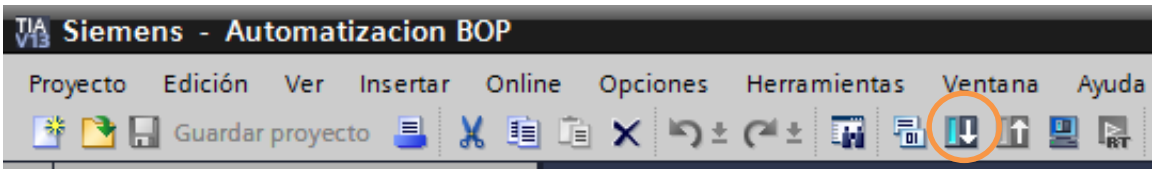


Imagen No 40 Creación de la simulación

Una vez creada la simulación, se importa el proyecto de TIA presionando el botón Cargar en dispositivo, se comprueba el código y nuevamente se compila; este proceso se realiza teniendo en cuenta que los valores del proyecto en TIA están inicializados en sus parámetros de origen.



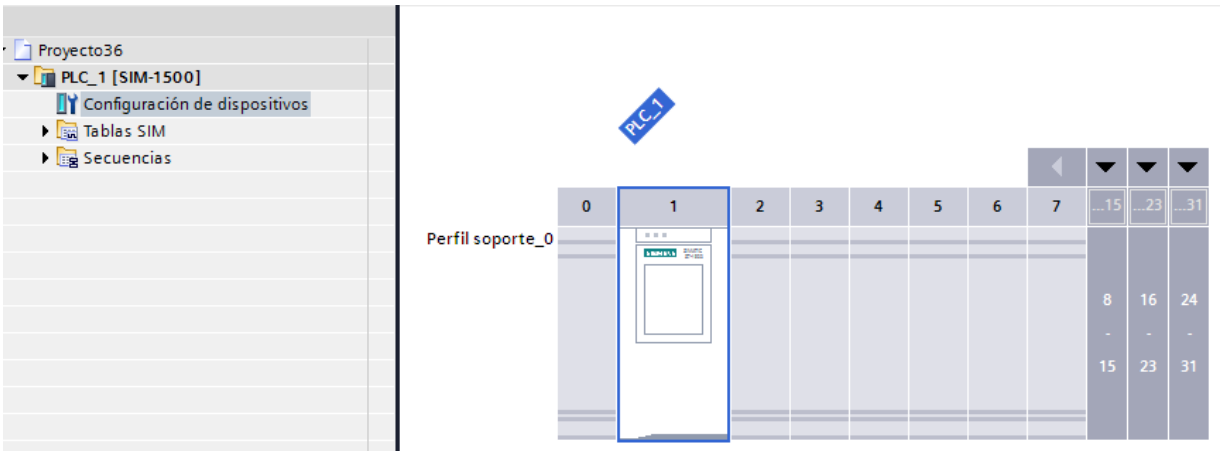


Imagen No 41 Antes de la importación

Una vez importado el proyecto, todos los módulos se reflejan en S7-PLCSIM, de esta manera se pueden activar todas las variables de entrada y simular todos los posibles casos de descontrol del pozo: por lo tanto, se obtuvieron más datos y se justifican los resultados.

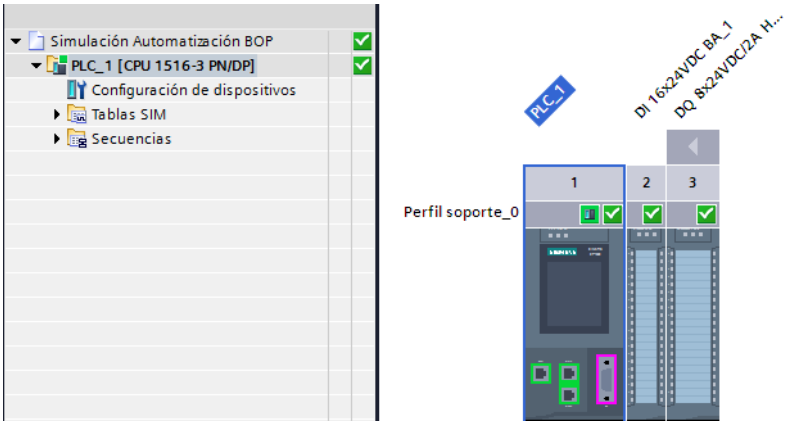


Imagen No 42 Proyecto importado

Para manejar todos los periféricos de entrada se agrega una tabla SIM; esta tabla se puede llenar dando el nombre de la variable o escribiendo la dirección asignada. Como se sabe, el orden de las entradas y salidas es ascendente, por lo tanto es más conveniente ponerlas por dirección.

Sim table_1

	Nombre	Dirección	Formato de visua..	Valor de observac..	Forzar inmediata...	Bits	Forzar coherente...
	"Encender"	%I0.0	Bool	FALSE	FALSE		FALSE
		i	DEC				

Imagen No 43 Tabla SIM

Lista la tabla de variables, se puede manipular su valor activando la casilla Bits, notándose que las salidas no tienen casilla activa ya que están condicionadas por las entradas y no se puede forzar la respuesta directamente.

	Nombre	Dirección	Formato de visua..	Valor de observac..	Forzar inmediata...	Bits	Forzar coherente...
	"Encender"	%I0.0	Bool	FALSE	FALSE		FALSE
	"Sensor_1"	%I0.1	Bool	FALSE	FALSE		FALSE
	"Sensor_2"	%I0.2	Bool	FALSE	FALSE		FALSE
	"Sensor_3"	%I0.3	Bool	FALSE	FALSE		FALSE
	"SensorTuberia"	%I0.4	Bool	FALSE	FALSE		FALSE
	"Solenoide 1Y"	%Q0.0	Bool	FALSE	FALSE		FALSE
	"Solenoide 2Y"	%Q0.1	Bool	FALSE	FALSE		FALSE
	"Solenoide 3Y"	%Q0.2	Bool	FALSE	FALSE		FALSE
	"Paro"	%Q0.3	Bool	TRUE	FALSE		FALSE
	"Bloque de datos_1".Presion		DEC+/-	20	0		0
	"Bloque de datos_1".Presion_Ma.		DEC+/-	24	0		0
	"Bloque de datos_1".Velocidad_..		DEC+/-	10	0		0
	"Bloque de datos_1".Velocidad_..		DEC+/-	12	0		0
			DEC				

Imagen No 44 Tabla completa con las E/S

3.2.3.1.1. Ejemplo de simulación

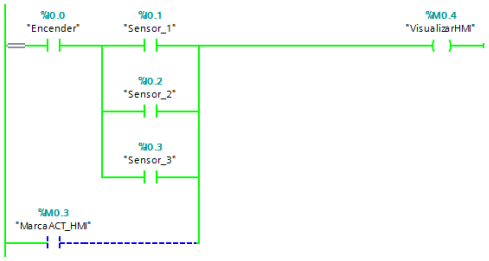
En este ejemplo se muestra la activación del RAM ciego, ya que se cumplen todas las condiciones para su activación.

1- EL sistema esta Encendido
2- El Sensor_1, Sensor_2 y el Sensor_3 están en TRUE
3- El Sensor SensorTuberia está en FALSE
4- Volumen_2>Volumen_1
5- Se activa Solenoide_2

Nombre	Dirección	Formato de visua...	Valor de observac...	Forzar inmediata...	Bits	Forzar coherente...
"Encender"	%I0.0	Bool	TRUE	FALSE	☑	FALSE
"Sensor_1"	%I0.1	Bool	TRUE	FALSE	☑	FALSE
"Sensor_2"	%I0.2	Bool	TRUE	FALSE	☑	FALSE
"Sensor_3"	%I0.3	Bool	TRUE	FALSE	☑	FALSE
"SensorTuberia"	%I0.4	Bool	FALSE	FALSE	☐	FALSE
"Solenoide 1Y"	%Q0.0	Bool	FALSE	FALSE	☐	FALSE
"Solenoide 2Y"	%Q0.1	Bool	TRUE	FALSE	☑	FALSE
"Solenoide 3Y"	%Q0.2	Bool	FALSE	FALSE	☐	FALSE
"Paro"	%Q0.3	Bool	TRUE	FALSE	☑	FALSE
"Bloque de datos_1".Presion		DEC+/-	20	0		0
"Bloque de datos_1".Presion_Ma.		DEC+/-	24	0		0
"Bloque de datos_1".Velocidad_..		DEC+/-	10	0		0
"Bloque de datos_1".Velocid...		DEC+/-	12	0		0

Imagen No 45 Ejemplo de simulación

Se presenta el código dados los valores de los periféricos de entrada, de esta manera se simuló uno de los tres casos posibles en caso de descontrol en pozos petroleros.



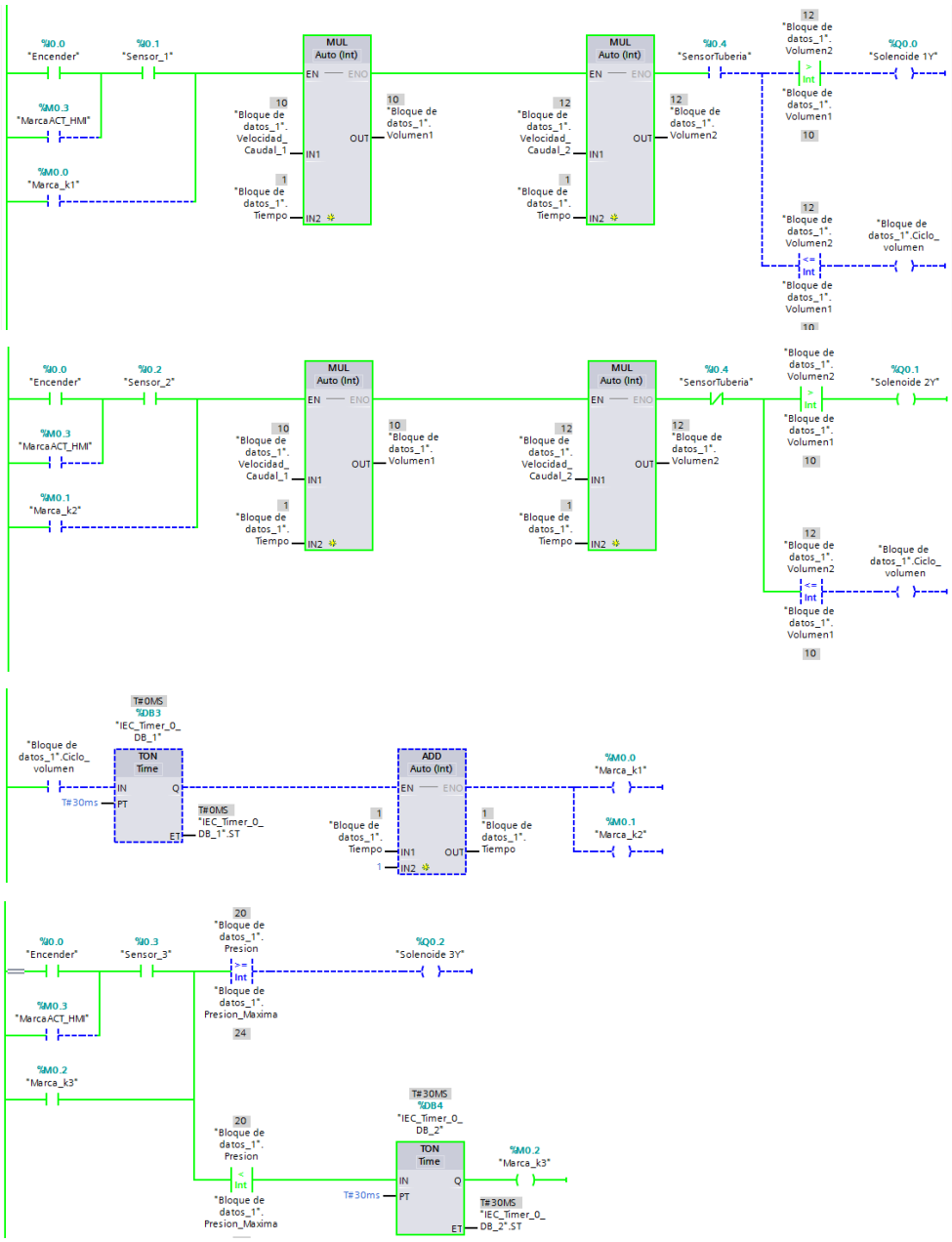


Imagen No 46 Código Ladder

Resumiendo el programa, se muestra el proceso en el Bloque de organización.

Imagen No 47 Bloque de organización simulado S7

3.2.3.2. Visualización HMI

La sigla HMI es la abreviación en ingles de Interfaz Hombre Maquina. Los sistemas HMI se pasan a una ventana de un proceso. Esta ventana es un dispositivo especial, como un panel de operador, pero en esta simulación el panel será la misma computadora. Este software de enlazamiento complementa la automatización; la simulación de este panel se lleva a cabo por medio de WinCC V13.

SIMATIC WinCC es un sistema SCADA que se utiliza para supervisar y controlar los procesos físicos que intervienen en esta industria de perforación y la infraestructura a gran escala y en largas distancias.

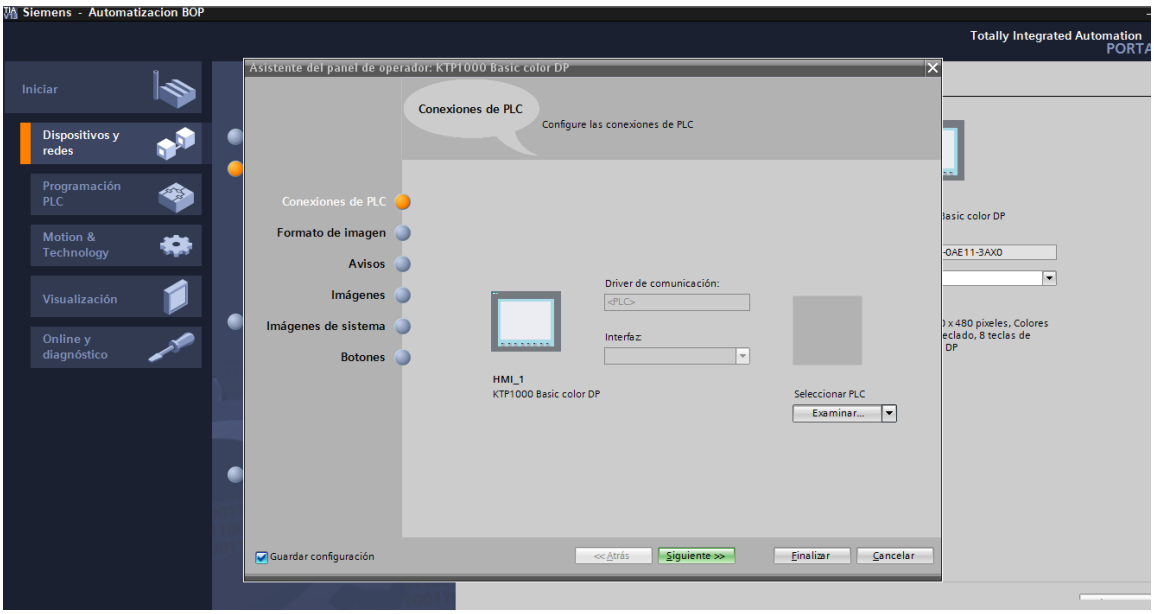


Imagen No 48 Construcción del panel de operador

Las funciones que abarca esta visualización HMI son:

Monitoreo: El operador tendrá la habilidad de obtener y mostrar datos de la planta en tiempo real. Los datos de volumen 1, volumen 2 y presión se muestran en forma numérica; los datos de sensor de tubería se expresan con un texto Sensor de tubería ON.

Supervisión: Esta función permite junto con el monitoreo la posibilidad de ajustar las condiciones oprimiendo un botón de paro de perforación.

Alarma: Esta capacidad dará aviso en caso de un descontrol de pozo, de esta manera los obreros tomaran las medidas pertinentes usando los conocimientos brindados en cursos como RIG PASS.

Control: Esta capacidad es la de cerrar los preventores dependiendo de las variables de entrada.

Para la simulación se consideraron otros botones en la interfaz HMI para el incremento de las variables de comparación y de esta forma observar la activación de los preventores.

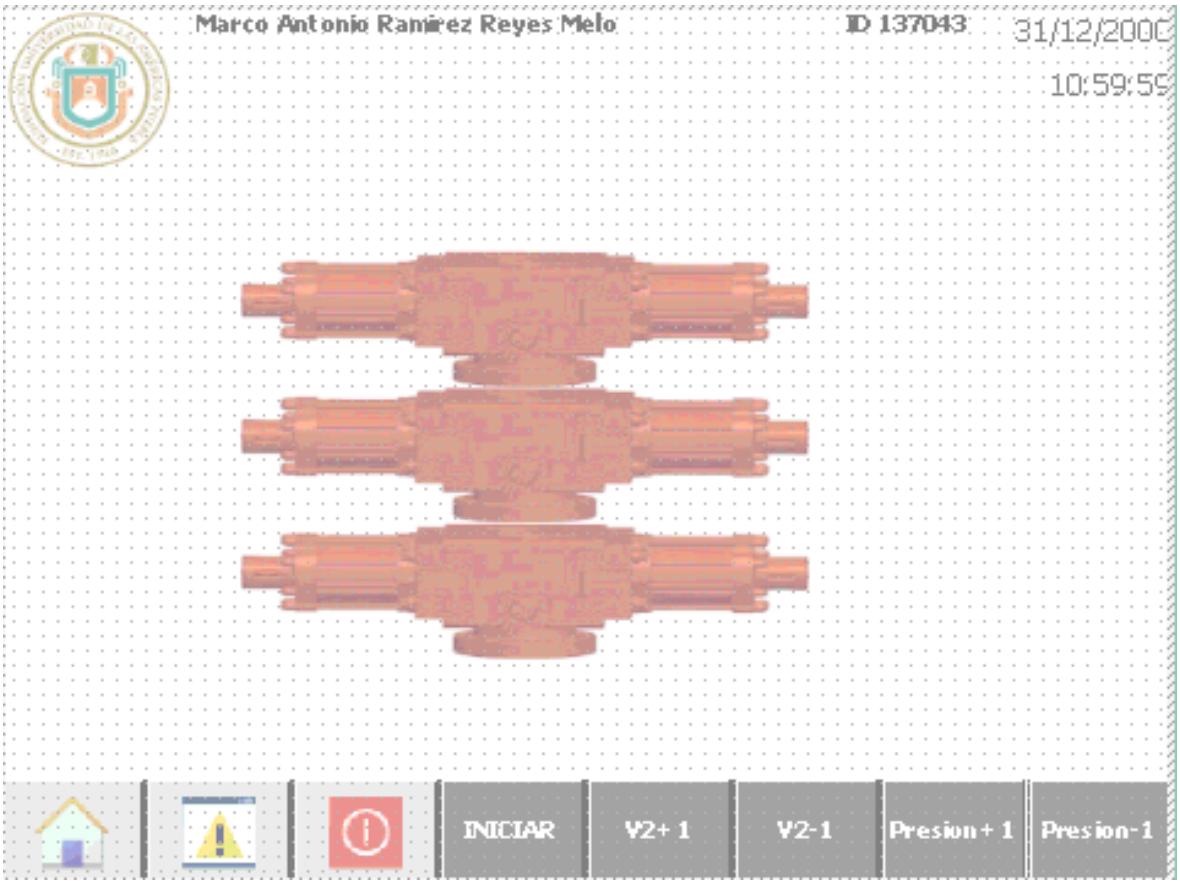


Imagen No 49 Raíz HMI

Se crean 8 botones para poder generar valores para las variables Volumen2 y Presión que se van a controlar y de esta manera poder observar el funcionamiento del código KOP y poder observar la eficiencia de este proceso.

Botón	Descripción
	Restaurar los valores de inicio
	Ver mensaje de proceso
	Apagar proceso
INICIAR	Iniciar proceso
V2+ 1	Incremento en la variable volumen 2 una unidad
V2-1	Decremento en la variable Volumen 2 una unidad
Presion+ 1	Incremento en la variable Presión una unidad
Presion-1	Decremento en la variable Presión una unidad

Tabla No 10 Botones HMI

Teniendo la interface HMI y el PLC, se realiza el enlazamiento de los dos para crear una red de datos.

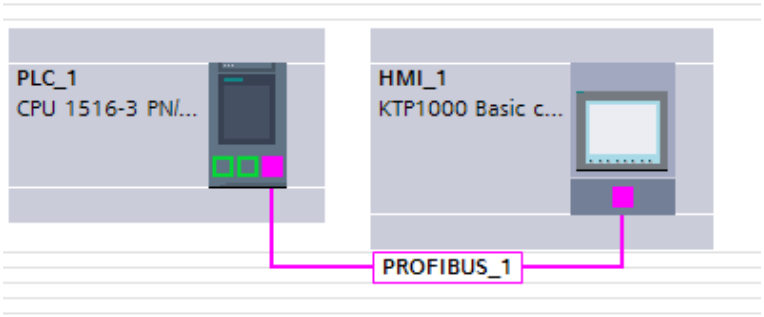


Imagen No 49 Enlazamiento interfaz HMI y PLC

3.2.4. Vinculación con FluidSIM

Se simula una estancia neumática para el programa realizado en TIA autómata S7-1500 que se migra después a un S7-300 con PLCSIM. Para la realización de ello se llevaron a cabo los siguientes requisitos: se necesitó el FluidSIM, simulador de FESTO y el Servidor EzOPC de la misma marca, ya que éste enlaza con el PLSIM, el simulador de Siemens, para los autómatas. Para la programación del autómata se necesitó el Administrador Simatic Step7 o el TIA Portal. En la imagen se muestran las conexiones de activación de los solenoides y su nombre de guía.

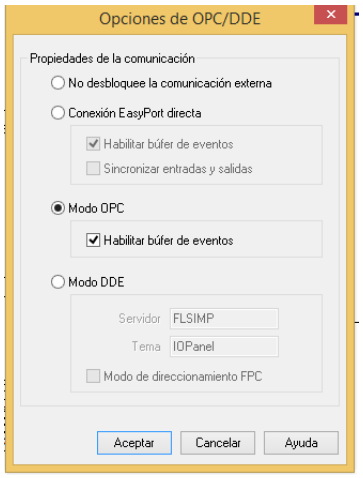


Imagen No 50 Servidor EzOPC

El programa se encuentra en siguiente link:

http://www.mediafire.com/download/17urm2zffk522ep/setup_ezopc_55a.zip.

http://www.mediafire.com/download/8ei6lefjc7lxvw7/setup_easyport_2.1.1.2.zip

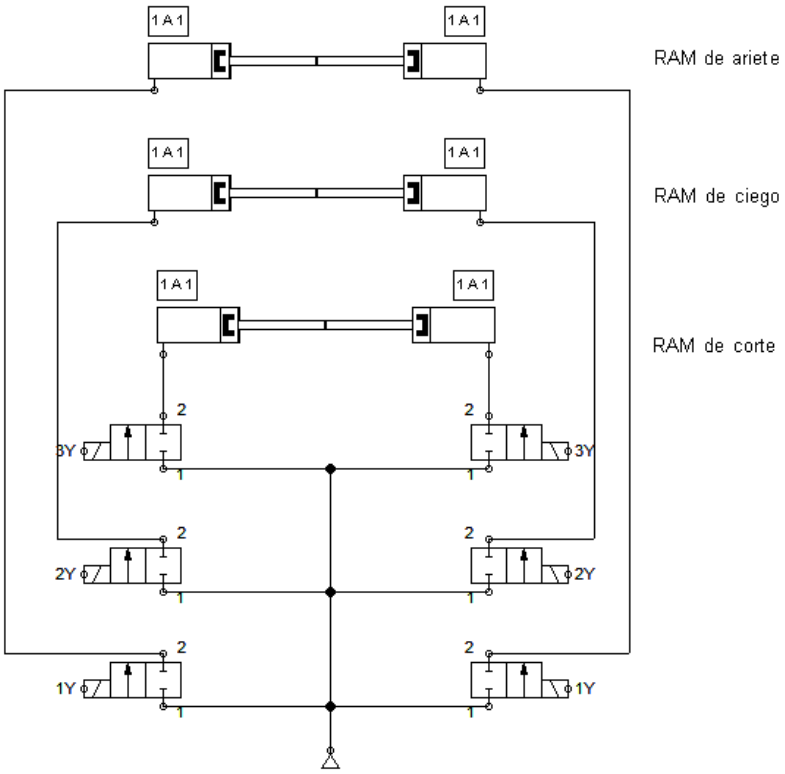


Imagen No 51 Sistema preventor conectado

Solenoide 1Y: RAM de Ariete

Solenoide 2Y: RAM Ciego

Solenoide 3Y: RAM de Corte

Para accionar los periféricos de entrada, se activan desde la plataforma de FluidSIM ya que esta es una simulación y no tenemos valores reales; por ello se agregaron unas condiciones adicionales en el programa Automatización BOP en lenguaje KOP, esto con el fin de poder incrementarlos desde la visualización HMI.

Los accionamientos de las variables de entrada se hacen por medio de interruptores (obturadores) simulados en FluidSIM y se muestra en la siguiente Tabla. Esto se hace con el fin de poder tener más control en las en las direcciones de entrada del PLC y las salidas del mismo.

Interruptor	Marca
Interruptor 1	0
Interruptor 2	1
Interruptor 3	2
Interruptor 4	3

Tabla No 11 Interruptores

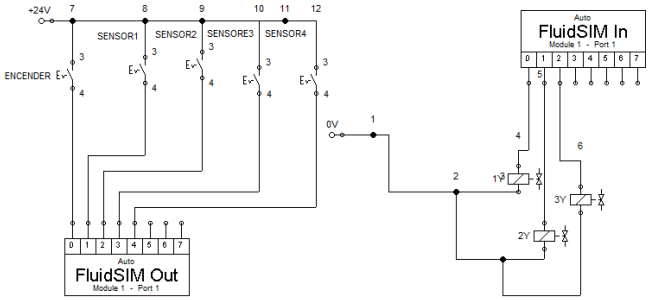


Imagen No 52 Circuito controlado por el PLC Auto

Para referenciar los programas con las direcciones virtuales del PLC, se abren los módulos de entradas y salidas para ser referenciados del servidor OPC al puerto de salida del elemento PLCSIM:EBO, seleccionando los puertos primarios 0.0; de la misma forma se referencian los puertos de entrada y de esta manera enlazamos las diferentes simulaciones creadas y el panel de control HMI con WinCC.

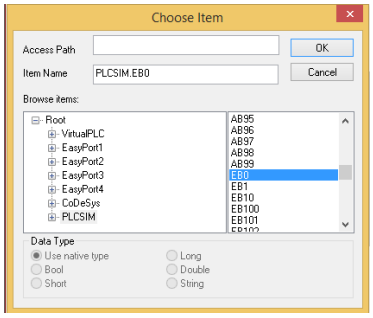


Imagen No 53 Conexión de entradas al PLCSIM virtual

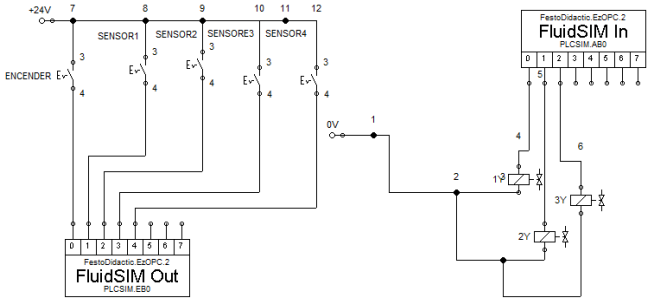


Imagen No 54 Circuito controlado por el PLCSIM

3.3. Acoplamiento del proyecto con los programas de simulación

Para en lazar todos los componentes de este proyecto primero se abre TIA; una vez abierto se compila y se abre la simulación creada en S7-PLCSIM para descargar el proyecto. Teniendo estos dos programas abiertos, abrimos WinCC que nos permitirá simular la interfaz HMI tomando las direcciones IP virtuales de los elementos del proyecto, terminada esta parte se abre FluidSIM y la estructura electroneumática con los módulos de entras y salidas.

Para el direccionamiento ya teniendo todos los programas abiertos y configurado los módulos en FluidSIM con OPC, se maneja una ventana por default llamada Festo Didactic EzOPC – V 5.5, en la cual enlazaremos los programas FluidSIM y S7-PLCSIM

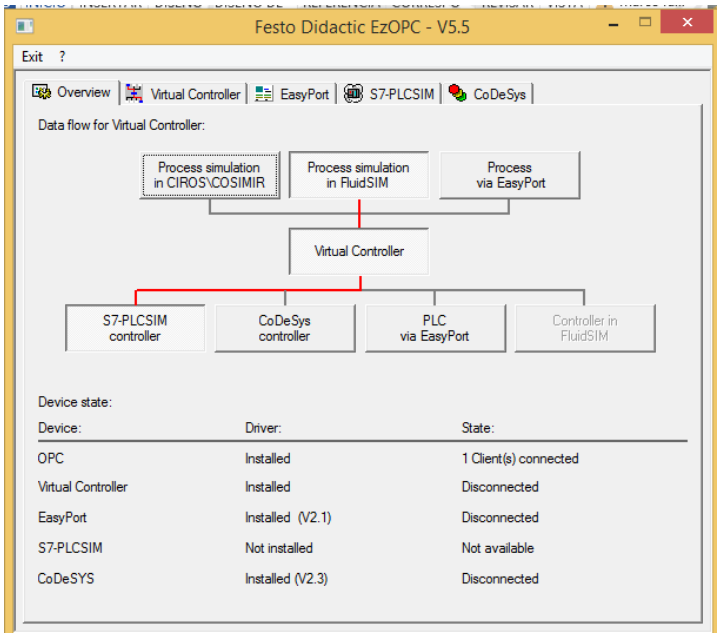


Imagen No 55 Enlazamiento