



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

DESARROLLO DE UN SISTEMA SCADA PARA
CONTROL DE PROCESOS.

TESIS

PARA OBTENER EL GRADO DE
INGENIERO EN SISTEMAS DE ENERGÍA

PRESENTA

LUIS FELIPE TAH ROMERO

DIRECTOR

DR. FREDDY IGNACIO CHAN PUC

ASESORES

MM. JESUS ORIFIEL ALVAREZ RUIZ

DR. HOMERO TORAL CRUZ

DR. EMMANUEL TORRES MONTALVO

DR. JAVIER VÁZQUEZ CASTILLO



CHETUMAL QUINTANA ROO, MÉXICO, OCTUBRE DE 2023



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

TRABAJO DE TESIS TITULADO

“DESARROLLO DE UN SISTEMA SCADA PARA CONTROL DE PROCESOS”

ELABORADO POR

LUIS FELIPE TAH ROMERO

BAJO SUPERVISIÓN DEL COMITÉ DEL PROGRAMA DE LICENCIATURA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

INGENIERO EN SISTEMAS DE ENERGÍA

COMITÉ SUPERVISOR

DIRECTOR:

DR. FREDDY IGNACIO CHAN PUC

SUPERVISOR:

MM. JESUS ORIFIEL ALVAREZ RUIZ

SUPERVISOR:

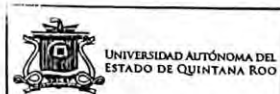
DR. HOMERO TORAL CRUZ

SUPERVISOR SUPLENTE:

DR. EMMANUEL TORRES MONTALVO

SUPERVISORA SUPLENTE:

DR. JAVIER VAZQUEZ CASTILLO



ÁREA DE TITULACIÓN

QUINTANA ROO, MÉXICO, OCTUBRE DE 2023



AGRADECIMIENTOS

Al Dr. Freddy Ignacio y MM. Jesús Álvarez y por ser buenos mentores en el área de electrónica y automatización. Agradezco por darme este tema de tesis y siempre estar pendiente de este proyecto.

DEDICATORIA

A mis padres por siempre apoyarme durante todo este proceso y apoyarme sin importa la distancia los momentos difíciles, las enfermedades, ustedes nunca se rindieron por darme lo mejor y por darme una carrera profesional, gracias a ustedes y a su esfuerzo logre llegar hasta aquí.

Agradecer a mi pareja por sus buenos consejos, por motivarme a ser una mejor persona y siempre dar lo mejor.

A mis amigos y compañeros que estuvieron durante todo este proceso.

Gracias al profe Jesús Álvarez por brindarme su apoyo, amistad y sus buenos consejos.

RESUMEN

Se presenta el desarrollo e instalación de un sistema SCADA (Supervisión, Control y Adquisición de Datos) utilizando un PLC de la marca DELTA modelo DVP20SX2. El sistema consiste en el monitoreo de variables eléctricas básicas de un motor monofásico. Las variables a medir son tensión eléctrica, corriente y potencia, además se agregó la temperatura para monitorear posibles sobrecalentamientos. La programación de las instrucciones para el PLC se hizo en lenguaje escalera usando el software ISPSOft. Se agrega una interfaz HMI (Interfaz Humano-Maquina) para la visualización de las variables eléctricas y de temperatura del motor usando el software DOPSOft.

Contenido

| | |
|---|----|
| Capítulo 1 Introducción | 1 |
| 1.1 Justificación..... | 3 |
| 1.2 Objetivo general | 4 |
| 1.3 Objetivos específicos | 4 |
| 1.4 Alcances y limitaciones | 4 |
| Capítulo 2 Marco Teórico | 5 |
| 2.1 Descripción de los niveles de automatización | 6 |
| 2.1.1 Nivel de campo, producción o procesos: | 7 |
| 2.1.2 Nivel de control | 13 |
| 2.1.3 Nivel de supervisión | 19 |
| 2.1.4 Nivel de planeación | 22 |
| 2.1.5 Nivel de gestión | 23 |
| Capítulo 3. Programación del PLC | 24 |
| 3.1 Instalación del Software ISPSOft..... | 24 |
| 3.2 Iniciar el Software ISPSOft | 27 |
| 3.3 Programación del PLC..... | 30 |
| 3.3.1 Estructura del diagrama escalera. | 30 |
| 3.3.2 Tipos de bloques de control | 32 |
| 3.3.3 registros y Relés de PLC Delta | 34 |
| 3.3.4 Registros de entradas y salidas analógicas | 35 |
| 3.3.5 Registros de condiciones y parámetros de entrada y salida analógica | 36 |
| 3.3.6 Tipos de datos | 39 |

| | |
|---|----|
| 3.3.7 Declaración de variables | 40 |
| 3.4 Diagrama de flujo del algoritmo | 42 |
| 3.5 Programa realizado | 43 |
| 3.5.1 Programación del PLC | 44 |
| 3.5.2 Programación de la pantalla HMI | 45 |
| 3.5.3 Programación LD del sensor de temperatura LM-35 | 46 |
| 3.5.4 Programación LD del sensor de corriente LA 25-NP | 50 |
| Capítulo 4 Programación de la pantalla | 53 |
| 4.1 Instalación del software DOPSoft 2.0 | 53 |
| 4.2 Iniciar el software DOPSoft..... | 57 |
| 4.3 Programación de la pantalla | 59 |
| 4.3.1 Explicación de los bloques | 59 |
| 4.3.2 Diseño de la pantalla | 61 |
| Capítulo 5 Sistema SCADA..... | 71 |
| 5.1 Conexión entre los dispositivos | 71 |
| 5.2 Resultados..... | 76 |
| Conclusión..... | 80 |
| Bibliografía | 81 |
| Anexos 1 Mapa de memoria para los SX2 | 84 |
| Anexos 2 Lista numérica de instrucciones | 85 |

CAPÍTULO 1 INTRODUCCIÓN

“La automatización industrial consiste en gobernar la actividad y la evolución de los procesos sin la intervención continua de un operador humano” (Pérez López, 2015). En los últimos años se ha estado desarrollando un sistema nombrado SCADA que en sus siglas en ingles significa (Supervisory Control And Data Acquisition o Control con Supervisión y Adquisición de Datos) por lo tanto, se puede automatizar con control del sistema en tiempo real, recopilando información de las variables realizando análisis de monitoreo y control de un proceso. Para ello se debe utilizar diversos softwares de aplicación, sistemas de comunicación, etc. Que le permite al operador tener acceso completo al proceso mediante su visualización en una pantalla HMI.

Hoy en día la industria 4.0 buscan estrategias para agilizar el proceso de producción utilizando redes de comunicaciones en los diferentes niveles de la planta, con la finalidad de optimizar y de lograr hacer eficiente el proceso de manufactura. En ese sentido la pirámide de la automatización se ocupa para una comunicación entre humano, maquina y gestión de producción/venta que ocurre dentro de la empresa.

La base de la automatización comienza con los actuadores y sensores estos elementos son los que hacen el trabajo, mientras el nivel dos se encuentra los dispositivos de control como los Lógicos Programables (PLC'S) y las Interfases Hombre Maquina (HMIs). El nivel tres se trata de los sistemas SCADA donde hay comunicación remotamente en cualquier parte de planta dando gestión, información y control de los actuadores y sensores a través de una red de supervisión (HMIS, celulares u ordenadores) este proyecto de tesis abarca estos puntos. Mientras el nivel cuatro es la parte de la red de operación conocida como Sistema de Ejecución de Manufactura (MES) es una herramienta tecnológica que se utiliza para crear soluciones ligadas al proceso de producción de la planta. Reúne información de los procesos de ventas/pedidos esto se enlaza mas que nada al nivel dos y tres y por

ultimo Los Sistemas de Planificación de Recursos de la Empresa (ERP) son sistemas de gestión que da información sobre clientes, proveedores, contabilidad, costes, etc.

La propuesta de este trabajo pretende desarrollar e instalar un sistema de control, supervisión y monitoreo de Sistema y Adquisición de Datos (SCADA) basado en un controlador de la serie Delta del modelo DVP20SX2 (PLC), con entradas y salidas analógicas entradas digitales y salidas a relevador con comunicación de interfaz RS-485 compatible con una pantalla táctil e igualmente de la marca Delta modelo DOP-B03S211 Interfaz Hombre Maquina (HMI) en sus siglas en ingles.

Con la necesidad de ampliar la infraestructura del laboratorio de electrónica de la Universidad Autónoma del Estado de Quintana Roo; la cual se pretende incentivar el interés de los nuevos aspirantes y futuros a egresar la carrera en Ingeniería en Sistemas de Energía.

Uno de los problemas más desafiantes en la enseñanza de la teoría de la automatización, es la falta de sensores y equipos industriales, para aplicaciones realista. Lo cual la teoría con la práctica debe ser perfectamente empalmada para una buen aprovechamiento.

El diseño de este banco experimental se realiza con la finalidad de ser empleado como equipo de prueba para diferentes aplicaciones en el área de control, automatización y maquinas eléctricas. El contenido de esta tesis es principio para crear automatización y control en las distintas áreas de interés.

1.1 JUSTIFICACIÓN

El proyecto consiste en un sistema SCADA basado en un PLC de la marca DELTA para el monitoreo de las variables eléctricas de un motor monofásico. Este tipo de sistemas ayudan a la detección de fallas enviando alertas al personal de mantenimiento en caso de una sobrecarga de corriente o por algún sobrecalentamiento en el embobinado del estator, incluso detectar si el motor no está operando por una posible avería. Además, facilita al operador visualizar y manipular ciertas variables, de igual manera ayuda a tener un mejor control y un monitoreo constante, lo que se conlleva al ahorro de tiempo, personal y presupuesto.

Debido a que estos sistemas son sumamente populares por la introducción de la industria 4.0, este proyecto puede servir como guía a trabajadores que se encuentren en el campo laboral para la configuración de los procesos automatizados, empleando una lógica de escalera lo cual es lenguaje de programación aplicado para PLC'S. Por otro lado, beneficia a los alumnos que están en la formación profesional y están interesados en este tipo de proyectos, puesto que en la actualidad este tipo de sistemas son muy aplicados en los procesos industriales y el campo laboral es muy amplio. En ese sentido se desea exponer el principio de aplicación de estos equipos de tal forma que el alumno conozca la sistematización, configuración y la aplicación que se puede hacer con un PLC y una pantalla HMI, lo anterior sienta las bases a futuros alumnos que cursan ingeniería.

1.2 OBJETIVO GENERAL

Implementar un sistema SCADA para el control y monitoreo de las variables eléctricas y de temperatura de un motor monofásico usando PLC y pantallas HMI de la marca DELTA.

1.3 OBJETIVOS ESPECÍFICOS

Seleccionar los dispositivos y sensores para implementar el sistema SCADA.

- Instalar y configurar la pantalla Delta (HMI) modelo: DOP-B03S211, para la visualización y monitoreo de corriente y temperatura de un motor monofásico.
- Programar y configurar el PLC modelo: DVP20SX211T con entradas analógicas y digitales.
- Desarrollar un banco experimental para el laboratorio de PLC'S.
- Realizar pruebas del sistema para el análisis del proyecto.

1.4 ALCANCES Y LIMITACIONES

Los alcances contemplados en esta propuesta son los siguientes:

- Se delimitará en conocer el software ISPSOFT orientando a utilizar una lógica de escalera para sus diferentes bloques de operación que permita el monitoreo constante de un motor monofásico.
- Diseño de una interfaz con el software DOPSOFT que permita el monitoreo constante de las variables analógicas de corriente y temperatura en tiempo real para el equipo de potencia. En ese sentido para que funcione y no sobrepase los límites dañando el motor.

Las limitaciones contempladas en esta propuesta son las siguientes:

- No se cuenta con sensores industriales para una aplicación más realista.

CAPÍTULO 2 MARCO TEÓRICO

El desarrollo de la electricidad y de la electrónica permitió la aparición de una nueva generación de autómatas, capaces de imitar realmente algunas funciones intelectuales y no solo de producir determinados comportamientos. La era moderna de la automatización comienza con la aparición, en 1775 de la máquina de vapor de simple efecto inventada por James Watt. La máquina de doble efecto de 1784 estaba provista de dos automatismos: el distribuidor de vapor y el regulador de bolas, que mantenía constante la velocidad del árbol de salida a pesar de las fluctuaciones de carga. En el contexto actual, la AUTOMÁTICA se define como la Ciencia y la Técnica de la automatización, que agrupa el conjunto de las disciplinas teóricas y tecnológicas que intervienen en la concepción, la construcción y el empleo de los sistemas automáticos (Emilio, 2020, págs. 28-29).

Si realizamos una breve reseña histórica acerca de cómo se ha desarrollado los sistemas de automatización hasta nuestros días, se puede resumir de la siguiente:

1950 – Concepción de un sistema de control por computadora. Brow and Campbell.

1954 – Primera computadora digital para control. Aplicada en control de vuelos.

1958 – Primer sistema para monitoreo en una planta de potencia.

1959 – Primer sistema en lazo cerrado en una refinería en Texas, 40 lazos supervisorios.

1962 – Primer Control Digital Directo (DDC) en una planta química en el Reino Unido.

1963 – Aparecen los sistemas operativos para tiempo real con superlenguajes.

1974 – Con el desarrollo de los microprocesadores y microcomputadoras, aparecen los sistemas de control distribuido con sus diferentes variantes y estructuras.

(Eduardo, 2012, pág. 8)

2.1 DESCRIPCIÓN DE LOS NIVELES DE AUTOMATIZACIÓN

Para un mejor estudio de los sistemas de automatización, vamos a referirnos a la forma en la que se organiza un sistema automático, para ello la pirámide de la automatización es una estrategia orientada a la intercomunicación de cada uno de los niveles en el proceso de producción en una empresa. Esta estructura que integra personas, procesos, información y estructuras tecnológicas, proporciona un método más eficaz de gestión que ofrece muchas ventajas competitivas para la empresa (L. Blevins, K. McMillan, K. Wojsznis, & W. Brown, 2008, pág. 22).

En relación con eso la automatización puede ser realizada en diferentes niveles o bloques según el tipo de empresa o industria, y allí es donde entra en juego la pirámide de la automatización (SDI, 2022), lo que permite visualizar cada nivel de proceso de producción. La toma de datos y medición de variables se lleva a cabo en el nivel más bajo de la pirámide que finalmente asciende hasta niveles de control, supervisión y gestión que generan la toma de decisiones para la planificación del mejoramiento continuo de la producción (L. Blevins, K. McMillan, K. Wojsznis, & W. Brown, 2008), por lo tanto, los niveles que conforman la pirámide son básicamente cinco ordenados de forma ascendentes (ver la figura 1).

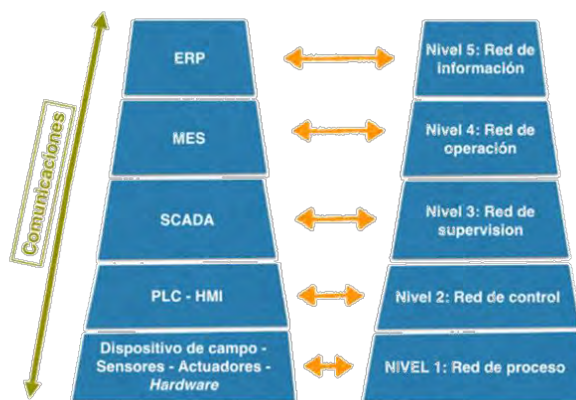


Figura 1 Pirámide de la automatización Industria 4.0
Gestión por procesos.

Fuente: (WITORG, 2023)

2.1.1 Nivel de campo, producción o procesos:

Estos equipos informan al control PLC del nivel 2 al igual pueden actuar al recibir una orden del PLC en ese sentido son actuadores y sensores.

¿Qué es un sensor?

Un sensor se define como un dispositivo de entrada que provee una salida manipulable de la variable física medida (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, pág. 17). El sensor solo puede ser un dispositivo de entrada, ya que este último siempre será un intermediario entre la variable física y el sistema de medida.

Hoy en día, los sensores entregan señales eléctricas a la salida, ya sean analógicas o digitales, debido a que este tipo de dominio físico es el más utilizado en los sistemas de medidas actuales.

En el campo industrial se puede ver algunos de estos equipos que tienen la finalidad de realizar y controlar un proceso. A continuación, se presenta las clasificaciones de los sensores por el principio de transducción (ver la figura 2).

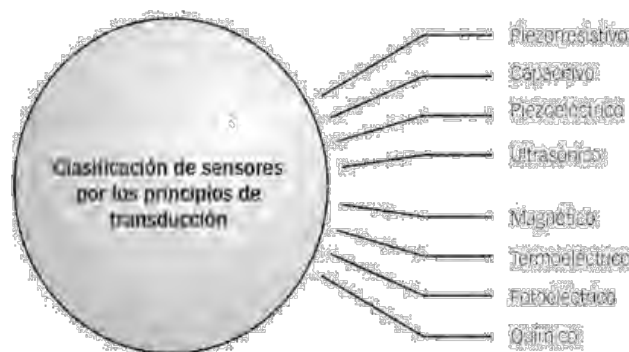


Figura 2 Clasificación de sensores por los principios de transducción

Fuente: (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, pág. 17)

Esta clasificación suele ser la más común ya que tu principio de funcionamiento siempre será el mismo, lo cual únicamente depende del tipo de configuración en que se coloque y cómo se interprete la señal de salida de este. En la tabla 1 se puede ver la clasificación de sensores por el tipo de variable de medida, sin importar cuál sea su tipo de principio de transducción o qué tipo de variable física sea la que midan, siempre tienen características particulares que los distinguen entre sí.

Tabla 1 Clasificación de sensores por el tipo de variable medida

| | |
|--|--------------------------------------|
| Clasificación de los sensores según la variable física a medir | De posición, velocidad y aceleración |
| | De nivel y proximidad |
| | De humedad y temperatura |
| | De fuerza y deformación |
| | De flujo y presión |
| | De color, luz y visión |
| | De gas y pH |
| | Biométricos |
| | De corriente |

Fuente: (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, pág. 19)

Sensores de temperatura

Se usa con el fin de monitorear la temperatura en zonas críticas específicas del proceso se usa de manera redundante. Una de estas son las termocuplas que simboliza el sensor de temperatura más común utilizado industrialmente (ver la figura 3). Estos dispositivos están compuestos por dos alambres de distinto material, unidos en un extremo. En general, lo que hace este equipo es generar pequeñas tensiones por parte del sensor, que debe ser amplificadas para ser interpretadas correctamente, ya que al aplicarle temperatura producirán un pequeño voltaje en el orden de los milivoltios entre los extremos opuestos (Peña Palacios & Lechuga Sierra, 2008, pág. 36), (JM Industrial, 2023).



Figura 3 Tipos de sensores termocuplas

Sensor de temperatura LM35

Es un sensor monolítico de entrada analógica que tiene un rango de censado de la temperatura de -55°C hasta 150°C esto se traduce en valores de voltaje de 550mV hasta 1500 mV, como se puede ver en la hoja de datos del fabricante de la figura 4

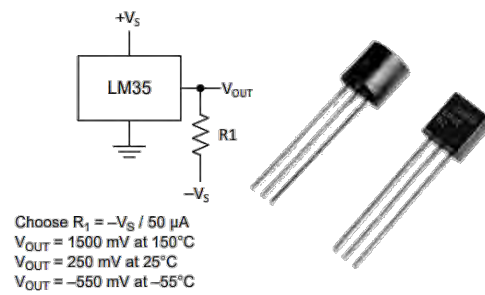


Figura 4 Sensor LM35

Fuente: (Texas Instruments, 2017, pág. 1)

Sensores de corriente

Un sensor de corriente es un dispositivo eléctrico que mide el flujo de corriente eléctrica a lo largo de una línea eléctrica, lo logra tomando una medida precisa de la caída de voltaje en una carga colocada en el camino de la corriente, básicamente, estos sensores se usan en aplicaciones de control automático la salida de este sensor puede ser análoga o usarse a manera de interruptor. En ese sentido, todo, estos sensores son de gran relevancia, ya que, ofrecen un grado de seguridad que permite garantizar el cuidado de nuestros equipos cuando en la línea haya una corriente mayor que pueda afectar el equipo (SDI, 2022). En la figura 5 se observa cómo es un sensor de corriente tipo industrial.



Figura 5 Sensor de corriente de núcleo cerrado

Fuente (PROCOEN, 2018)

Sensor de corriente LA 25-NP

Es un transductor de corriente. Cuenta con salida instantánea, precisión excelente, muy buena linealidad, baja desviación de temperatura, tiempo de respuesta optimizado. Este transductor de corriente se utiliza para la medición electrónica de corrientes (CD y AC pulsada, mixta). Se utiliza principalmente en controladores de velocidad variable AC, controladores de motor CD o AC, fuentes de alimentación no interrumpidas (UPS) y en fuentes de alimentación para aplicaciones de soldadura (Farnell, 2023). En la figura 6 se presenta el sensor de corriente LA 25-NP.

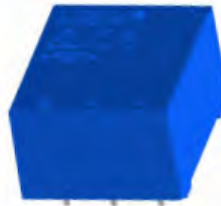


Figura 6 Sensor de corriente LA 25-NP de 25 At

A continuación, en la figura 7 se presenta la configuración eléctrica del sensor de corriente LA 25-NP.

| | | | | | | | |
|-----------|----------------------------------|----------------------------------|------------------------------------|----------------------------------|--------------------|--------------------|--------------------|
| I_{PN} | Primary nominal RMS current | 25 | At | | | | |
| I_{PM} | Primary current, measuring range | 0 ... ± 36 | At | | | | |
| R_M | Measuring resistance @ | $T_A = 70\text{ }^\circ\text{C}$ | | $T_A = 85\text{ }^\circ\text{C}$ | | | |
| | | | | $R_{M\text{ min}}$ | $R_{M\text{ max}}$ | $R_{M\text{ min}}$ | $R_{M\text{ max}}$ |
| | | with $\pm 15\text{ V}$ | @ $\pm 25\text{ At}_{\text{ max}}$ | 100 | 320 | 100 | 315 |
| | | | @ $\pm 36\text{ At}_{\text{ max}}$ | 100 | 190 | 100 | 185 |
| | | | | | Ω | | |
| I_{SN} | Secondary nominal RMS current | 25 | mA | | | | |
| N_p/N_s | Turns ratio | 1-2-3-4-5 | : 1000 | | | | |
| U_C | Supply voltage ($\pm 5\%$) | ± 15 | V | | | | |
| I_C | Current consumption | $10 + I_S$ | mA | | | | |

Figura 7 Características eléctricas del sensor de corriente LA 25-NP

La tabla 2 muestra la configuración y los rangos de corriente con su respectiva conexión.

Tabla 2 Configuración y conexión de la corriente nominal

| Number of primary turns | Primary current | | Nominal output current I_{BN} [mA] | Turns ratio N_p/N_s | Primary resistance R_p [mΩ] | Primary insertion inductance L_p [μH] | Recommended connections |
|-------------------------|-------------------------|----------------------|---|--------------------------|----------------------------------|--|-------------------------|
| | nominal I_{pn} [A] | maximum I_p [A] | | | | | |
| 1 | 25 | 36 | 25 | 1 / 1000 | 0.3 | 0.023 | |
| 2 | 12 | 18 | 24 | 2 / 1000 | 1.1 | 0.09 | |
| 3 | 8 | 12 | 24 | 3 / 1000 | 2.5 | 0.21 | |
| 4 | 6 | 9 | 24 | 4 / 1000 | 4.4 | 0.37 | |
| 5 | 5 | 7 | 25 | 5 / 1000 | 6.3 | 0.58 | |

Fuente: 3760314.pdf (farnell.com)

¿Qué es un actuador?

Un actuador es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía. Por lo regular los actuadores se clasifican en dos grandes grupos:

1. Por el tipo de energía utilizada: Actuador neumáticos, hidráulicos y eléctricos (ver la figura 8).
2. Por el tipo de movimiento que generan: actuador lineal y rotatorio

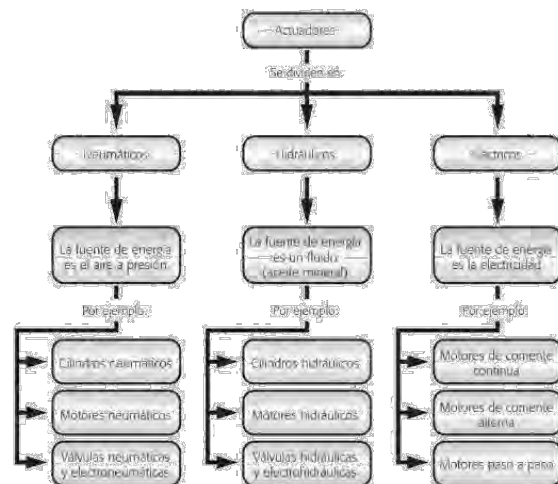


Figura 8 Clasificación de los actuadores

Fuente: (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, pág. 25)

Los actuadores neumáticos transforman la energía acumulada del aire comprimiendo en trabajo mecánico de movimiento rotatorio o movimiento rectilíneo y los hidráulicos es muy semejante, ya que ambos dispositivos que transforman la energía almacenada de un flujo a presión en trabajo mecánico de movimiento circular o rectilíneo, con la única excepción de que, en el caso de los actuadores hidráulicos, el flujo a presión que se utiliza no es el aire, sin algún tipo de aceite mineral (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, págs. 25,27).

Los actuadores eléctricos transforman la energía eléctrica en energía mecánica, ya sea rotacional o lineal. De los actuadores disponibles en el mercado, éstos son lo que se usan con mayor frecuencia (ver la figura 9), ya que su fuente de alimentación es la energía eléctrica, que es el tipo de energía que se encuentra disponible en la red de distribución eléctrica; por su parte, los actuadores que son alimentados con energía neumática o hidráulica requieren compresores para la generación de esta. (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2019, pág. 30)



Figura 9 Actuadores

2.1.2 Nivel de control

La función principal es la de control y gobierno de este. Recibiendo información del nivel 1, envía órdenes para que los elementos del equipo ejecuten sus funciones. Estamos hablando de los PLC y para conocer mejor.

¿Qué es un PLC?

Un Controlador Lógico Programable (PLC) es una máquina electrónica diseñada para controlar en tiempo real y en medio industrial procesos secuenciales (Aguilera Martinez, 2002, pág. 14).

Otra definición es que permite controlar un proceso o realiza una secuencia de acciones de manera automática a partir de un programa definido por el usuario. Esto lo realiza ejecutando en forma cíclica una secuencia de instrucciones que, a partir de la información que llega a sus entradas, desde los sensores, deciden cuándo conmutar sus salidas, donde se encuentran conectados los actuadores (Daneri, 2008, pág. 89).

Funciones básicas de un PLC:

- La detección: Lectura de la señal de los captadores distribuidos por el sistema de fabricación.
- El mando: Elabora y envía las acciones al sistema mediante los accionadores y preaccionadores.
- El diálogo hombre máquina: Mantener un diálogo con los operarios de producción obedeciendo sus consignas e informando del estado del proceso.
- Sistema de supervisión: También los autómatas permiten comunicarse con computadoras provistas de programas de supervisión industrial. En comunicación se realiza por una simple conexión por el puerto serie de la computadora.
- Control de procesos continuos: Además de dedicarse al control de sistemas de eventos discretos los autómatas llevan incorporadas funciones que permiten el control de procesos continuos. Disponen de módulos de entradas digitales y analógicas al igual que sus salidas a relevador y analógicas y la posibilidad de ejecutar reguladores PID que están programados en el autómata.

- Entradas/Salidas distribuidas: Los módulos de entrada/salida pueden estar distribuidos por la instalación, se comunican con la unidad central de autómeta mediante un cable de red.
- Buses de campo: Mediante un solo cable de comunicación se pueden conectar al bus de captadores y accionadores reemplazando el cableado tradicional.

(Aguilera Martinez, 2002, págs. 15-16)

Campo de aplicación

en ese aspecto sus actualizaciones son fundamentales en instalaciones donde es necesario un proceso de maniobra, control, señalización, etc. Por lo tanto, su aplicación abarca desde procesos tan complejos como lo es en las fábricas industriales hasta los procesos más simples como automatizar el hogar.

Una de las ventajas de los PLC'S

- No es necesario dibujar el esquema de contactos.
- Posibilidad de introducir modificaciones sin cambiar el cableado ni añadir aparatos.
- Mínimo espacio de ocupación.
- Menor costo de mano de obra en la instalación.
- Economía de mantenimiento. Además de aumentar la fiabilidad del sistema, al eliminar contactos móviles, los mismos autómetas pueden indicar y detectar defectos.
- Posibilidad de gobernar varias máquinas con un mismo autómeta.

(Aguilera Martinez, 2002, págs. 18-19)

Unidades de entrada y salida de datos

Generalmente los PLC disponen de dos tipos de entrada y salidas:

- Digital
- Analógica

Las E/S digitales no conducen señal alguna o poseen un nivel mínimo de tensión. Estas se manejan en nivel de bit dentro del programa de usuario.

Las E/S analógicas pueden poseer cualquier valor dentro de un rango determinado, se basan en convertidores A/D y D/A. Estas señales se manejan en nivel de byte o palabra (8/16) dentro del programa de usuario.

Interfaces

Todo PLC, salvo a algunos equipos, posee la virtud de poder comunicarse con otros dispositivos como HMI o computadora personal. Las entradas que tienen los PLC modernos son serie de tipo RS-232 / RS-422 / RS-485. Esto suele emplearse para la monitorización del proceso.

El PLC Delta

Los controladores lógicos programables de la serie DVP de Delta ofrece aplicaciones de alta velocidad, estables y altamente confiables en todo tipo de máquinas de automatización industrial. Además de una operación lógica rápida, instrucciones abundantes y múltiples tarjetas de función, el rentable PLC-DVP también admite varios protocolos de comunicación, conectado el accionamiento de motor de CA, servos, la interfaz hombre máquina (HMI) y el controlador de temperatura (DELTA, 2023). En la figura 10 se presenta los PLC Delta de la serie DVP.



Figura 10 PLC Delta de la serie DVP

Fuente: ID: 14427620830 (indiamart.com)

Software de programación ISPSOft

El ISPSOft es una aplicación (ver la figura 11) de software de programación altamente accesible para los controladores lógicos programables (PLC) de Delta. El ISPSOft es fácil de usar lo cual proporciona interfaces gráficas y asistentes convenientes para mejorar la accesibilidad del software, así como la flexibilidad de edición. ISPSOft ofrece hasta cinco herramientas de programación, incluyendo: diagrama escalera (LD), diagrama a bloques de funciones (FBD), gráfico de funciones secuenciales (SFC), lista de instrucciones (IL) y texto estructurado (ST), lo que permite a los usuarios elegir herramientas competentes para una mejor eficiencia de edición (DELTA, 2023).



Figura 11 ISPSOft software de programación

¿Qué es una HMI?

Es el acrónimo de Human Machine Interfaz (Interfaz Humano-Máquina). La definición de una interfaz humano-maquina es la de una interfaz de usuario o un panel de control que combina software y hardware para ayudar al operario a comunicarse entre sistemas y maquinas (SICMA21, 2021).

Las HMIs se pueden relacionar como una ventana de desarrollo de software en la cual se ven los mímicos de los objetos y elementos que conforman un proceso determinado. Las señales de campo o señales de proceso son conducidas hacía las HMI por medio de tarjetas de entradas y salidas de las computadoras, Controladores Lógicos Programables (PLC), unidades remotas de entradas y salidas (RUT) o variadores de velocidad (Drives). En ese sentido los sistemas HMI, conjunto con los PLC, suelen ser la columna vertebral de la línea de producción en las industrias, por lo tanto, la integración de la pantalla HMI en la fabricación ha mejorado enormemente las operaciones. Permite el control de supervisión y la adquisición de

datos en todo el sistema (Rodríguez de Avila, 2012, pág. 12). A continuación, se menciona las principales funciones que tiene una HMI.

- Monitoreo: Es la habilidad de poder mostrar las variables de la planta en tiempo real y pueden ser mostradas en números, textos y gráficos lo cual pueden ayudar al buen entendimiento del operador.
- Supervisión: Esta función junto con la anterior, pueden lograr ajustes en las condiciones de trabajo directamente desde la computadora.
- Alarmas: Es la capacidad de reconocer eventos excepcionales dentro del proceso y reportarlos, basándose en los límites de control preestablecidos por los ingenieros de diseño.
- Control: Es asociada la capacidad de aplicar algoritmos que ajustan los valores de procesos a ciertos límites permisibles.
- Histórico: Es la capacidad de muestrear y almacenar archivo, datos de proceso a una determinada frecuencia, que se convierte en una herramienta fundamental y muy utilizada para la optimización y corrección de proceso.

(Rodríguez de Avila, 2012, pág. 12)

Para proponer una correcta visualización del proceso es importante y pertinente entender las condiciones y comportamientos de este, los expertos han estandarizado el uso de las HMI en cuatro niveles distintos.

- ✓ Nivel 1: Área de procesos de información general (para conocimiento de la situación).
- ✓ Nivel 2: Proceso de control de la unidad (para la manipulación de procesos en recurso).
- ✓ Nivel 3: Proceso de detalle de la unidad (para un examen minucioso y detallado del proceso).
- ✓ Nivel 4: Proceso de mantenimiento de la unidad y de diagnóstico (para la solución de problemas).

Los sistemas de gráficos de procesos deben utilizar banderas para mostrar la ocurrencia de las condiciones anormales del proceso, animaciones, colores y parpadeo de luces para

reconocer rápidamente el significado del evento (Rodríguez de Avila, 2012, pág. 13). A continuación, se presenta la tabla 3 los colores aplicables según el estado de operación.

Tabla 3 Colores de operación

| | |
|----------|------------------------------------|
| ROJO | Paro, alarma, peligro, prohibición |
| AMARILLO | Espera, listo, prealarma |
| VERDE | Marcha, correcto, sin defectos |
| AZUL | Mando, acción |

Fuente: (Rodríguez Penin, 2007, pág. 58)

Pantalla HMI Delta de la serie DOP

Delta además que ofrece PLC'S también comercializa pantallas HMI (ver la figura 12) de alta calidad, son dispositivos capaces de controlar y monitorear datos, de manera más que intuitivas, dando soluciones como:

- ✓ Detectar fallas en tiempo real (avisa a los usuarios correspondientes).
- ✓ Limita al usuario no deseados a cambiar valores no autorizados.
- ✓ Adquiere y guarda datos en un Excel para generar reportes.
- ✓ Compatibilidad y comunicación transparente con diversas marcas.

Aplicaciones:

- Toma el control de uno o varios Servos drives Delta sin la necesidad de un PLC.
- Desde tu controlador visualiza y controla tus sensores y/o actuadores.
- Mantiene informado de lo que sucede vía remota en tu proceso.



Figura 12 PLC Delta de la serie DOP-100

2.1.3 Nivel de supervisión

La garantía de la flexibilidad y la reconfigurabilidad en el proceso para bajar la inteligencia a nivel de piso de planta. Es por ello que el supervisor debe ser concebido como un sistema distribuidor inteligente, el cual supervisa el control regulatorio de los equipos y coordina los diferentes, es decir, unidades, subprocesos, celdas de trabajo de los diferentes tipos de procesos como las unidades de producción y las líneas de trabajo, respectivamente, así como gestionar los recursos de cada uno de ellos asociados al flujo de producto y al cumplimiento de la receta. Todo esto es la concepción del sistema SCADA (Cerrada, 2011, pág. 6).

En relación con eso los sistemas de nivel de supervisión sistemas SCADA permite el acceso a datos remotos de un proceso, utilizando las herramientas de comunicación necesarias en cada caso, el control de este, no se trata de un sistema de control, sino de una utilidad software de monitorización o supervisión, que realiza la tarea de interfases entre los niveles de control PLC y los de gestión a un nivel superior (Rodríguez Penin, 2007, pág. 36)

Otros dicen que un sistema SCADA es una aplicación de software especialmente diseñada para funcionar sobre ordenadores en el control de producción que proporciona comunicación entre los dispositivos de campo, llamados también RTU (Remote Terminal Units o Unidades Remotas), donde se pueden encontrar elementos tales como controladores autónomos o autómatas programables, y un centro de control o unidad central (MTU, Master Terminal Unit), en el que se controla el proceso de forma automática desde la pantalla de uno o varios ordenadores (Rodríguez Penin, Sistema SCADA 3a edición, 2012, pág. 30)

Una de las ventajas de este sistema son los siguientes:

- Sencillez en la instalación, sin exigencia de hardware, interfases amigables lo cual hace fáciles de utilizar.
- Funciones de mando y supervisión integradas.
- Comunicación flexible para comunicarse con facilidad de forma transparente al usuario con el equipo de planta.
- Capacidad de crecer o adaptarse según a las necesidades de la empresa.

Los sistemas SCADA su objetivo es supervisar y mandar lo cual ayuda a economizar, en el sentido de ver lo que ocurre en la instalación desde la planta u oficina que enviar a un operario a realizar dicha tarea, haciendo que algunos inconvenientes se vuelvan innecesarias. Otra aportación importante es la accesibilidad supongamos que la empresa de empaque de limón para su exportación cuenta con varias bandas transportadoras, sensor de temperatura, variadores de frecuencia, de corriente etc. Todo lo que engloba esta parte, lo tenemos al alcance de un solo clic. Será posible modificar los parámetros de funcionamiento de cada motor que den señales anormales en su hora de funcionamiento con la facilidad de poner fuera de servicio sin necesidad de perjudicar los demás sistemas que están en proceso. La conectividad de esta red busca sistemas abiertos por un protocolo de interconexión a sistemas de diferentes proveedores y evitar la existencia de lagunas informativas que puedan causar fallos en el funcionamiento.

Los sistemas SCADA de mayor o menor complejidad en su Interfase Hombre maquina (HMI) cumple con toda la serie de funciones a establecer una comunicación lo más clara y sencilla posible haciendo que no tenga alguna experiencia alguna para que exista un lazo posible entre el proceso y el operador (ver la figura 13)

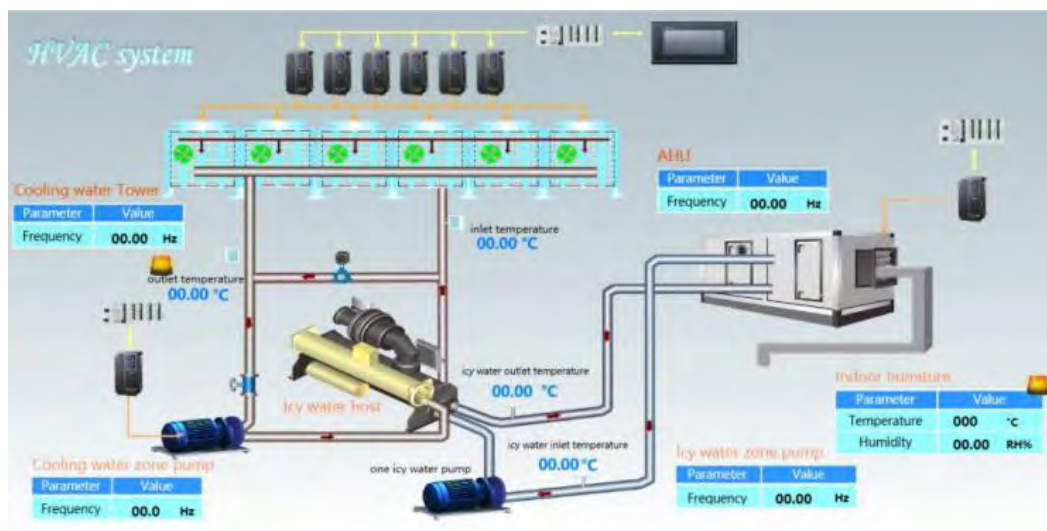


Figura 13 línea de un sistema (realizado en Delta DIAView SCADA System)

Fuente: (DELTA, 2018)

Arquitectura básica del sistema SCADA

Una vez entendida la distribución de control en la planta vista previamente en la pirámide de la automatización. De esta manera, el sistema queda dividido en tres bloques principales (ver la figura 14).

- Software de adquisición de datos y control (Scada).
- Sistemas de adquisición y mando (sensores y actuadores).
- Sistema de interconexión (comunicaciones).

El sistema de proceso capta el estado del sistema a través de los elementos sensores e informa al usuario a través de las herramientas HMI, basándose en los comandos ejecutados por el usuario (Rodríguez Penin, 2012, pág. 29).

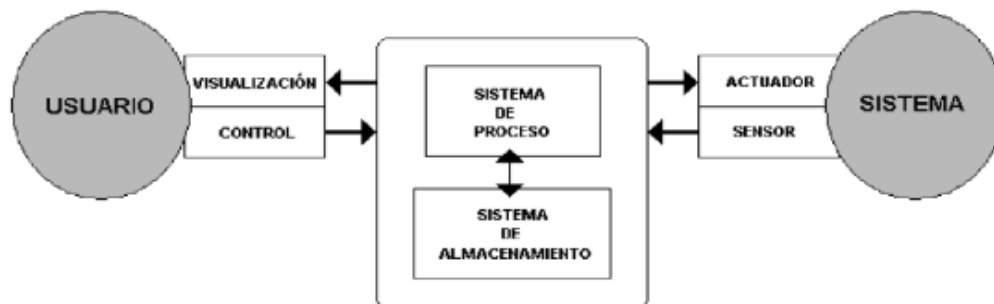


Figura 14 Sistema SCADA, arquitectura básica

Fuente: (Rodríguez Penin, 2012)

Aplicaciones de sistemas SCADA

Los SCADA han sido utilizados inicialmente en el ámbito industrial para controlar los procesos productivos, pero en los últimos años también se ha extendido entre empresas del sector agroalimentario, farmacéuticos, energético. El sistema SCADA funciona bien en diferentes tipos de empresas porque puede abarcar desde configuración simples hasta instalaciones grandes y complejas. Prácticamente puede aplicarse a cualquier proceso desde el sector petroquímico hasta tratamiento de aguas, telecomunicaciones, transporte, gas, hotelería, en el hogar, etc (WEG AUTRIAL, 2019).

2.1.4 Nivel de planeación

Este nivel emite los programas hacia el nivel 2 y recibe de éste las incidencias de la planta de tal forma, que se realice un análisis del comportamiento de la planta y se tomen decisiones al respecto. Esta información se comunica al nivel superior de la pirámide donde se hace la planeación de los recursos de la empresa. Lo cual en este grado permite la planificación de la producción, facilita la ingeniería de procesos, al dar a los responsables información global actualizada de todo el proceso productivo. Las herramientas de ingeniería en este nivel están dirigidas a la optimización de activos, eficiencia de ingeniería y sistemas de administración de la información (Lozano Lozano & Zamora Muñoz, 2008, pág. 29).

Básicamente en este nivel se realizan las siguientes tareas:

- Gestión de mantenimiento
- Gestión de materiales
- Control de inventarios
- Gestión de compras

El sistema de ejecución de manufactura (MES) es una herramienta tecnológica que se utiliza para crear soluciones ligadas al proceso de producción de la planta. Los sistemas proveen datos en tiempo real para que el control de procesos sea más eficiente. Los sistemas de ejecución de manufactura reúnen información de los procesos de ventas/pedidos, proveen instrucción a los operadores y a los equipos, y comunican el estado de la producción, consumo de material, control de calidad, e información de mantenimiento, que retorna al equipo administrativo (Lozano Lozano & Zamora Muñoz, 2008, pág. 30).

2.1.5 Nivel de gestión

El nivel de gestión y de planeación están englobados como un mismo sistema. Esto se debe a la frecuencia de actualización de la información sobre clientes, proveedores, contabilidades, costes, gestión, proyectos, etc (WITORG, 2023), (Lozano Lozano & Zamora Muñoz, 2008, pág. 33).

En este nivel se caracteriza por estar constituido por computadoras donde se administra toda la información de planeación y gestión de la empresa. A continuación, se presenta las siguientes tareas realizadas en este nivel:

- Gestión comercial y de marketing
- Planeación estratégica
- Financiera y administrativa
- Gestión de recursos humanos
- Ingeniería de producto
- Ingeniería de proceso
- Investigación y desarrollo

Los sistemas de planificación de recursos de la empresa (en inglés ERP) son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa. El propósito fundamental de un ERO es otorgar apoyo a los clientes del negocio, tiempos rápidos de respuesta a sus problemas, así como un eficiente manejo de información que permita la toma de decisiones y disminución de costo totales de operación (Lozano Lozano & Zamora Muñoz, 2008, págs. 32-33).

CAPÍTULO 3. PROGRAMACIÓN DEL PLC

Como se había mencionado en el capítulo 2 los PLC'S Delta usan el software ISPSoft y este permite programar en cinco modos diferentes. En este caso la programación se realizó en lenguaje de diagrama escalera (LD) para una programación rápida y sencilla. El Software ISPSoft dispone de una variedad de instrucciones básicas: temporizador, contadores, operadores, comparadores, operacionales de punto flotante, control de posicionamiento, etc.

Entre algunas de las funciones que puede ofrecer ISPSoft se mencionan las siguientes:

- Transferir el programa directamente en los controladores con el puerto USB de PC al PLC Delta.
- Para modelos de PLC DVP20SX2 se puede elegir Entradas/Salidas (E/S) físicas e internas tanto como digitales como analógicas.
- Se puede realizar la simulación del programa realizado.
- Posibilidad de modificar la programación a nivel de que el sistema vaya evolucionando.

3.1 INSTALACIÓN DEL SOFTWARE ISPSoft

Para la instalación del software ISPSoft se necesitan las siguientes características técnicas mínimas:

- Windows® XP / Vista 7 de 32 Bits / 64 Bits
- Windows 10 de 32 Bits / 64 Bits

Seguidamente se presenta los pasos para la instalación:

Paso 1: Buscar en Google: "Download ISPSoft" y seleccionar el primer enlace el cual te envía a la página oficial de Delta. Debajo de la descripción del programa se muestra el enlace para descargar el software "Software Download".

Paso 2: En el lado izquierdo ubicar Filter y elegir la opción de software (ver la figura 15).

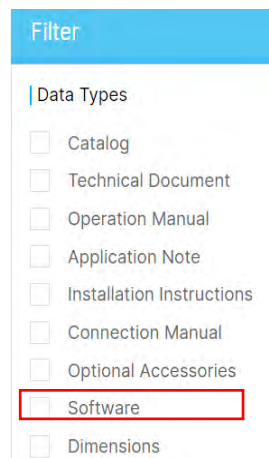


Figura 15 Selección de paquetes de Software de Delta

Paso 3: Ubicar el software ISPSOFT y seleccionar la versión más actualizada como se muestra en la figura 16.



Figura 16 Download ISPSOFT V3.17

Paso 4: Ubicar en la carpeta de descargas el archivo **ISPOFT 314.exe**, posteriormente clic derecho y ejecutar como administrador e instalar. Se siguen los pasos mostrados en la figura 17.

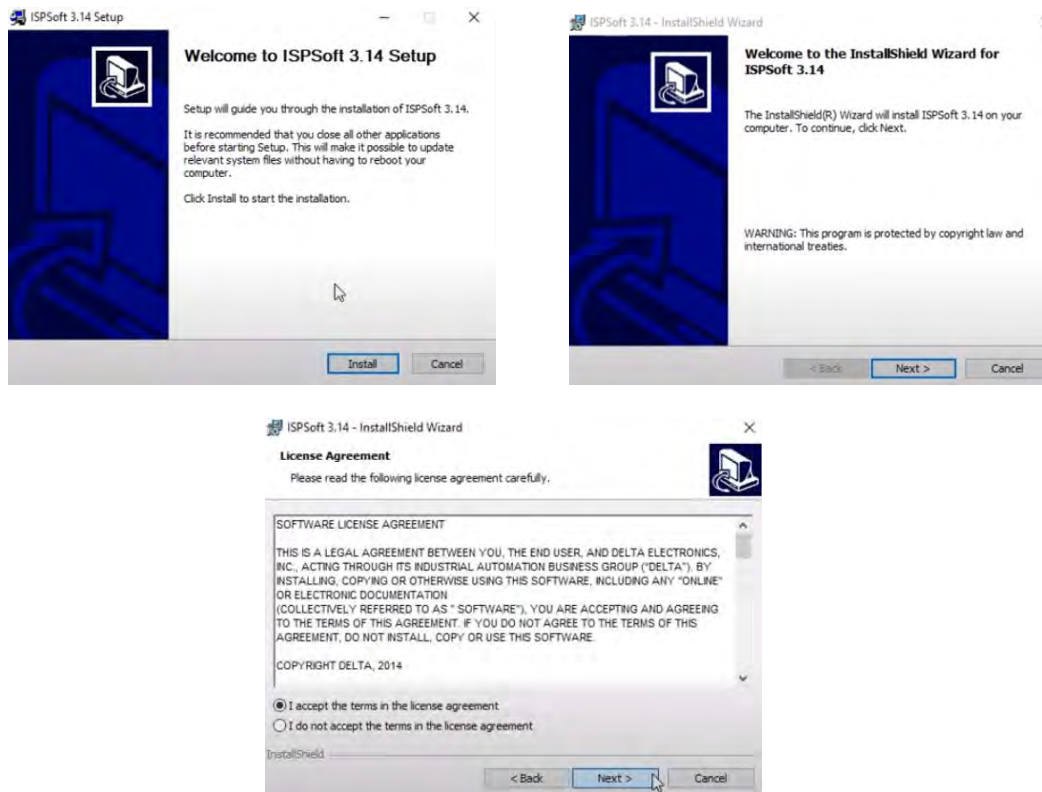


Figura 17 Asistente de instalación de ISPSOft.

Paso 5: Se abrirá una ventana solicitando la información del usuario, en este apartado se asigna el nombre y la organización (ver la figura 18).



Figura 18 Información del usuario

Paso 6: El asistente de instalación mostrara una serie de ventanas, hay que aceptar y finalizar la instalación.

Paso 7: Ejecutar como administrador el siguiente programa que se instaló previamente. Posteriormente seleccionar la ubicación de la instalación (ver la figura 19).

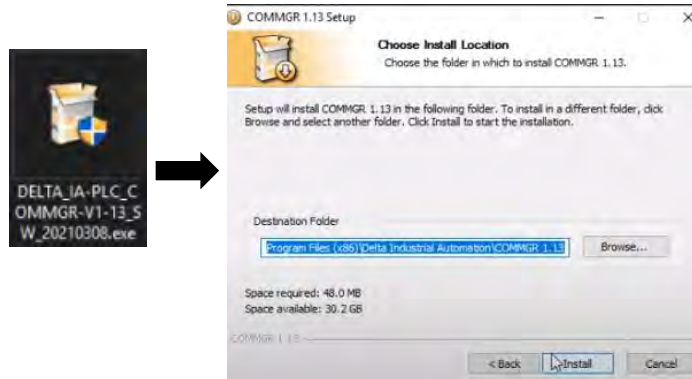


Figura 19 Instalación del COMMGR y ubicación del programa

Paso 8: Se finaliza la instalación.

3.2 INICIAR EL SOFTWARE ISPSOFT

Localizar el icono de ISPSOft, (ver la figura 20) y accede al programa.



Figura 20 Icono de ISPSOft en el escritorio

Para crear un proyecto dar clic en **Archivo**, y seleccionar **Nuevo**, como se muestra en la figura 21.

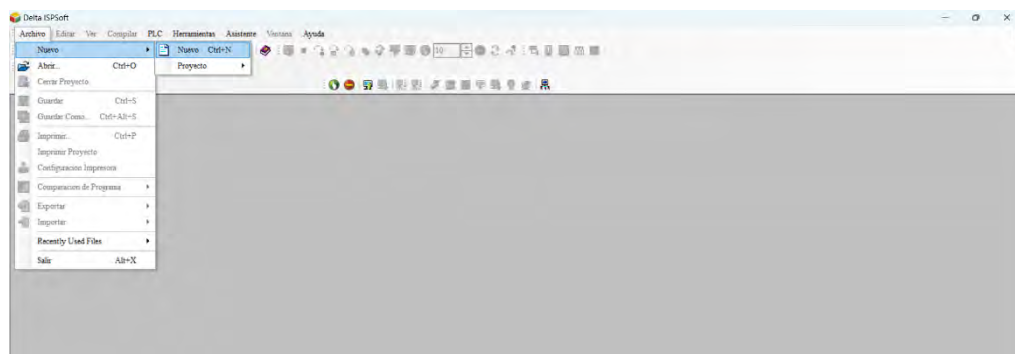


Figura 21 Generar un nuevo proyecto

Se abrirá un recuadro seleccionando el tipo de controlador y el modelo del PLC, en ese caso se necesita las características del PLC, el dato se obtiene de la siguiente forma (ver figura 22).



Figura 22 PLC modelo DVP20SX2 (Izquierdo) y PLC modelo DVP-14SS2 (derecho).

Cada una de las siglas del modelo del PLC indican una característica del dispositivo. Por ejemplo, para el **PLC DVP20SX2**, las siglas indican los siguientes datos:

- ✓ El **DVP** significa la serie del PLC.
- ✓ El **20** significa puntos MPU, en otras palabras 20 entradas de múltiples procesos (8DI/6DO 4IA/2AO).
- ✓ El **SX2** es el modelo del PLC.

Para el PLC modelo **DVP-14SS2** se tiene lo siguiente:

- ✓ El **DVP** significa la serie de PLC.
- ✓ El **14** significa puntos MPU de múltiples procesos (8DI/6DO) este dato se ve en las características del PLC.
- ✓ El **SS2** es el modelo de PLC.

Identificando el modelo, en la configuración del programa también debe coincidir los datos del PLC, como se muestra en la figura 23, se elige la serie y el modelo del PLC.

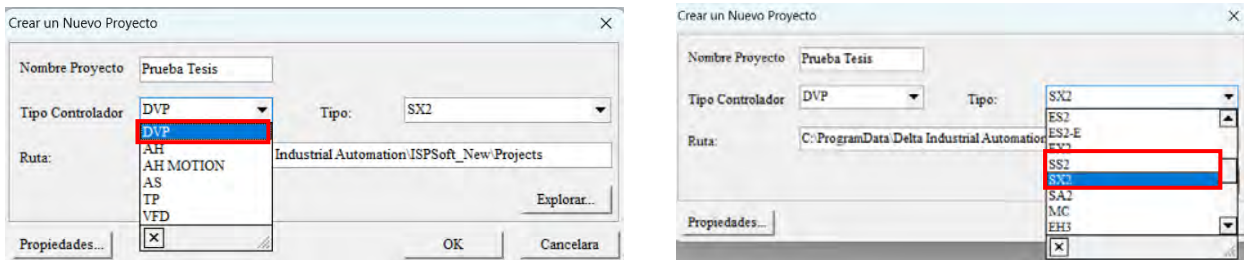


Figura 23 Elijiendo la serie y el modelo del PLC

Luego se generará un nuevo proyecto, dar clic derecho donde dice **Programa** y seleccionar nuevo. Asignar un nombre de proyecto en **Nombre de POU**. El programa tiene la posibilidad de protegerse con una contraseña por seguridad, el tipo de **Lenguaje** es (LD) lo demás permanece como se muestra (ver la figura 24)

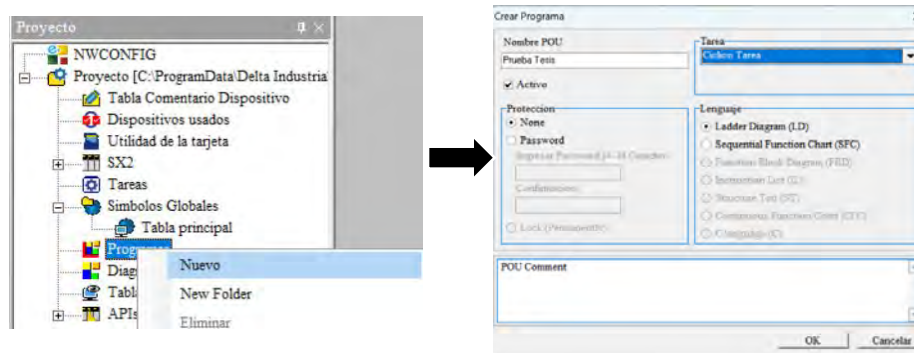


Figura 24 Generación de proyecto (LD)

Finalmente, ya está listo para programar en (LD) (ver la figura 25)

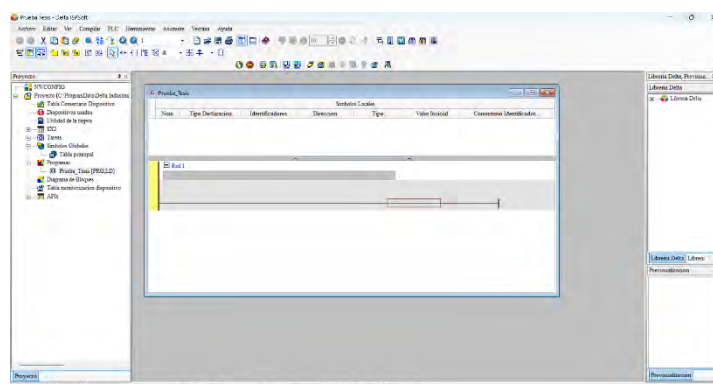


Figura 25 Programa LD en ISPSOFT

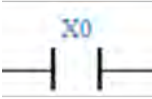
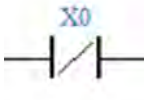
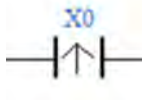
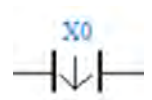
3.3 PROGRAMACIÓN DEL PLC

3.3.1 Estructura del diagrama escalera.

Los elementos básicos del lenguaje LADDER, también denominado lenguaje de contactos o de escalera; es un lenguaje de programación gráfico muy popular dentro de los Controladores Lógicos Programables, esto se debe a que está basado en los esquemas eléctricos de control clásico. De esta manera, con los conocimientos necesarios que todo técnico electricista posee, puede programar de manera fácil y adaptarse a la programación de este tipo de lenguaje. La principal ventaja de este lenguaje es que los símbolos básicos que se utiliza están normalizados según las normas NEMA (National Electrical Manufacturers Association) y son empleados por todos los fabricantes (Ba Villarreal, 2021, pág. 39)

En la tabla 4, se puede observar los símbolos de los elementos básicos junto con sus respectivas descripciones.

Tabla 4 Elementos básicos del lenguaje LADDER (LD)

| Símbolo | Nombre | Descripción | Variable aceptable | Tipo |
|---|---|--|--|---------------------------|
|  | Contacto Normalmente Abierto (NA) | Su función es ON/OFF (activando/desactivando), su estado es de (0 a 1) lo cual se activa cuando hay un uno lógico en la variable que representa. | Entrada (X), salida (Y), memoria o relé interno (M), relé escalera (S), temporizador (T) y contador (C). | Bool Timer Contador |
|  | Contacto Normalmente Cerrado (NC) | Su función es ON/OFF (activando/desactivando), su estado es de (1 a 0) lo cual se activa cuando hay un cero lógico en la variable que representa. | Entrada (X), salida (Y), memoria o relé interno (M), relé escalera (S), temporizador (T) y contador (C). | Bool Timer Contador |
|  | Interruptor de disparador de franco ascendente (NO) | Es un simple contacto normalmente abierto (NO) cuando el franco detecta un cambio en la variable es cuando permite el paso de corriente en solo ese instante en este caso va de (0 a 1). | Entrada (X), salida (Y), memoria o relé interno (M), relé escalera (S), temporizador (T) y contador (C). | Bool |
|  | Interruptor de disparador de franco | Es un simple contacto normalmente abierto (NO) cuando el franco detecta un cambio en la variable es | Entrada (X), salida (Y), memoria o relé interno (M), relé | Bool |

El orden de ejecución de la programación de ISPSOft es de arriba hacia abajo y de izquierda a derecha. Primeramente, van los contactos luego posiblemente un bloque de control y por ultimo las bobinas (ver la figura 26).

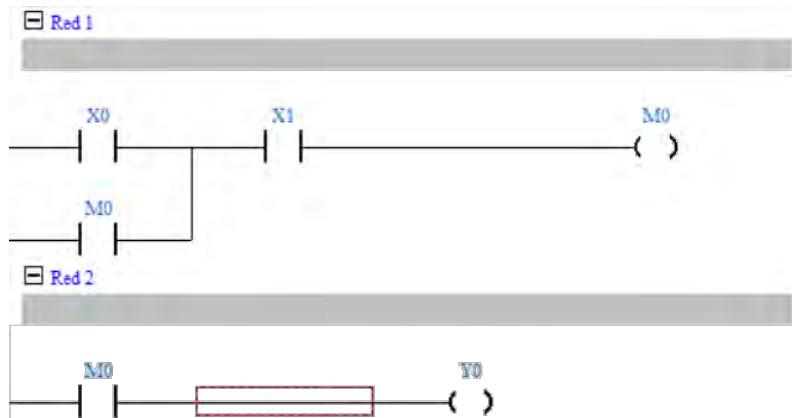


Figura 26 Formato de programación lenguaje escuadra (LD) en ISPSOft

3.3.2 Tipos de bloques de control

Es un lenguaje de programación gráfico que permite al usuario programar elementos (bloques de control y de funciones del PLC) que se interconectan al igual que un circuito eléctrico. Generalmente se utilizan instrucciones o símbolos lógicos para representar al bloque.

En ese sentido las instrucciones API, son las instrucciones de programación del PLC DVP20SX2 (ver la tabla 5). Se observa el numero API de instrucción, el código mnemónico único central, el prefijo D indica instrucciones de 32 bits y el prefijo P indica una instrucción de pulso, la función y los escalones ocupados puede ser de 16 bits/32 bits/ pulso.

Tabla 5 Formato de instrucciones de programación API

| API | Mnemónico | Operandos | Función | Controladores | | | | | | | | | | | | | |
|------------|---------------------|-----------|---------|---------------------|-------------------------|---|-----|-----|-----|-----|---|---|-----------------------|---|---|---|---------------------------|
| 12 | D MOV P | (S) (D) | Mover | ES2/EX2 SS2 EX2 SX2 | | | | | | | | | | | | | |
| Tipo OP | Dispositivos de bit | | | | Dispositivos de palabra | | | | | | | | Escalones de programa | | | | |
| | X | Y | M | S | K | H | KnX | KnY | KnM | KnS | T | C | | D | E | F | |
| S | | | | | * | * | * | * | * | * | * | * | * | * | * | | MOV, MOV P: 5 escalones |
| D | | | | | | | * | * | * | * | * | * | * | * | * | | DMOV, DMOV P: 9 escalones |

Fuente: Manual PLC Delta DVP-ES2-EX2-SS2-SA2-SX2-SE-TP.pdf - Google Drive.

En el anexo 2 se encuentran las listas numéricas de instrucciones API, ahora con el código mnemónico en ISPSOft se programa los bloques de control lo que permite al usuario

construir sus propios bloques, de acuerdo con sus requerimientos. En la figura 27, se muestra el mismo ejemplo corroborando el formato de instrucción del manual del PLC Delta. En el programa de ISPSOft el numero API es 12 y el nombre mnemónico es MOV, con sus variantes dependiendo del tipo de variable: DMOV, DMOVP, MOV y MOVP pertenece al mismo número API, pero con diferentes escalones de bit.

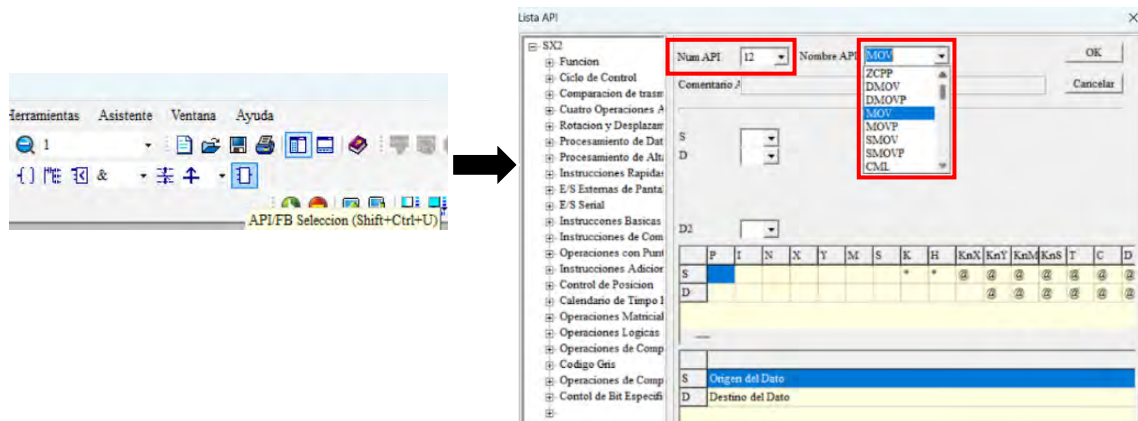


Figura 27 (izquierdo) símbolo para ubicar las instrucciones API/FB, (derecho) lista API

A continuación, se presenta un ejemplo de programación mediante bloques de función (ver la figura 28).

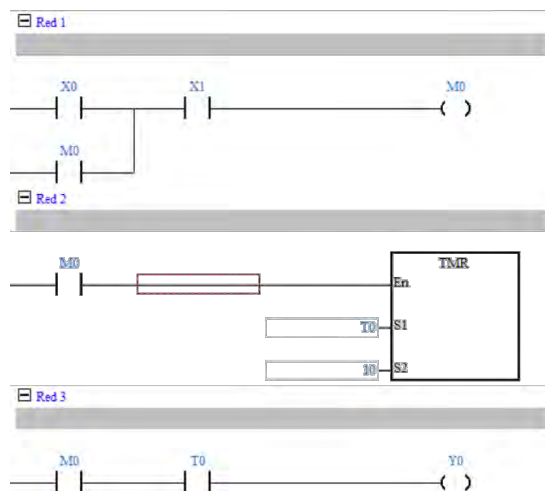


Figura 28 Ejemplo bloques de función en programación de lenguaje LD

3.3.3 registros y Relés de PLC Delta

Entre los registros existen tipos de identificadores. En la serie DOV de delta se puede agregar direcciones, como **Relé de entrada (X)** que es una variable tipo **BOOL**, donde aplican sensores que se accionan como (ON/OFF). El PLC SX2 cuenta con entradas físicas X0~X7 (ver la figura 29). Las salidas físicas son Y0~Y5 (ver la figura 29) su dirección de **Relé de salida (Y)** es una variable de tipo **BOOL** a relevador donde se conectan los actuadores como motores, válvulas servos, cilindros, bombas, etc.

El PLC cuenta con relevadores virtuales y cada uno cuenta con direcciones internas, las cuales se presentan en el anexo 1. Este tipo de relevadores se le conoce como **Relé auxiliar (M)** y son de tipo **BOOL**. Sus direcciones cuentan con registros especiales y son utilizados para distintas funciones, entre ellas como memoria o como una variable principal que manipule el funcionamiento de algún actuador o sensor por medio de la pantalla HMI. Las variables de entrada como la de salida pueden ser programados por un **temporizador (T)**, que ayuda a controlar procesos por medio de tiempos o de pulso, la dirección **Contador (C)** aplica en procesos que por ende se requiere conteo y finalmente la dirección **Registro de datos (D)** es una variable que puede ser de tipo **WORD, DWORD O REAL**. Básicamente se registran datos de números enteros positivos y números enteros negativos al igual que números de punto flotante. Esta dirección cuenta con registros especiales que servirá para programar la pantalla HMI y las entradas analógicas.

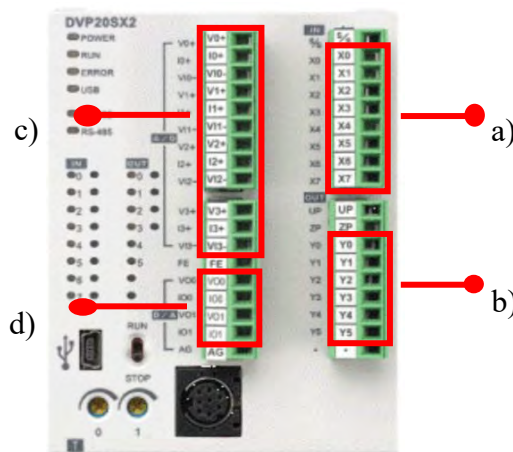


Figura 29 a) Entradas físicas del PLC X0-X7, b) Salidas físicas del PLC Y0-Y5, c) Entrada analógica lectura V+ entrada a voltaje, I+ entrada a corriente y VI- entrada común (tierra), d) salidas analógicas.

3.3.4 Registros de entradas y salidas analógicas

El PLC cuenta con cuatro entradas analógicas (IA) y dos salidas analógicas (OA) de modo que se puede conectar sensores que trabajen con variables como: temperatura, corriente, voltaje, presión, etc. Cada sensor tiene una lectura que puede ser de modo voltaje o corriente según la configuración del fabricante. En la figura 30 se visualiza las entradas y salidas físicas con su respectivo registro especial **D**, en la tabla 6 están los canales de E/S analógicas, para los modelos de PLC EX2/SX2.

Tabla 6 Registro especial para entradas y salidas analógicas modelos EX2/SX2.

| Registro de datos (entrada especial) | Canal |
|--------------------------------------|----------------|
| D1110 | Canal 0 (AD 0) |
| D1111 | Canal 1 (AD 1) |
| D1112 | Canal 2 (AD 2) |
| D1113 | Canal 3 (AD 3) |
| D1116 | Salida 0 |
| D1117 | Salida 1 |

Los registros D1110~D1113 corresponden a los puertos de entrada analógicos y los registros D1116~D1117 a las salidas analógicas del PLC, como se muestra en la figura 30.



Figura 30 Registros analógicos con su respectiva entrada y salida

3.3.5 Registros de condiciones y parámetros de entrada y salida analógica

Para especificar ciertas condiciones es necesario ocupar el registro **D1114**, el cual Habilita/Deshabilita las entradas analógicas vistas anteriormente. Por otro lado, el registro **D1115** define el tipo de señal Voltaje/Corriente. Esto se puede observar en la tabla 7.

Tabla 7 Palabra D1114 y D1115 información sintetizada

| Registro D | Función |
|--------------|--|
| D1114 | Habilitar/Deshabilitar entrada y salida de donde “0” es habilitado y “1” deshabilitado. |
| D1115 | Modo de lectura “0” modo voltaje y “1” modo corriente. Para modo corriente las lecturas pueden ser de (-20 mA a 20 mA) “0” y “1” (4 mA a 20 mA). |

El registro D1114 consta de 16 bits (ver la tabla 8) por lo tanto, el bit 0 representa la entrada analógica 0, el bit 1 a la entrada analógica 1 y así sucesivamente. En este caso el modelo del PLC DVP20SX2 consta de 4 puertos analógicos de entrada, por lo tanto, solamente es necesario usar los cuatro primeros bits (0-3) del registro mencionado. Entonces, para habilitar un puerto se usa un “0” lógico y un “1” lógico desactiva el puerto. En la tabla 8 se observa que todas las entradas analógicas se encuentran activadas.

Tabla 8 Todas las entradas analógicas habilitadas

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | IN3 | IN2 | IN1 | IN0 | Hex | Dec |
|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Haciendo referencia a la configuración de la tabla 8, el numero binario 0000 0000 0000 0000 a su forma decimal es 0 y ese valor se mueve al registro D1114. En la figura 31 se puede observar cómo se realizaría la programación en ISPSOft. Para la configuración de los registros mencionados.

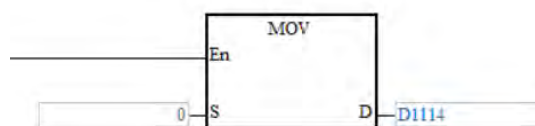


Figura 31 Instrucción MOV para el registro D1114 con todas las entradas analógicas activadas

La configuración depende del tipo de aplicación, por ejemplo, en la tabla 9 se muestra la configuración si se requiere que la entrada 4 y 3 estén deshabilitadas.

Tabla 9 Ejemplo de dos entradas deshabilitadas IN2(registro especial D1112) y IN3 (registro especial D1113)

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | IN3 Bit 3 | IN2 Bit 2 | IN1 Bit 1 | IN0 Bit 0 | Hex | Dec |
|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|--------------|--------------|--------------|--------------|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | C | 12 |

Por lo tanto, el numero binario 0000 0000 0000 1100 su equivalente decimal es 12 por lo tanto se mueve al registro D114 (ver la figura 32) programación en ISPSOft.

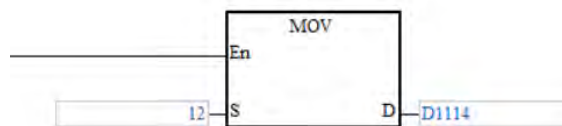


Figura 32 Instrucción MOV para el registro D1114 con dos entradas deshabilitadas (IN2 y IN3) y dos entradas habilitadas (IN0 y IN1)

Por otro lado, el registro D1115 que define el parámetro de señal a recibir en los puertos de E/S, también tiene un tamaño de 16 bits. En este caso, el bit 0 al bit 5 indican el tipo de señal que se va a recibir, esto lo determina el dispositivo o sensor conectado a los puertos de E/S. entonces para indicar una señal de voltaje se usa un "0" lógico y para corriente un "1" lógico. Si las configuraciones de los puertos se establecen en modo corriente se debe definir el rango de operación de estos. En este caso hay dos rangos para señales de corriente: el primero de **-20 mA a 20 mA** y el segundo **4 mA a 20 mA**. Si se requiere el primer rango de operación los puertos deben establecerse con un "0" lógico y si se requiere el segundo se establecen con un "1" lógico aplica del bit 8 al bit 13.

Tabla 10 Configuración de lectura entrada v salida a modo voltaje

| Bit 15 | Bit 14 | OUT1 Bit 13 | OUT0 Bit 12 | IN3 Bit 11 | IN2 Bit 10 | IN1 Bit 9 | IN0 Bit 8 | Bit 7 | Bit 6 | OUT1 Bit 5 | OUT0 Bit 4 | IN3 Bit 3 | IN2 Bit 2 | IN1 Bit 1 | IN0 Bit 0 | Hex | Dec |
|--------|--------|----------------|----------------|---------------|---------------|--------------|--------------|-------|-------|---------------|---------------|--------------|--------------|--------------|--------------|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Modo corriente (-20 mA a 20 mA) "0" (4 mA a 20 mA) "1"

Entrada a Voltaje "0" Corriente "1"

Haciendo referencia a la tabla 10 el número binario 0000 0000 0000 0000 a su forma decimal es 0 ese valor se mueve al registro D1115. En la figura 33 se puede observar cómo se realizaría la programación en ISPSOft. Para la configuración de los registros mencionados

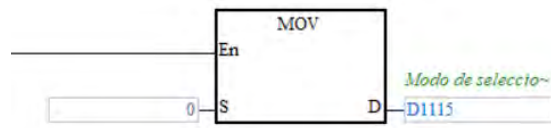


Figura 33 Instrucción MOV de las entradas y salidas con lecturas a modo Voltaje

Las entradas especificadas en la tabla son de referencia para la palabra D1115 la entrada analógica IN2 y IN3 están configurado a modo corriente mientras las entradas IN1 y IN0 a modo voltaje. En ese sentido en la tabla 11 en los 16 bits en la parte superior se especifica IN0 (bit 0) y con su correspondiente IN0 (bit 8), In1 (bit 1) con IN1 (bit 9) así sucesivamente con sus E/S para una mejor visualización. Ahora en la entrada IN2 (bit 2 con su correspondiente IN2 (bit 10) está configurado de -20 mA a 20 mA mientras que la entrada IN3 (bit 3) con su correspondiente IN3 (bit 11) está configurado de 4 mA a 20 mA.

Tabla 11 Configuración de lectura Voltaje/Corriente

| OUT1 | | OUT0 | | IN3 | | IN2 | | IN1 | | IN0 | | OUT1 | | OUT0 | | IN3 | | IN2 | | IN1 | | IN0 | | Hex | Dec |
|--------|--------|--------|--------|--|--------|--|-------|-------|-------|---------------------------|-------|-------------------------|-------|-------------------------|-------|----------------------------|------|----------------------------|--|-----|--|-----|--|-----|-----|
| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Hex | Dec | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 80C | 2060 | | | | | | | | |
| | | | | Entrada D1113 lectura de (4 mA a 20 mA) " 1" | | Entrada D1112 lectura de (-20 mA a 20 mA) " 0" | | | | Salida D1117 Voltaje " 0" | | Entrada D1113 Corriente | | Entrada D1112 Corriente | | Entrada D1111 Voltaje " 0" | | Entrada D1110 Voltaje " 0" | | | | | | | |

Siguiendo con la configuración de la tabla 11 con las condiciones de lectura de entrada los parámetros de corriente se obtiene el siguiente número binario 0000 1000 0000 1100 su número decimal es 2060 por lo cual en la figura 34 se puede observar cómo se realizaría la programación en ISPSOft.

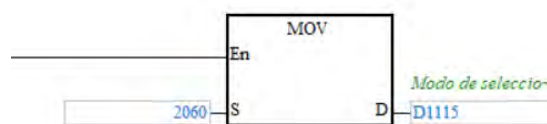


Figura 34 Instrucción MOV de las E/S con lecturas a modo Voltaje/Corriente

3.3.6 Tipos de datos

En ISPSOft se tiene varios tipos de variables. Cada una de ellas tiene un límite inferior y un límite superior, esos parámetros están determinados por el tamaño en bits de la variable. Esto se puede observar en la tabla 12.

Tabla 12 Tipos de variables

| Tipo de dato | Límite inferior | Límite superior | Bit |
|-----------------|---------------------|---------------------|---------|
| BOOL | 0 | 1 | 8 bits |
| WORD | -32,768 | 32,767 | 16 bits |
| DWORD | -2,147,483,648 | 2,147,483,647 | 32 bits |
| REAL | $\pm 3.4028110E-38$ | $\pm 3.4028110E+38$ | 32 bits |
| TIMER | 0 | 32,767 | 16 bits |
| COUNTER | 0 | 32,767 | 16 bits |
| DCOUNTER | -2,147,483,648 | 2,147,483,647 | 32 bits |

La variable más sencilla es la de tipo **BOOL** y únicamente almacena dos valores (True and False). Si se requiere trabajar con números enteros positivos al igual que con números enteros negativos se puede usar una variable tipo **WORD** de 16 bits y **DWORD** de 32 bits, esta última con más capacidad de almacenamiento. También, se puede expresar números con punto flotante de tipo **REAL** con un tamaño de 32 bits. Las variables **TIMER** y **COUNTER**, los cuales básicamente son variables ascendentes de 16 bits y por último la variable tipo **DCOUNTER** es un contador de tipo ascendente y descendente de 32 bits.

3.3.7 Declaración de variables

Para declarar una variable se tiene que considerar ciertos parámetros como el tipo de registro o relevador (E/S) que se asignara la variable. En la figura 35 se presenta los tipos de variables de ISPSOft.



Figura 35 Declaración de variables

Supongamos que declaramos las variables mostradas en la tabla 13

Tabla 13 Lo que no se debe hacer al declarar las variables

| Identificador | Tipo de variable/Dato | Dirección/Registro |
|---------------------|-----------------------|--------------------|
| Lectura a entero | Word | D0 |
| Conversión real | Real | D1 |
| Conversión a entero | Word | D2 |

Es muy importante entender lo siguiente al momento de declarar las variables. Los de tipo Word y Bool ocupan un solo registro (D0, D1, D2, D3, D4... Dn) de datos, mientras tanto, las variables de tipo Dword o Real ocupan dos registros de datos consecutivos (D0, D2, D4, D6, D8... Dn).

La declaración de la tabla 13 es incorrecta ya que la variable **Conversión real** está compartiendo el registro **D2** con la variable **Conversión a entero**. El programa no avisara si hay un error si los registros se traslapan, esto provocara que los valores sean incorrectos a la hora de hacer algún calculo o una conversión.

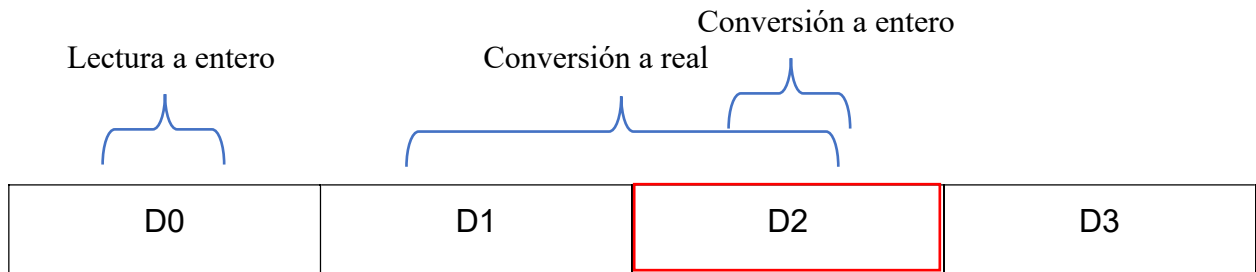


Figura 36 Configuración errónea

una forma correcta de declarar es como se muestra en la siguiente tabla 14

Tabla 14 Configuración correcta

| Identificador | Tipo de variable/Dato | Dirección/Registro |
|---------------------|-----------------------|--------------------|
| Lectura a entero | Word | D0 |
| Conversión real | Real | D1 |
| Conversión a entero | Word | D3 |

En ese sentido el identificador **Lectura a entero** ocupa el registro **D0** mientras la **Conversión a real** ocupa dos registros (**D1 y D2**) por lo tanto para la siguiente variable **Conversión a entero** ocupa solamente un registro (**D3**) por ser tipo Word, si fuese Dword ocuparía dos registros (D3 y D4) y esta es su forma correcta.

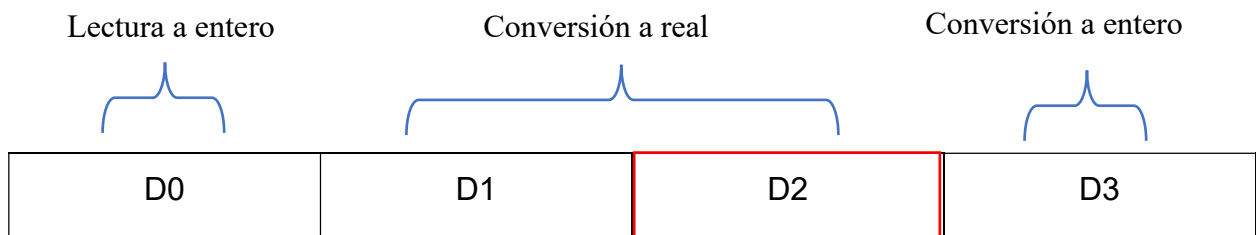


Figura 37 Configuración correcta

3.4 Diagrama de flujo del algoritmo

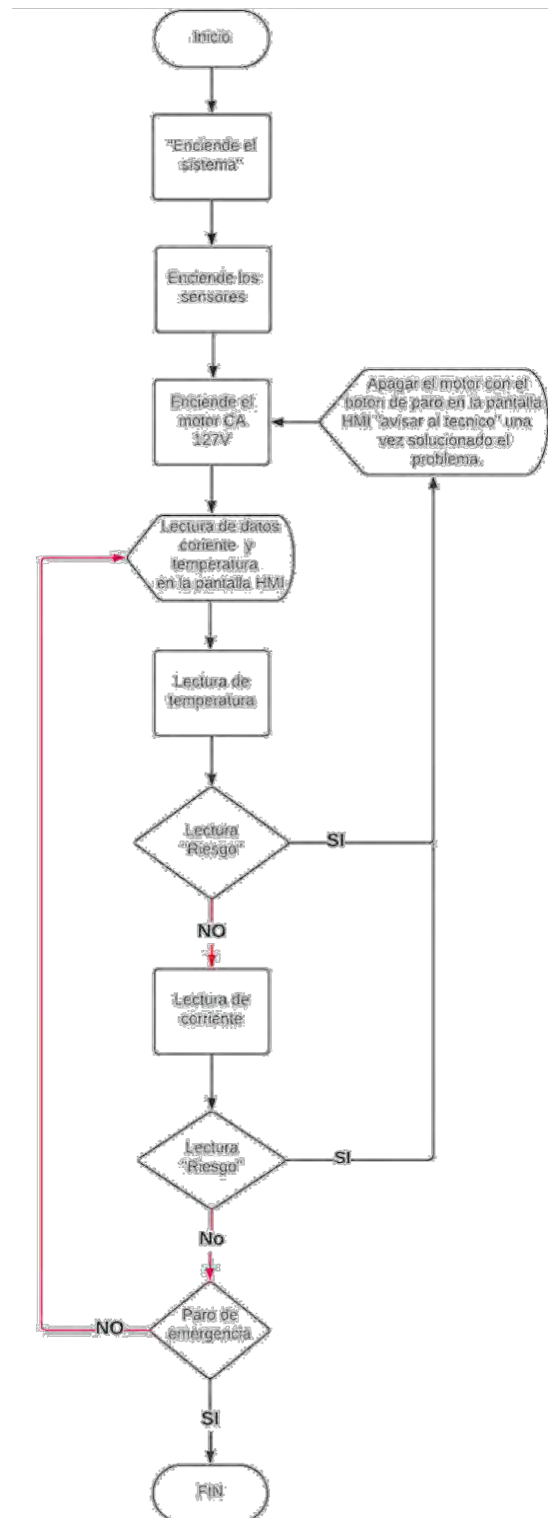


Figura 38 Diagrama de flujo del funcionamiento de la programación de control de un motor lectura en tiempo real de Corriente y Temperatura en ISPSOft.

3.5 Programa realizado

De acuerdo con las necesidades se decidió implementar un programa que cumpla con las siguientes características:

- Activar el sistema con un botón principal en la pantalla HMI.
- Monitoreo en tiempo real de los sensores de corriente y temperatura.
- Enviar alertas de colores según el estado de operación del motor, de tal forma que sea posible observar el comportamiento en tiempo real por medio de la HMI.
- Apagar el motor cuando envíe una alerta de riesgo ya sea de corriente o temperatura. Esta condición se realizó con la finalidad de proteger los equipos monitoreando con los sensores necesarios. En este caso se desea proteger un motor monofásico de 127 V.
- Un botón ON/OFF para el motor monofásico ubicado en la pantalla HMI. Esto ayudará, por ejemplo, cuando el técnico requiera hacer un mantenimiento y no tenga la necesidad de apagar todo el sistema.
- Apagar todo el sistema con un botón de paro de emergencia.

3.5.1 Programación del PLC

El diagrama de flujo anterior hace alusión a la lógica de programación del PLC DVP20SX2 que va en conjunto con la pantalla HMI DOP-B03S211, cuando los sensores (ver la figura 38) detecten una avería en el motor monofásico la función del PLC es detectar las fallas mediante la lógica de programación y esto será visto por medio de una alerta roja como símbolo de riesgo en la pantalla HMI. La pantalla HMI cuenta con la posibilidad de visualizar en tiempo real la lectura de corriente y temperatura del motor por lo tanto cuando el técnico requiera hacer servicio de mantenimiento al equipo se ocupa un botón solamente para apagar/encender el motor sin necesidad de parar todo el sistema.

Para implementar lo antes mencionado, fue necesario utilizar el software ISPSoft, el cual cuenta con las capacidades suficientes para poder llevar a cabo la tarea específica que se necesita, que es recopilar datos analógicos y conectar actuadores. En este caso el PLC DVP20SX2 es capaz de cumplir con esta tarea y los datos recopilados serán enviados por vía interfaz RS-485 para visualizar en tiempo real las variables en la pantalla HMI. Esta función es de vital importancia para el buen funcionamiento del sistema ya que los motores a veces se sobrecalientan o sobrepasan su corriente rebasando las condiciones estándar lo cual el objetivo es llevar un control.

En la tabla 15 se presenta las variables de la programación ocupando un total de 29 declaraciones.

Tabla 15 variables utilizadas en el proyecto

| Num. | Tipo Declaracion | Identificadores | Simbolos Locales | | | Valor Inicial |
|------|------------------|-------------------------|------------------|---------|--|---------------|
| | | | Direccion | Tipo... | | |
| 1 | VAR | LECTURA_AO_0 | D1 | REAL | | 0.00 |
| 2 | VAR | R1_AO_0 | D3 | REAL | | 0.00 |
| 3 | VAR | R2_AO_0 | D5 | REAL | | 0.00 |
| 4 | VAR | Conversion | D9 | REAL | | 0.00 |
| 5 | VAR | Conversion*C_2 | D11 | REAL | | 0.00 |
| 6 | VAR | LECTURA_AO_1 | D13 | REAL | | 0.0 |
| 7 | VAR | R1_AO_1 | D15 | REAL | | 0.0 |
| 8 | VAR | R2_AO_1 | D17 | REAL | | 0.0 |
| 9 | VAR | Conversion_A | D19 | REAL | | 0.0 |
| 10 | VAR | Conversion_A2_Corriente | D21 | REAL | | 0.0 |
| 11 | VAR | Diferencia_de_corriente | D23 | REAL | | 0.0 |
| 12 | VAR | Entero_°C | D25 | WORD | | 0 |
| 13 | VAR | Entero_A2_Corriente | D26 | WORD | | 0 |
| 14 | VAR | Entrada_AO_0 | D1110 | WORD | | 0 |
| 15 | VAR | Entrada_AO_1 | D1111 | WORD | | 0 |
| 16 | VAR | Start | M0 | BOOL | | N/A |
| 17 | VAR | Stop1 | M1 | BOOL | | N/A |
| 18 | VAR | Arranque | M2 | BOOL | | N/A |
| 19 | VAR | Optimo_Tem | M5 | BOOL | | FALSE |
| 20 | VAR | Preventivo_Tem | M4 | BOOL | | FALSE |
| 21 | VAR | Riesgo_Tem | M5 | BOOL | | FALSE |
| 22 | VAR | ON_Motor | M6 | BOOL | | FALSE |
| 23 | VAR | OFF_Motor | M7 | BOOL | | FALSE |
| 24 | VAR | Optimo_Corriente | M8 | BOOL | | FALSE |
| 25 | VAR | Preventivo_Corriente | M9 | BOOL | | FALSE |
| 26 | VAR | Riesgo_Corriente | M10 | BOOL | | FALSE |
| 27 | VAR | Temporizador_1 | T0 | TIMER | | N/A |
| 28 | VAR | Temporizador_2 | T1 | TIMER | | N/A |
| 29 | VAR | Motor | Y0 | BOOL | | FALSE |

3.5.2 Programación de la pantalla HMI

La red 1 es para la comunicación de la pantalla HMI con el PLC (ver la figura 39), los siguientes bloques de códigos permite la comunicación bidireccional RC y TX.

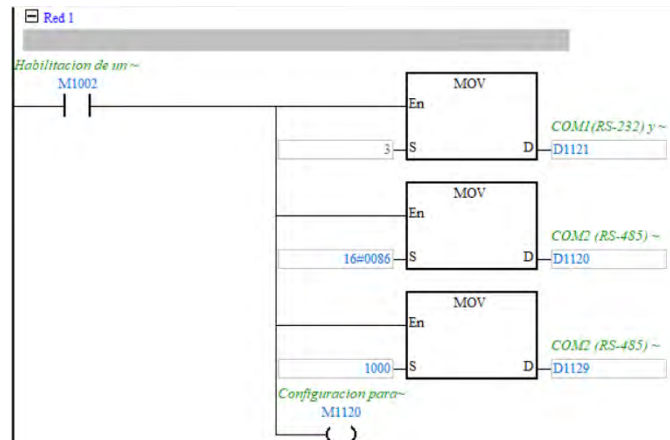


Figura 39 Configuración para comunicación RS-485

El registro especial M1002 su función es habilitar pulso positivo simple al momento en que se activa RUN del PLC (normalmente OFF).

El registro D1121 configura la dirección de comunicación esclavo del ID del PLC. Ocupa las direcciones de comunicación COM1 (RS-232) y COM2 (RS-485). Los números de ID van de 1 hasta el 16 en la figura 39 se observa que se ocupó el ID 3 para el PLC DVP20SX2.

Los PLC delta permiten usar cuatro puertos de comunicación. Para este caso se configura para el puerto RS-485 a 9600 que pertenece a la configuración de velocidades, longitud de datos, bits de paridad, entre otros, son necesarios para la comunicación serial RS-485 que se guardan en el registro especial H86 (16#0086, número hexadecimal). Esta configuración se mueve al registro D1120.

La configuración de tiempo de espera de comunicación se puede modificar por medio del registro D1129 con un rango disponible de $200 \leq D1129 \leq 3000$. El PLC tomará el valor limite superior/inferior como valor establecido.

Cuando el PLC cambia de STOP a RUN, el programa detectará si el relevador interno o marca M1120 (ver tabla 16) está en ON en la primera exploración. Si M1120 está en ON, el programa modificará la configuración de comunicación COM conforme al valor establecido en D1120 (DELTA, 2011, pág. 94).

Tabla 16 Formato de comunicación del PLC al HMI

| Puerto COM | RS-232 (COM1) | RS-485 (COM2) | RS-485 (COM3) | RS-485 (COM3) |
|---------------------------------------|----------------|---|---------------|-------------------------|
| Parámetro | | | | |
| Velocidad de transmisión | 110~115200 bps | 110~921000 bps | | 110~115200 bps |
| Longitud de datos | | 7~8 bits | | |
| Paridad | | Verificación de paridad Par / Impar / Ninguna | | |
| Longitud de bit de parada | | 1~2 bits | | |
| Registro para configuración | D1036 | D1120 | | D1109 |
| Retener formato de configuración | M1138 | M1120 | | M1136 |
| Modo ASCII | | Disponible para ambos maestro/esclavo | | Disponible para esclavo |
| Modo RTU | | Disponible para ambos maestro/esclavo | | Disponible para esclavo |
| Selección de modo ASCII/RTU | M1139 | M1143 | | M1320 |
| Dirección de comunicación de Esclavo | | D1121 | | D1255 |
| Longitud de datos para acceso (ASCII) | | 100 registros | | |
| Longitud de datos para acceso (RTU) | | 100 registros | | |

3.5.3 Programación LD del sensor de temperatura LM-35

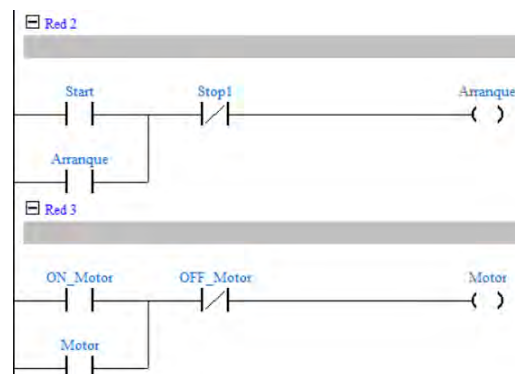


Figura 40 Arranque/Paro del todo el sistema y ON/OFF del motor

En la red 2 se cuenta con 3 variables Star, Stop y Arranque siendo registros de tipo relé auxiliar (M) (ver la tabla 15). Su función es encender y apagar todo el sistema (motores y sensores).

En la red 3 también contamos con tres variables ON_Motor y OFF_Motor. Estos registros son de tipo relé auxiliar (M) y la variable Motor es de dirección relé salida (Y) (ver la tabla 15). Su función es encender y apagar el motor de forma independiente.

En la Red 4 y 5 se utiliza un bloque TMR (temporizador) para estimar la corriente por pulsos de 10 ms (ver la figura 41). la variable temporizador_1 se ocupará más adelante en la

red de programación debido a que la lectura del sensor LM-35 y el sensor LA 25-NP la velocidad de muestreo en la pantalla HMI son demasiado rápidos.

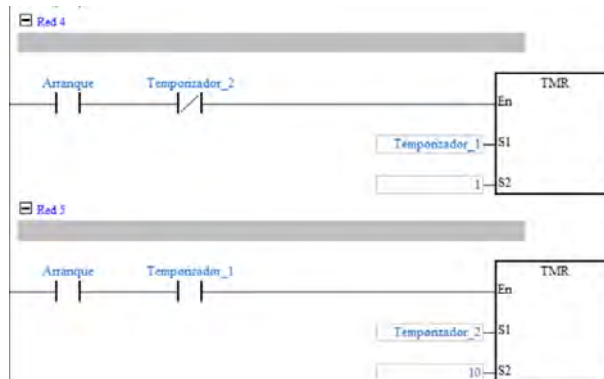


Figura 41 Uso del bloque TMR para controlar el sensor LA 25-NP

La red 6 ocupa el bloque de control MOV lo cual se utiliza la misma configuración de la tabla 8 donde el numero decimal 0 se mueve al registro D1114 por ello los canales analógicos del PLC DVP20SX2 están habilitados, en la red 7 el registro D1115 el sensor se configuro a modo voltaje como en la tabla 10 y finalmente en la red 8 para que entrada física aplica esta configuración en este caso la entrada analógica 0 con el registro D1110 (ver la tabla 15) nombrado Entrada:OA-0 y ese registro se mueve a la de Lectura_OA_0 (ver la figura 42).

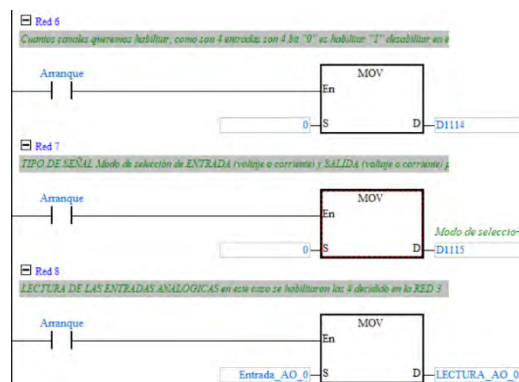


Figura 42 Bloque de control MOV para el registro D1114, D1115 y la entrada física del puerto analógico D1110

El rango de lectura de las entradas analógicas es de 0 a 2000 por lo tanto el sensor LM-35 tiene un rango de -550 mV a 1500 mV lo que se traduce de la siguiente manera, -550 mV equivale a 0 siendo el valor mínimo y 1500 mV equivale a 2000 siendo el valor máximo de lectura. Por lo tanto, en la red 8 se ocupa un bloque de control FLT para punto decimal. En la

red 10 y 11 se realiza operaciones matemáticas para obtener la lectura en modo voltaje (ver la figura 43).

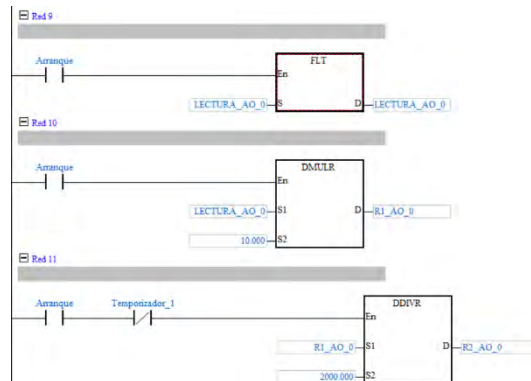


Figura 43 bloque de control y bloques operacionales para la configuración a modo voltaje (sensor LM-35)

En la red 13 de la figura 44 se realiza una conversión de voltaje a grados °C. Se ocupa el bloque DMOV cuando el motor se apague esta reinicie a cero el muestreo del sensor LM-35 en la pantalla HMI.

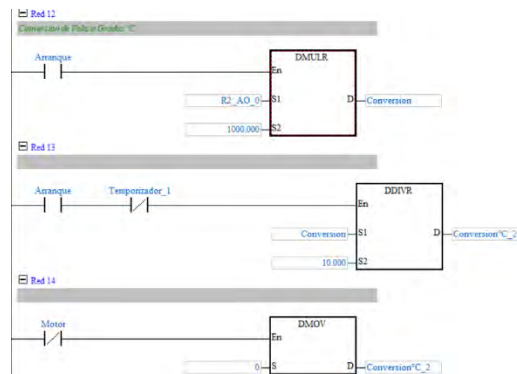


Figura 44 conversión de voltaje a grados °C

Se ocupa el bloque INT con el número API 129 (ver el anexo 2) para convertir los grados °C de punto flotante a entero. En la red 16 se ocupa el bloque MOV cuando estas dos variables Arranque o Motor se apague reinicie a cero el muestreo del sensor LM-35 en la pantalla HMI (ver la figura 45).

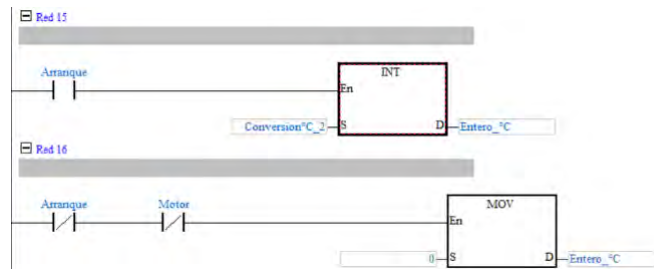


Figura 45 Conversión de punto flotante a número entero los grados °C

En la red 17 se ocupa una bobina de relé auxiliar su función es enviar una alerta por medio de un bloque de control en ese sentido si la variable Entero_°C va de $0 < 40$ °C esta activara la bobina Optimo_Tem. En la red 18 aplica cuando la temperatura este 40 °C < 65 °C se apaga la bobina Optimo:Tem y se activa la bobina Preventivo:Tem. En la red 19 cuando rebase los 65 °C se desactiva la bobina Optimo:Tem y se activara la bobina Riesgo:Tem admitiendo un rango de $65 \leq 100$ °C (ver la figura 46)

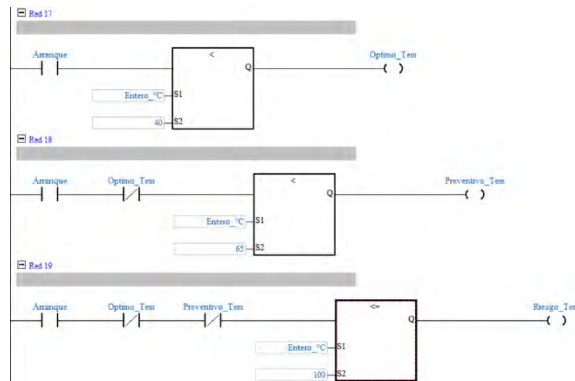


Figura 46 Parámetros temperatura del sensor LM-35

3.5.4 Programación LD del sensor de corriente LA 25-NP

La red 20 ocupa el bloque de control MOV se utiliza la misma configuración de la tabla 8 donde el numero decimal 0 se mueve al registro D1114 lo cual significa que los canales analógicos del PLC DVP20SX2 están habilitados. En la red 21 el registro D1115 el sensor se configuro a modo voltaje como en la tabla 10 y finalmente en la red 22 para que entrada física aplica esta configuración en este caso la entrada analógica 1 con el registro D1111 (ver la tabla 15) nombrado Entrada_AO_1 y ese registro se mueve a la Lectura_AO_1 (ver la figura 47).

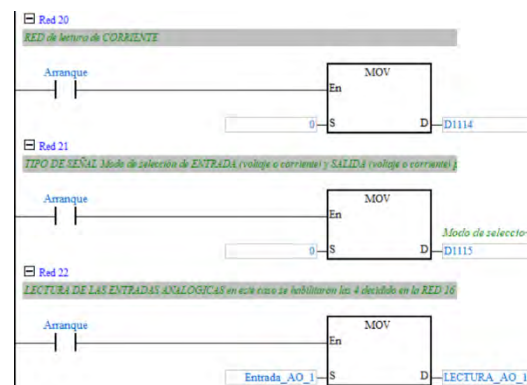


Figura 47 Bloque de control MOV para el registro D1114, D1115 y la entrada física del puerto analógico D1111

El rango de lectura de las entradas analógicas es de 0 a 2000 por lo tanto el sensor LA 25-NP tiene un rango de 0 amperes a 25 amperes lo que se traduce de la siguiente manera, 0 amperes equivalen a 0 siendo el valor mínimo y 25 amperes equivale a 2000 siendo el valor máximo de lectura. Por lo tanto, en la red 23 se ocupa un bloque de control FLT para punto decimal. En la red 24 y 25 se realiza operaciones matemáticas para obtener la lectura en modo voltaje (ver la figura 48).

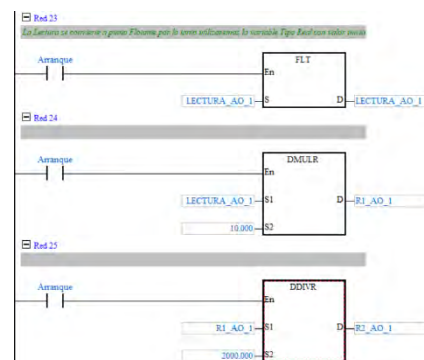


Figura 48 Bloque de control y bloques operacionales para la configuración a modo voltaje (sensor LA 25-NP)

El siguiente paso es convertir el resultado a modo corriente realizando una diferencia de corriente (ver la figura 49).

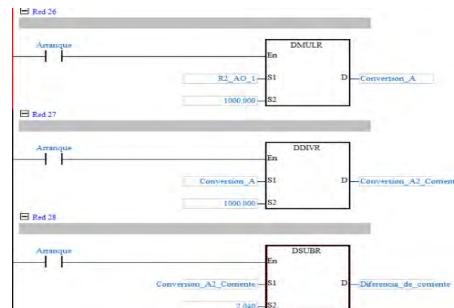


Figura 49 Conversión de voltaje a corriente

En la red 29 y 31 se agrega un bloque DMOV y INT su función es reiniciar a cero la lectura del sensor cuando este apagado.

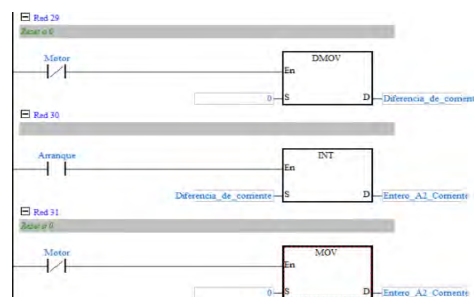


Figura 50 Reset a 0 la lectura del sensor LA 25-NP

En la red 32 se ocupa una bobina de relé auxiliar su función es enviar una alerta por medio de un bloque de control en ese sentido si la variable Entero_A2_Corriente va de **0A <7A** esta activara la bobina Optimo_Corriente. en la red 33 aplica cuando la corriente este **7A < 10A** se apagara la bobina Optimo_Corriente y se activara la bobina Preventivo_Corriente. en la red 19 cuando rebasa los 10A se desactivará la bobina Optimo_Corriente y se activará la bobina Riesgo_Corriente admitiendo un rango de **10A < 12A** (ver la figura 51).

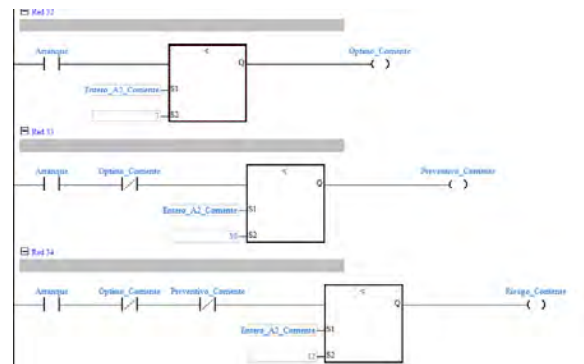


Figura 51 Parámetros corriente del sensor LA 25-NP

CAPÍTULO 4 PROGRAMACIÓN DE LA PANTALLA

La pantalla HMI de Delta proporciona varios puertos para una comunicación rápida y sencilla en el control y monitoreo de varios dispositivos simultáneamente, tanto para un monitoreo local o para realizarlo remotamente en otro lugar de la planta. Aplica para una amplia gama de actuadores, sistemas e instalaciones. Pantalla táctil a color permite la entrada de parámetros de manera intuitiva y permite mostrar una gran variedad de formas graficas los datos, variables, elementos de alarma e incluso gráficos de tendencias. El software DOPSoft 2.0 sirve para programar las HMI de la marca Delta, series DOP-B, DOP-H y DOP-W (CIBERCHA, 2022).

4.1 Instalación del software DOPSoft 2.0

Para instalar el software DOPSoft se necesita las siguientes características técnicas mínimas:

- Sistema operativo Windows XP/ Vista / 7 / 8.1 / 10. Sean de 32 bits / 64 Bits.

Seguidamente se presenta los pasos para la instalación:

Paso 1: Se busca en Google “Download DOPSoft 2.00.07”, seleccionar el primer enlace el cual te enviara a la página oficial de Delta como se muestra en la figura 52, DOPSoft cuenta con varias versiones y es necesario elegir el software de programación adecuado al modelo de la pantalla. En esta ocasión se cuenta con una pantalla HMI de la serie **DOP-B03S211** y la versión del software le corresponde al 2.00.07.



Figura 52 Download DOPSoft 2.00.07

Paso 2: Ubicar en la carpeta de descargas el archivo “DOPSoft 2.00.07.04 Build DEC_14_2017” clic derecho y ejecutamos como administrador (ver la figura 53).



Figura 53 Ejecutar como administrador el programa DOPSoft

Paso 3: En ese instante aparecerá un recuadro donde se selecciona el idioma del instalador (ver la figura 54).

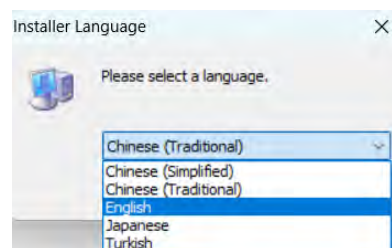


Figura 54 Selección del idioma del instalador.

Paso 4: Se abrirá una ventana donde informa que se instalará el programa principal y el complemento Microsoft SQL Server, en caso de que la computadora no lo tenga instalado se presiona instalar como se muestra en la figura 55.

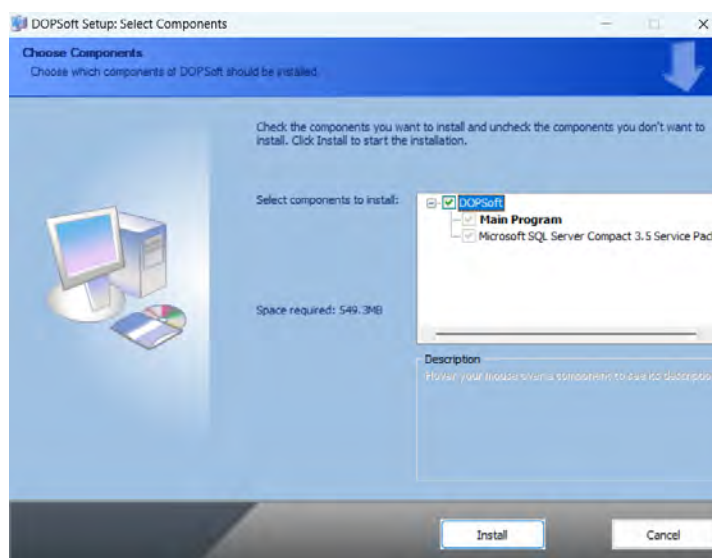


Figura 55 Instalación del programa DOPSoft y SQL

Paso 5: En la instalación del programa se abrirá un recuadro cuestionando si deseamos instalar los drivers para comunicar la computadora con el PLC. Selecciona la casilla “Siempre confiar en el software de Delta Electronics” e instalar (ver la figura 56).

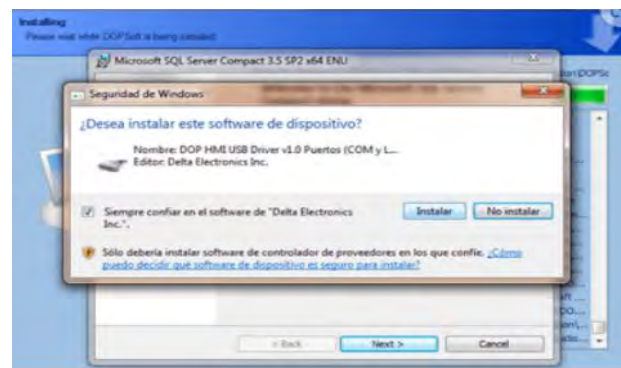


Figura 56 Instalación de los drivers para la comunicación del PLC Delta

Paso 6: Seguidamente aparecerá una ventana para instalar el software Microsoft SQL Server. Presiona el botón “Netx”. Ahora se desplegará el acuerdo de licencia aceptamos y más a delante selecciona “Next” (ver figura 57).

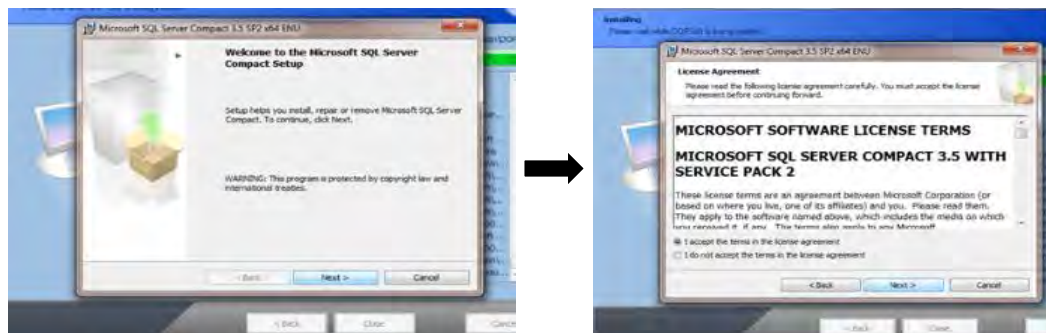


Figura 57 Aceptación de licencia del software Microsoft SQL Server.

Paso 7: Al finalizar la instalación presiona el botón “Finish” y por último “Close” (ver la figura 58).

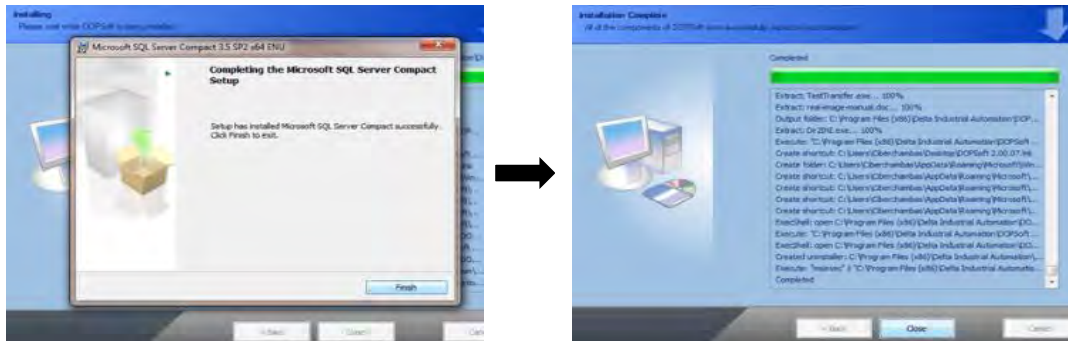


Figura 58 Finalización de la instalación del DOPSoft

Paso 8: Ya tendrá el software instalado y listo para usar en su ordenador.

4.2 Iniciar el software DOPSoft

Localizar el icono de DOPSoft, (ver la figura 59) accede al programa.

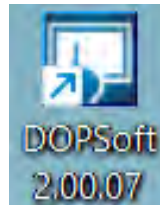


Figura 59 Icono de DOPSoft

Una vez abierto el programa, se crea un nuevo archivo dirigiéndose al menú “File” (ver la figura 60) y selecciona “New”. Se abrirá el siguiente asistente de configuración donde se tiene que elegir la serie de pantalla que se desea programar, para este caso se utiliza el modelo DOP-B03S21, ver la figura 60 en el recuadro remarcado de color rojo y a un costado aparecerá la lista de las pantallas HMI. Se ubica el modelo y se selecciona. En el apartado de “Project Setup” el usuario tiene la opción de elegir a libertad el nombre del proyecto, el número de pantallas y el tipo de lenguaje entre otros.

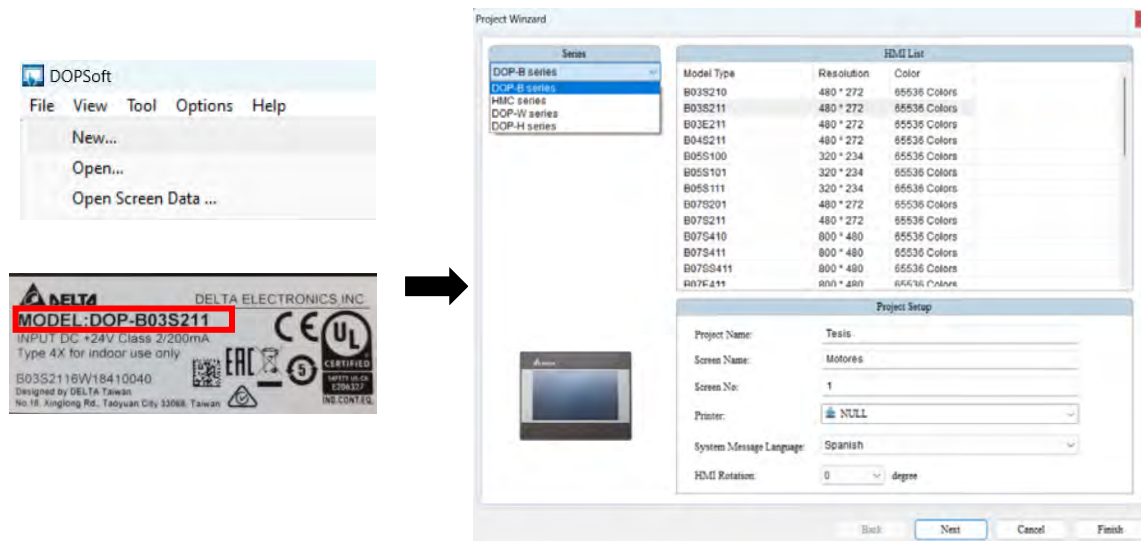


Figura 60 selección de la serie y el modelo de la pantalla HMI en DOPSoft

En la siguiente ventana se configura el puerto de comunicación que posteriormente se vincula con el PLC. Es muy importante recordar el registro D1121 que guarda el ID del PLC

maestro y el tiempo de espera para la comunicación que se eligió con el registro D1129, por lo tanto, debe de coincidir, como se muestra en el recuadro color rojo y azul (ver la figura 61).

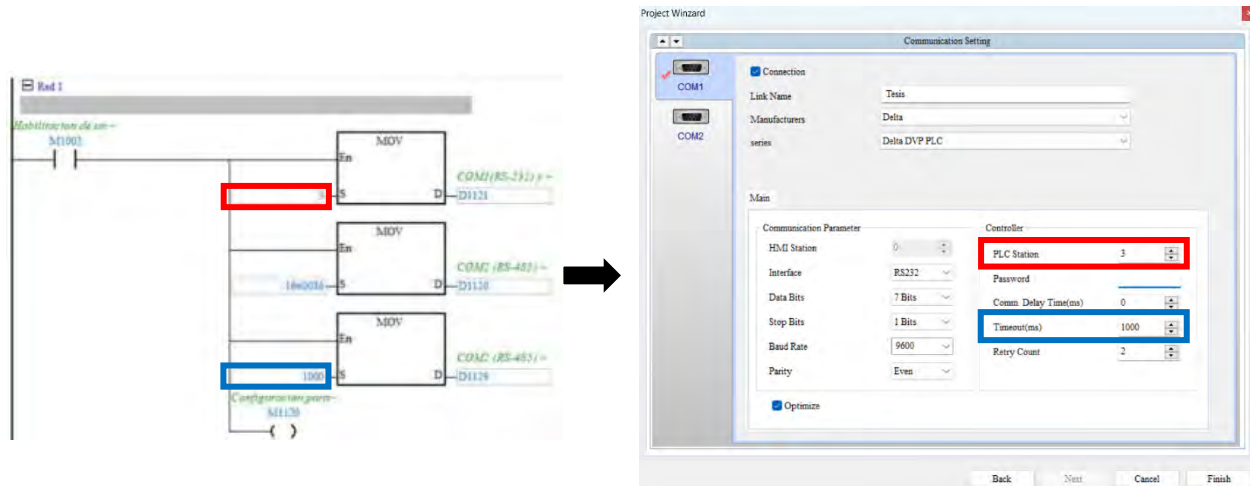


Figura 61 ID del PLC con el cual se requiere comunicar

Una vez realizadas las configuraciones correspondientes, se abrirá el área de trabajo.

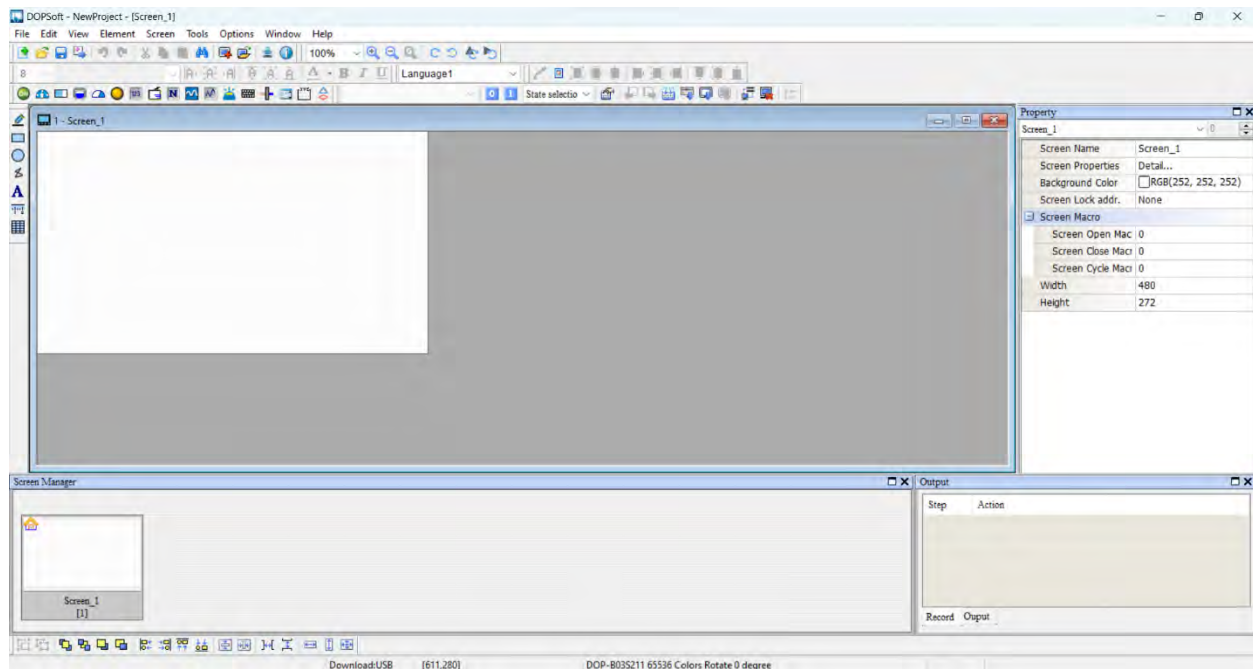


Figura 62 Programa DOPSoft para el modelo DOP-B03S211

4.3 Programación de la pantalla

El software DOPSoft sirve para realizar una interfaz gráfica con la respectiva programación de elementos virtuales como son botones, imágenes, histogramas, alarmas, indicadores, etc. (Santacruz Ron, 2017). En la figura 63 se observa las herramientas de programación de DOPSoft.

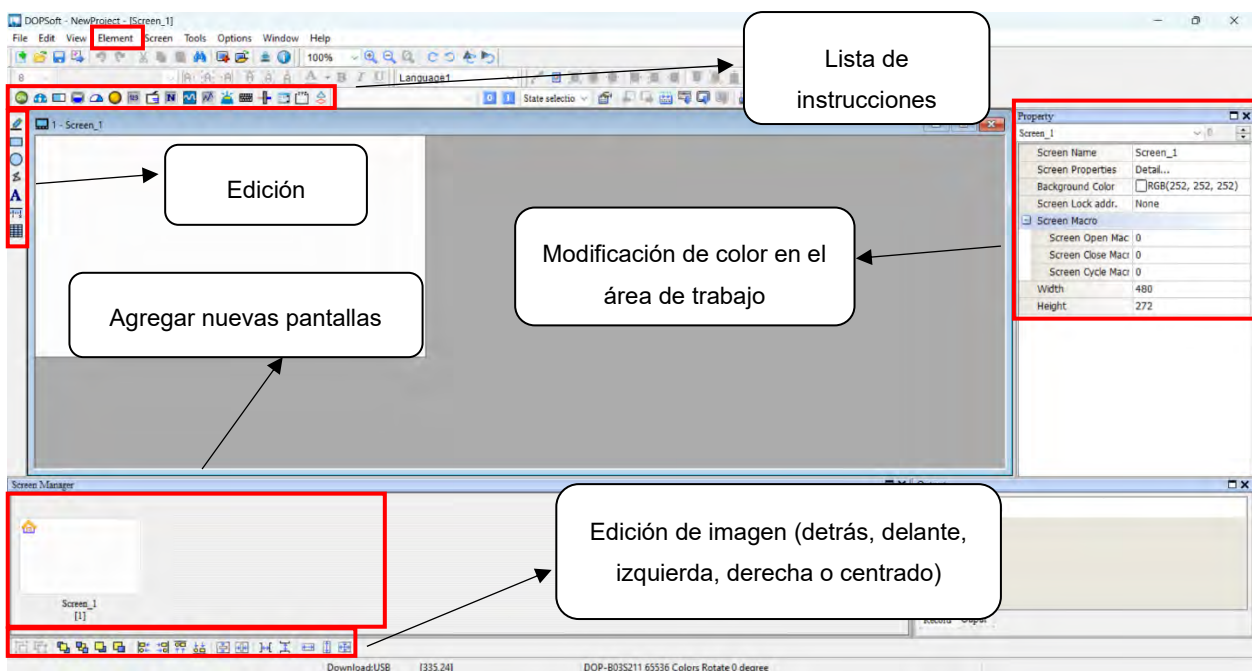


Figura 63 Elementos de programación de DOPSoft

4.3.1 Explicación de los bloques

DOPSoft tiene la facilidad de enlazar direcciones internas del PLC con los elementos virtuales, su fácil programación permite al usuario desarrollar una interfaz gráfica de monitoreo y de control.

A continuación, se muestra un ejemplo básico, la salida **Y0** se activará a través del contacto **X0**. La salida **Y1** se activará a través de la marca **M0**. Ubicamos en la barra de tareas el símbolo de botón ON parte superior izquierdo, seguidamente “Momentary” y en screen se mostrará un preview de color gris. Al hacer doble clic se abrirá un recuadro donde se configura como se muestra en la figura 64.

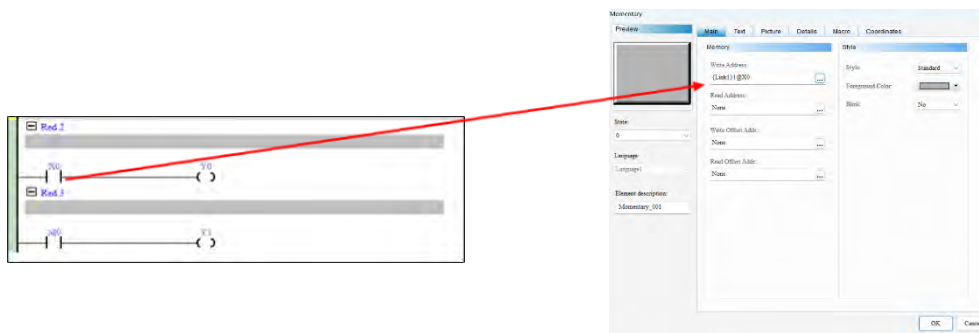


Figura 64 Configuración de botón Momentary en DOPSoft y ISPSOft

La entrada **X0** tiene la función de accionar como un ON/OF mientras la salida **Y0** se configura como un Multistate Indicador. La salida se activará de forma visual cuando se accione la entrada **X0** y se verá reflejado en la salida **Y0** que podría ser un foco, motor, una alarma, etc. En la figura 65 se muestra la configuración.

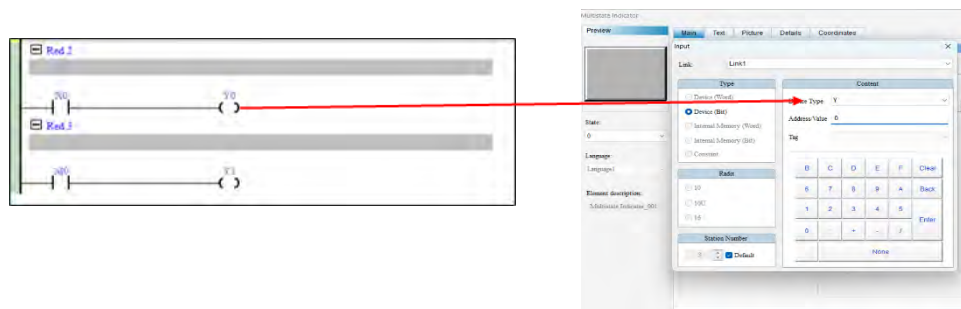


Figura 65 Configuración de una salida por medio de un Indicador Multistate en DOPSoft y ISPSOft

DOPSoft proporciona una gran variedad de iconos como: botones, focos, sensores, símbolos entre otros (ver la figura 66).

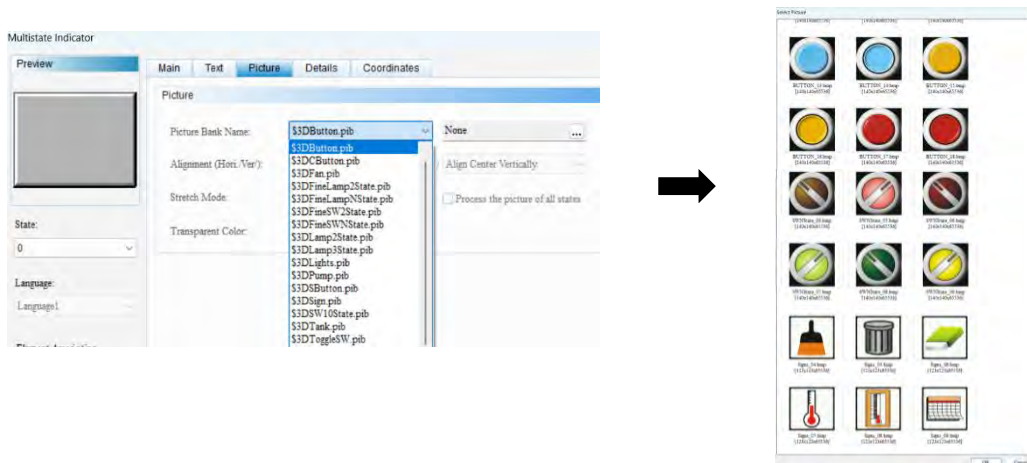


Figura 66 carpeta de imágenes y símbolos

Para añadir un botón y un indicador de activado y desactivado, en “Stretch Mode” selecciona “Stretch All” para que el botón se vea completo. En “Transparent Color” quita el fondo negro del botón, en la parte “State” hay dos estados 0 y 1 donde 0 es la condición estándar sin oprimir (OFF) y 1 es cuando se presiona el botón (ON) (ver la figura 67).

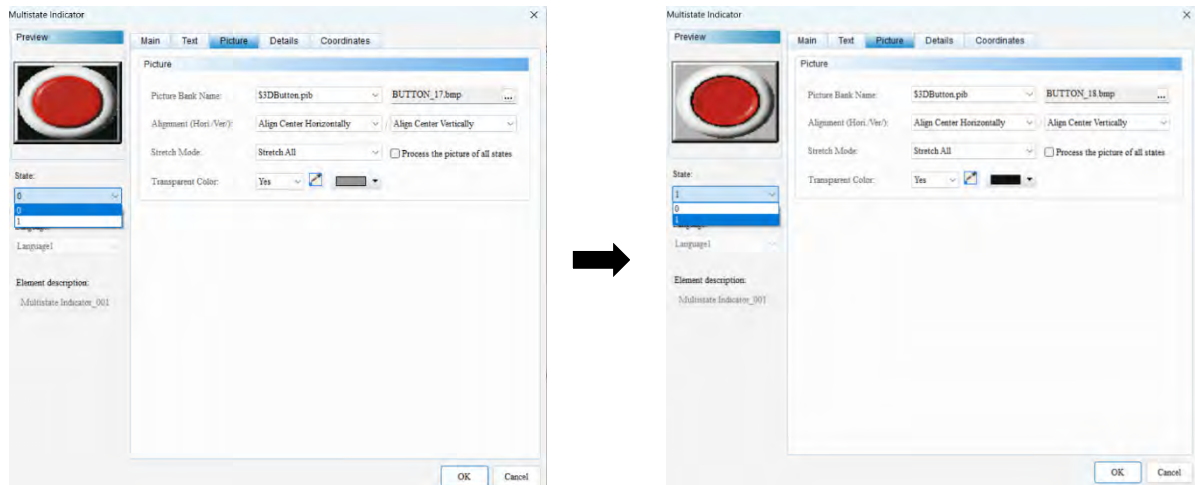


Figura 67 Procedimiento de agregar una imagen de botón

Finalmente, la interfaz gráfica se observa de la siguiente manera cuando se oprima el botón **X0** encenderá la salida **Y0** lo mismo ocurrirá con la marca **M0** y la salida **Y1** (ver la figura 68).

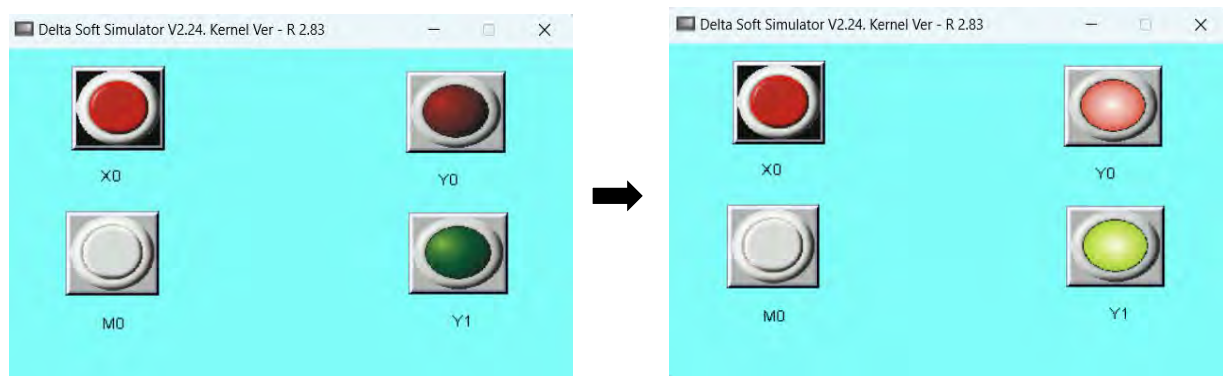


Figura 68 Ejemplo de interfaz grafica

4.3.2 Diseño de la pantalla

En este apartado se explicará el diseño de los screen utilizados en el proyecto.

4.3.2.1 Explicación de los bloques

En este diseño se ocupa un “Animasted Graphic” su función es crear una animación con imágenes. DOPSoft tiene la facilidad de cargar archivos de tipo GIF descargadas de internet. Para agregar un GIF a la pantalla HMI es ubicar la barra principal “Options” y después “Picture Bank” (ver la figura 69).

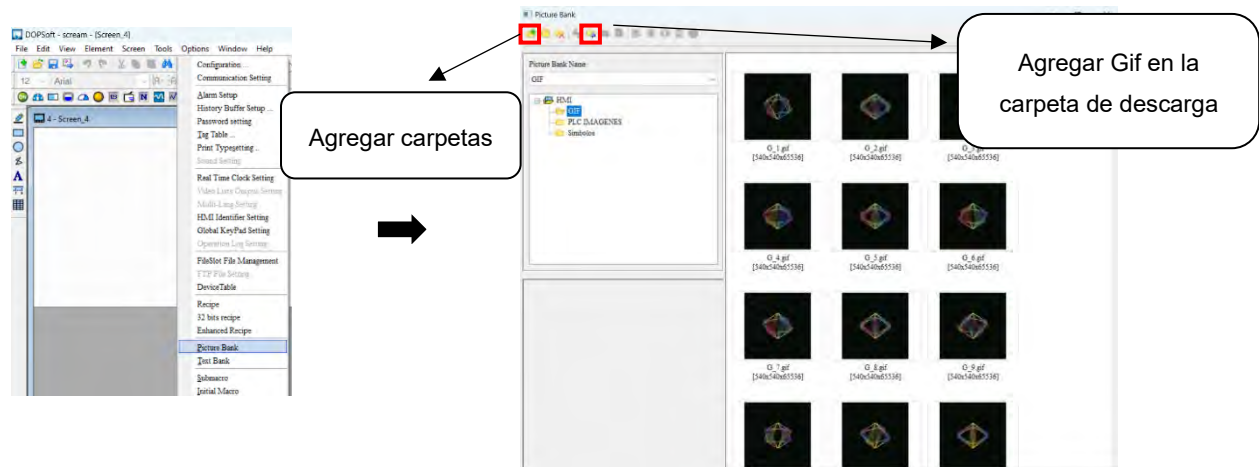


Figura 69 Como agregar una imagen o GIF de la carpeta de descargas a DOPSoft

En esta ocasión el GIF generó 90 imágenes, en la barra de herramientas se ubica “Movement” se selecciona “Animasted Graphic” y en el screen se inserta y se mostrará un preview de color blanco con un muñeco verde. Al hacer doble clic se abrirá un recuadro (ver la figura 70). En “State Counts” se agrega el número de imágenes que se desea agregar para hacer la animación considerando que se requiere 90 fotogramas en “State”, se enlista las 90 casillas para ir agregando imagen por imagen.

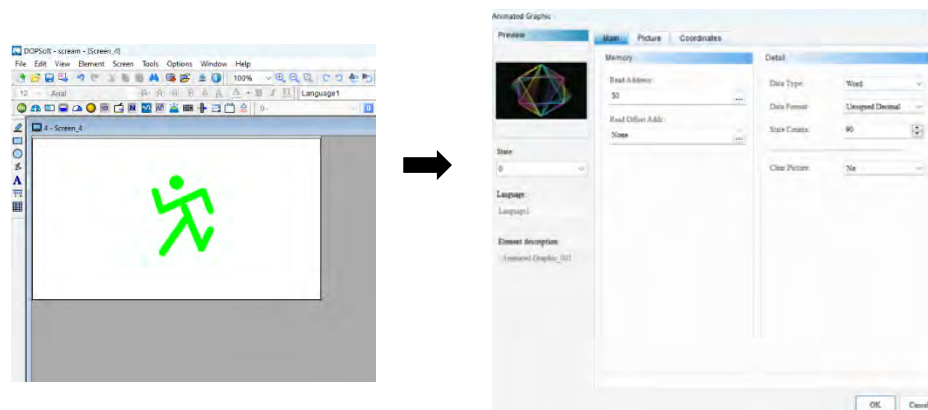
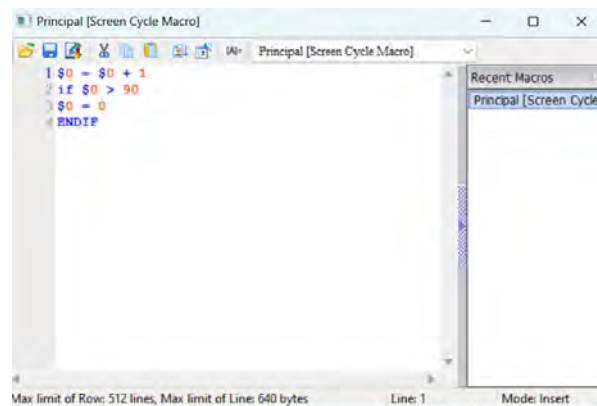


Figura 70 Configuración de Animated Graphic

En “Read Address” se asigna la variable \$0 para realizar una macro. Para esto se elige en la pestaña de “Screen” en DOPSoft y se selecciona “Screen Cycle Macro” (ver la figura 70). En la figura 71 se muestra la programación de esta variable independiente \$0. En el código en la fila uno se programa un contador ascendente, que irá incrementado de uno en uno. En la fila dos se agrega una condición IF, donde si la variable es mayor de 90 imágenes esta se reinicia a cero y el código termina con ENDIF. Esto se repite de manera infinita al momento de poner en marcha la pantalla HMI.



```

1 $0 = $0 + 1
2 IF $0 > 90
3 $0 = 0
ENDIF

```

Figura 71 Código para crear una animación en DOPSoft

Para mejorar la presentación de la pantalla HMI se le puede agregar bloques de fecha “Date display”, que están ubicados en la barra de elementos en la opción “Display”. Al igual se puede añadir la hora con “Timer display”, estos dos bloques se configuran de manera automática. Finalmente se presenta la pantalla principal en la figura 72.

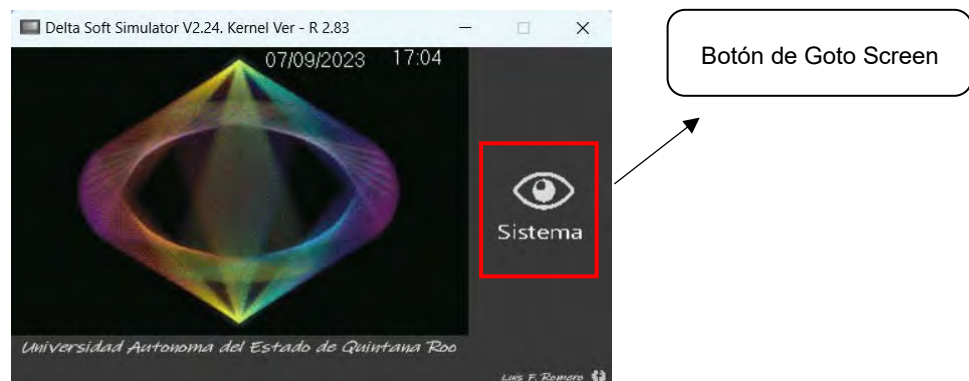


Figura 72 Pantalla Principal

4.3.2.2 Agregar una pantalla secundaria

A la pantalla principal se le puede agregar un botón virtual llamada “Goto Screen”, que su función es enviarte a una pantalla secundaria señalada previamente en la figura 72.

Para agregar la nueva pantalla y la configuración del botón Goto screen, es ubicar en la parte inferior la barra de “Screen manager” (ver la figura 63), se hace clic derecho y se selecciona “New screen” donde aparecerá una nueva ventana (ver la figura 73). En “Screen name” se le agrega un nombre para identificarlo, “Screen ID” es el número de pantalla en este caso es la pantalla dos y “Screen Type” se selecciona screen y se genera una pantalla secundaria llamada **Escritorio**.

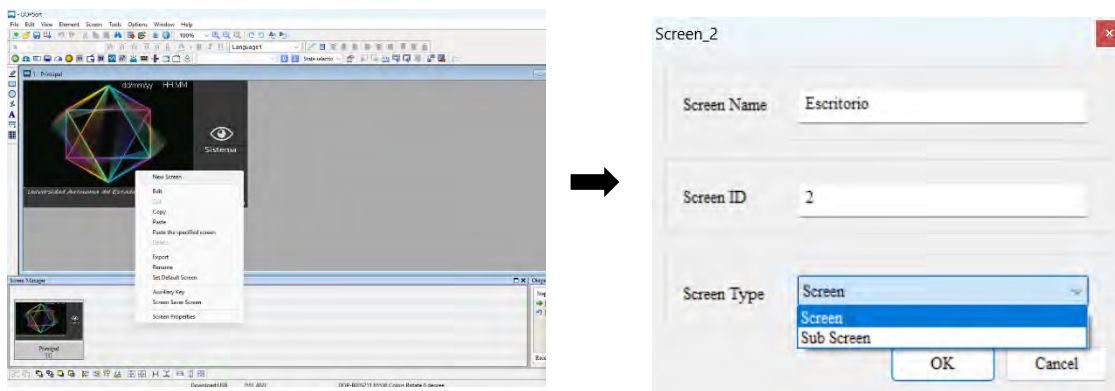


Figura 73 Agregar una pantalla secundaria en ISPSOft.

En la pantalla principal hay un botón nombrado **SISTEMA** (ver la figura 72) se le da doble clic y se abrirá un recuadro (ver la figura 74).

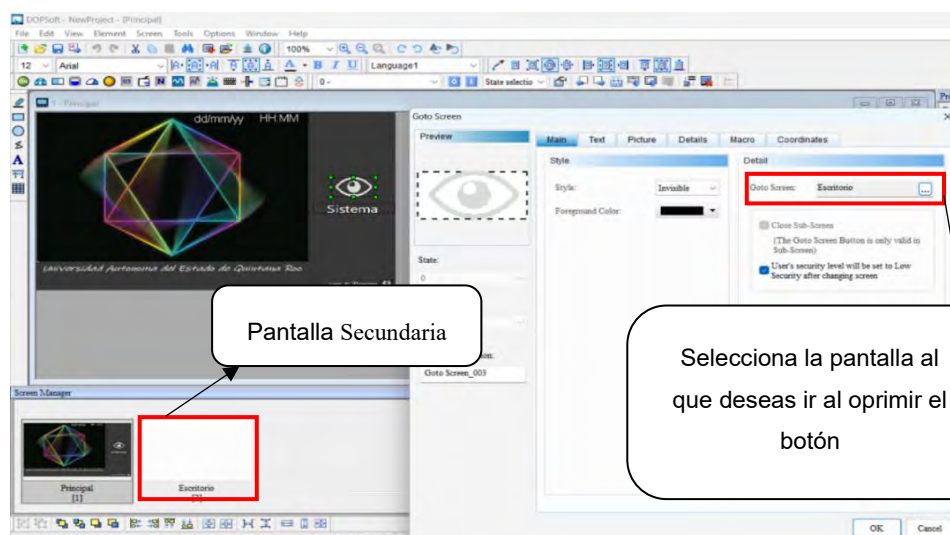


Figura 74 Explicación del botón Goto Screen

Ahora al momento de oprimir el botón de la pantalla principal esta te enviara a una pantalla secundaria denominado **ESCRITORIO**, donde se puede realizar otra interfaz.

4.3.2.3 Diseño de la pantalla secundaria

En este apartado se personaliza la segunda pantalla que tendrá una interfaz de escritorio para una mejor visualización, para ello se agrega un “Animasted Graphic” realizando el mismo procedimiento del punto **4.3.2.1**, seguidamente se agrega 4 botones que tendrán distintas direcciones;

1. **Escritorio (Botón Goto Screen)** esta tendrá la función de regresar a la misma pantalla de escritorio (ver la figura 75).

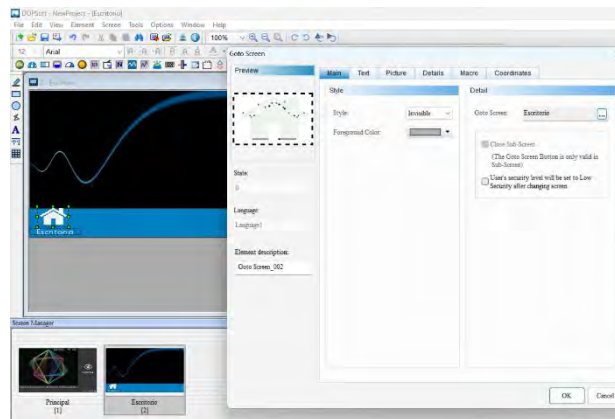


Figura 75 Configuración de botón Escritorio Goto Screen.

2. **General (Botón Goto Screen)** tiene la función de abrir una interfaz donde se visualiza la lectura en tiempo real de los sensores analógicos: corriente y temperatura con sus respectivos colores según el estado en el que se encuentre operando (ver la imagen 76).

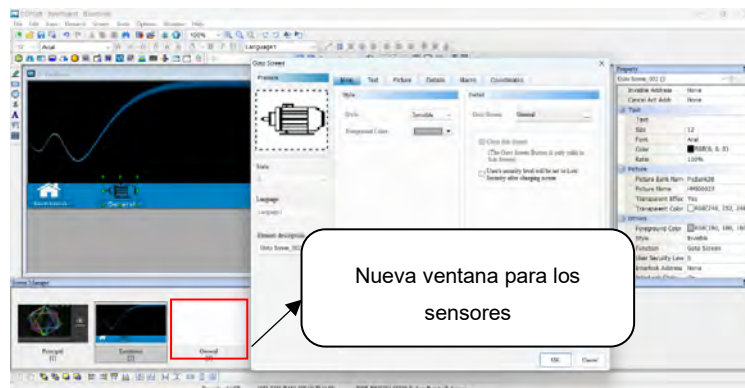


Figura 76 Botón de interfaz de los sensores

Esta interfaz ayuda en la supervisión de los sensores de corriente y temperatura. En DOPSoft hay un elemento llamado “Meter”, su función es leer datos de tipo Word y Dword. En el programa realizado en ISPSOft se ubica la variable donde se ejecuta la conversión de punto flotante a decimal, ubicado en la red 15. Los valores que envía el sensor al PLC se insertan en la variable D25 en el “Read Address” del elemento virtual Meter que corresponde a la lectura del sensor de temperatura (ver la imagen 77).

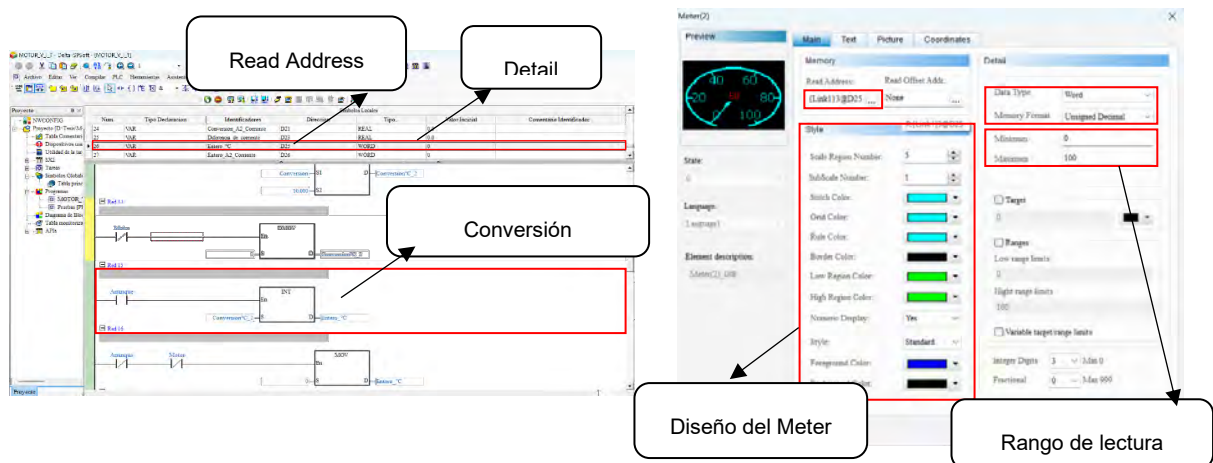


Figura 77 Aplicación del bloque Meter para el sensor de temperatura

Se ejecuta el mismo procedimiento para el sensor de corriente (ver la imagen 78)

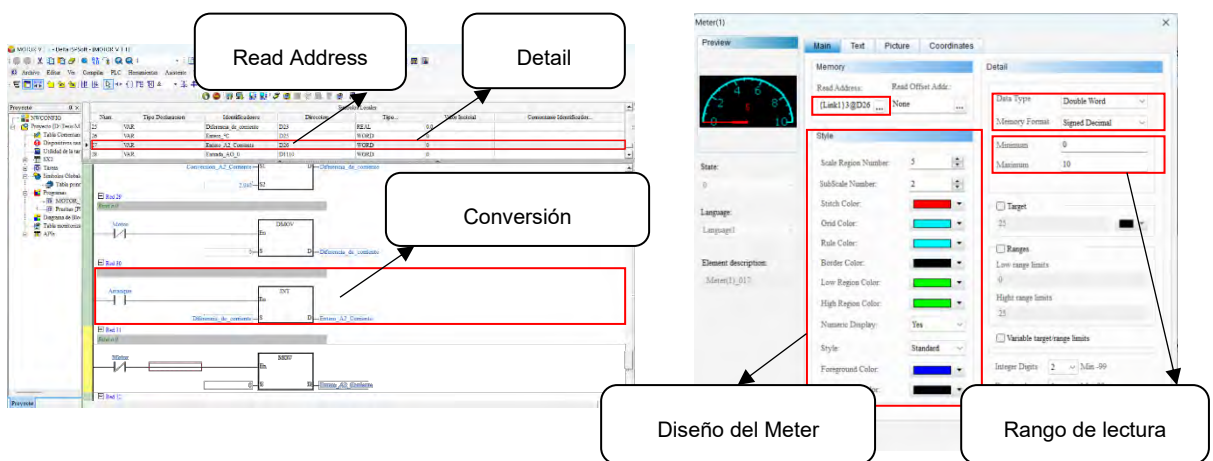


Figura 78 Aplicación del bloque Meter para el sensor de corriente

El único inconveniente del bloque Meter es que admite solamente valores enteros. Para complementar con más exactitud la lectura, se agrega un elemento “Numeric display” que cuenta con lectura de números enteros (Word y Dword) y punto flotante (Real).

En el programa realizado en ISPSoft ubicar la variable donde recibe las señales de los sensores en punto flotante (ver la imagen 79).

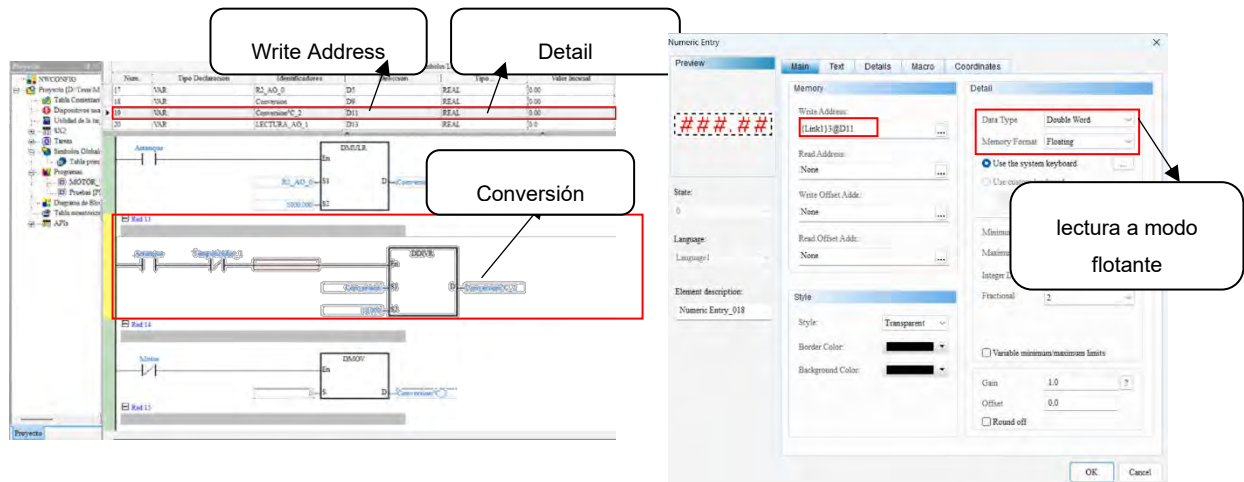


Figura 79 Bloque Numeric Display para números de punto flotante sensor de temperatura

El siguiente paso es agregar tres indicadores (**Optimo, Preventivo y Riesgo**) esta llevara un control por medio de leds de colores según el estado de funcionamiento, donde optimo pertenece al color verde, preventivo al color amarillo y riesgo al color rojo. Estas se configuran con el bloque de “Multistate indicador” (ver la figura 80) recuerda que este bloque tiene dos estados 0 y 1 donde 0 es apagado y 1 es encendido. Aquí se agrega la figura que se desee (ver el procedimiento de la figura 66 y 67) y aplica para los dos indicadores restantes.

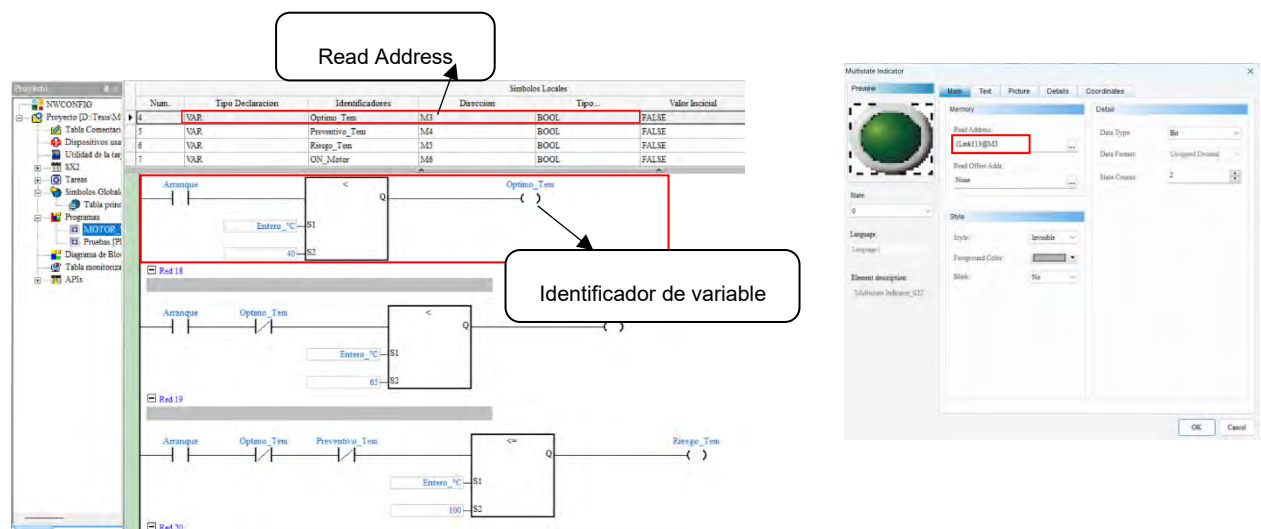


Figura 80 Procedimiento para configurar un indicador

Finalmente, cuando uno de los sensores indique una posible avería se tendrá un botón de Goto screen el cual abrirá una ventana donde solamente se visualice la medición, los indicadores y dos botones ON/OFF, lo que significa que es exclusivamente para apagar el actuador donde se encuentre el sensor para posibles mantenimientos (ver la figura 81).

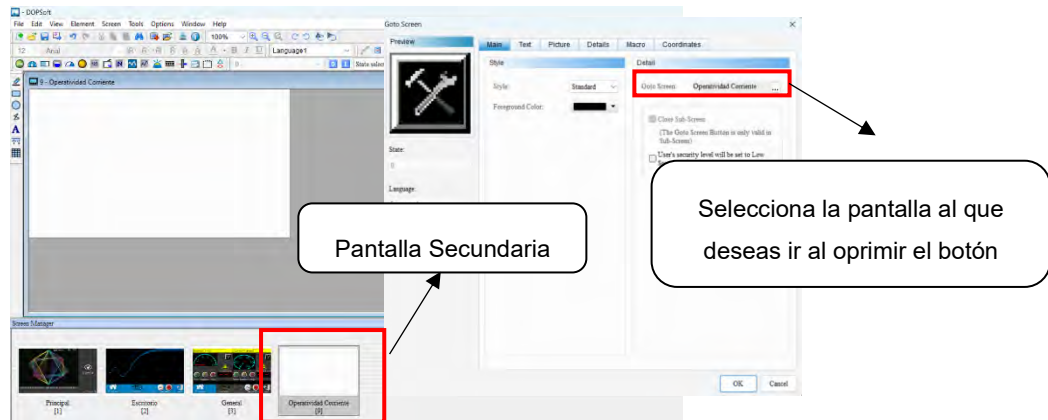


Figura 81 Nueva pantalla Para mantenimiento

El resultado de este punto es lograr un diseño sencillo pero entendible para el usuario, entonces la forma de interfaz se presenta en la figura 82.

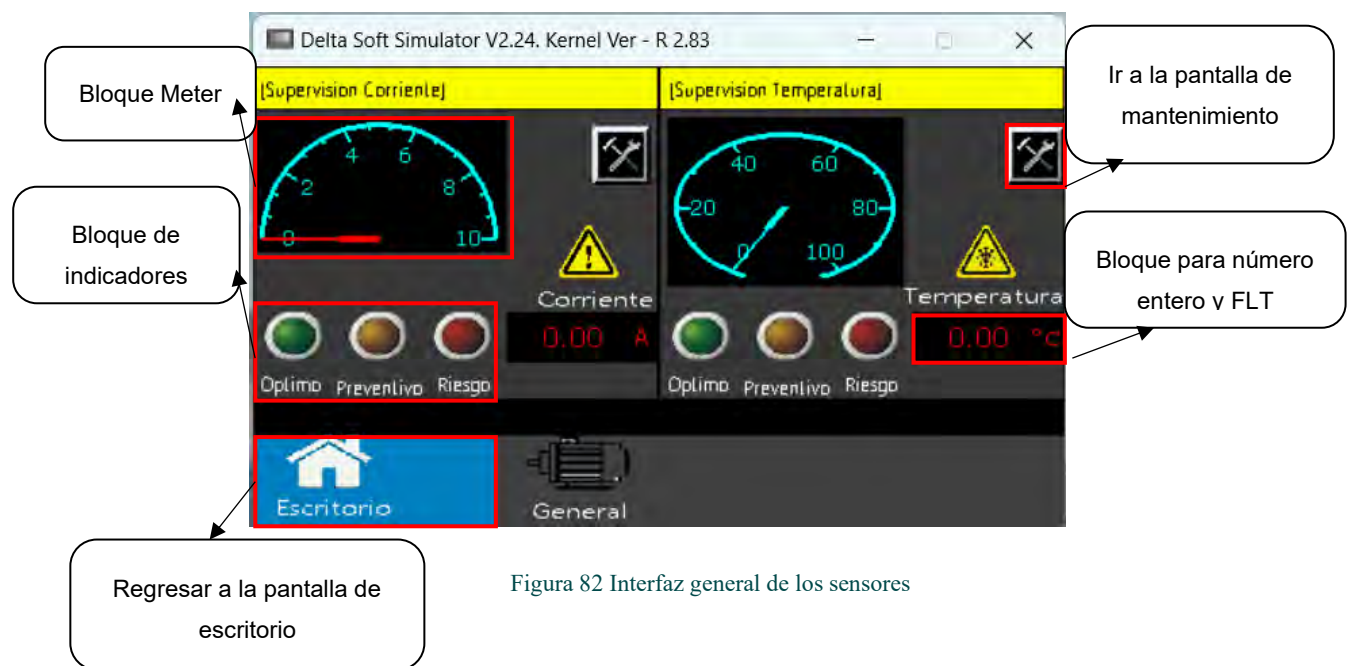


Figura 82 Interfaz general de los sensores

Para la pantalla de mantenimiento se copió de nuevo los bloques cada sensor y se reubico en otra pantalla para tener una visualización más amplia y con la posibilidad de apagar o encender el motor cuando la operación del sistema requiera mantenimiento. Los botones se

configuran como se muestra en la figura 64, solamente que ahora se configura con las direcciones que se muestra en la figura 83. De igual manera se realiza la misma interfaz para el sensor de temperatura.



Figura 83 Pantalla de mantenimiento con botón ON/OFF para el actuador

3. **Dos botones generales (Botón Momentary)** esta tendrá la función de prender y apagar todo el sistema (ver la imagen 84). Los botones se configuran de la misma forma como se muestra en la figura 64 solamente con diferente dirección para ON el “Write Address” es **M0** y OFF es **M1**.



Figura 84 Botón de paro de emergencia

Si se desea apagar el sistema esta tiene que ser desactivada por medio de una contraseña (ver la figura 85) DOPSoft tiene una configuración de contraseña en la parte “Options” seguidamente se selecciona “Password setting”, en donde hay una escala del 0 al 7, el usuario podrá encriptar botones con diferentes contraseñas con los niveles de seguridad. En ese sentido el botón OFF es un nivel de seguridad tipo 1 y para configurar esa contraseña hay que dirigirse al botón, dando doble clic y en el recuadro ubicar “Details” y en “User security level” seleccionar el nivel con el que fue configurado la

contraseña. Entonces al oprimir el botón ahora se abrirá un recuadro y solamente se desactivará con la contraseña.

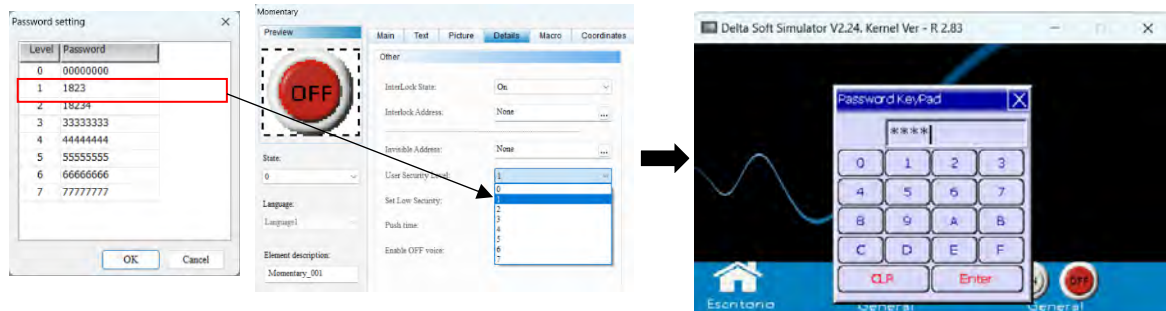
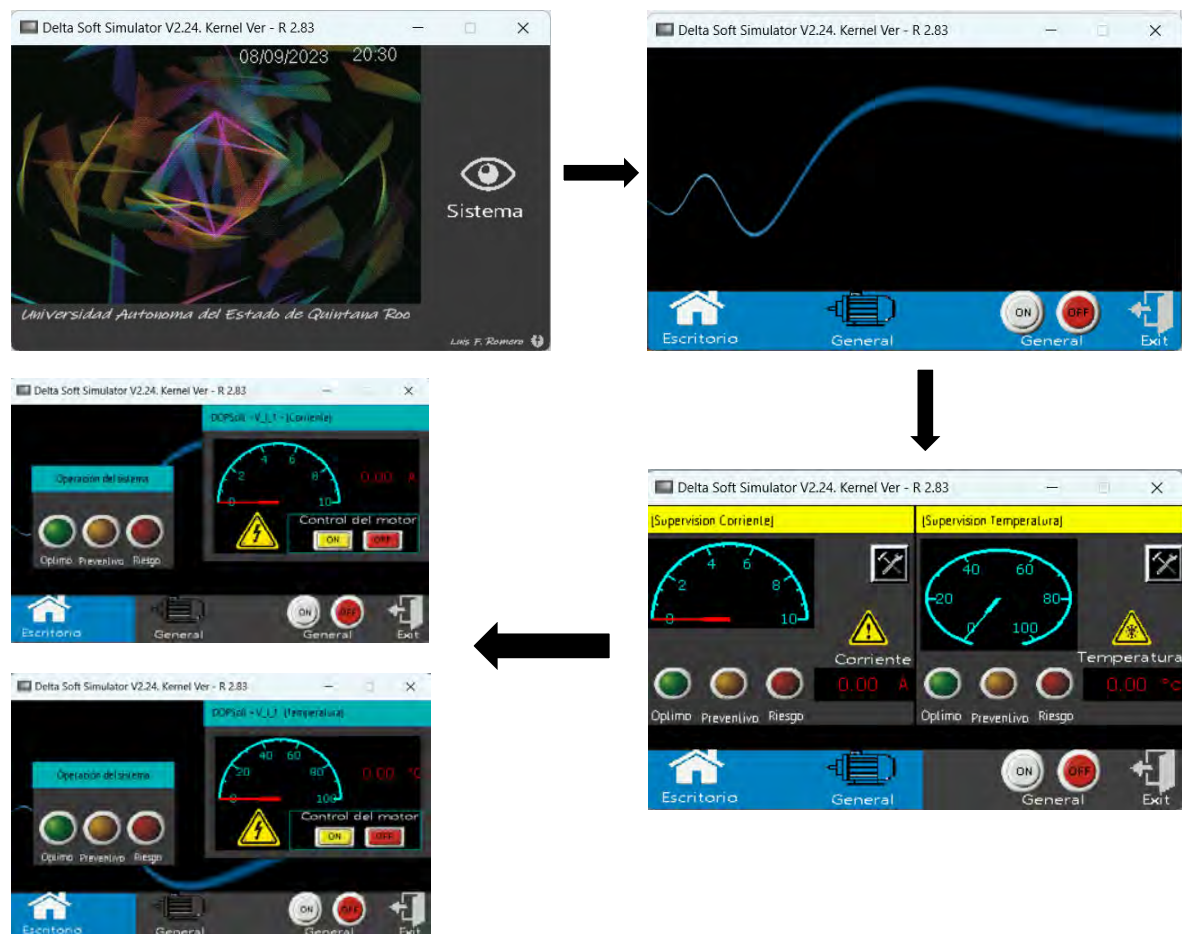


Figura 85 Asignar una contraseña de seguridad a un botón

4. Un botón de EXIT (Botón Goto Screen) esta tendrá la función de regresar a la pantalla principal.

Finalmente, las cinco pantallas personalizadas se visualizan de la siguiente manera



CAPÍTULO 5 SISTEMA SCADA

5.1 Conexión entre los dispositivos

El PLC DVP20SX2 cuenta con tres modelos, la diferencia entre ellos es el tipo de salida, las cuales pueden ser tipo: Relay, NPN Transistor y PNP transistor. En la tabla 17 se muestra la configuración I/O para cada uno de los modelos de esta serie.

Tabla 17 características de configuración I/O PLC DVP20SX2

| Model | Input | | Output | | I/O configuration | | | | | |
|----------|-------|------------------------|--------|-------------------|-------------------|------|------|------|------|-----|
| | Point | Type | Point | Type | Relay | | NPN | | PNP | |
| 20SX211R | 8 | DC (Sink Or Source) | 6 | Relay | V0+ | S/S | V0+ | S/S | V0+ | S/S |
| 20SX211T | | | | NPN Transistor | I0+ | X0 | I0+ | X0 | I0+ | X0 |
| | | | | | V10- | X1 | V10- | X1 | V10- | X1 |
| 20SX211S | 4 | Analog Input | 2 | Analog output | V1+ | X2 | V1+ | X2 | V1+ | X2 |
| I1+ | | | | | X3 | I1+ | X3 | I1+ | X3 | |
| V11- | | | | | X4 | V11- | X4 | V11- | X4 | |
| V2+ | | | | | X5 | V2+ | X5 | V2+ | X5 | |
| I2+ | | | | | X6 | I2+ | X6 | I2+ | X6 | |
| V12- | | | | | X7 | V12- | X7 | V12- | X7 | |
| V3+ | | | | | C0 | V3+ | UP | V3+ | UP | |
| I3+ | | | | | Y0 | I3+ | ZP | I3+ | ZP | |
| V13- | | | | | Y1 | V13- | Y0 | V13- | Y0 | |
| FE | | | | | Y2 | FE | Y1 | FE | Y1 | |
| VO0 | ● | VO0 | Y2 | VO0 | Y2 | | | | | |
| IO0 | C1 | IO0 | Y3 | IO0 | Y3 | | | | | |
| VO1 | Y3 | VO1 | Y4 | VO1 | Y4 | | | | | |
| IO1 | Y4 | IO1 | Y5 | IO1 | Y5 | | | | | |
| AG | Y5 | AG | ● | AG | ● | | | | | |

La tabla 17 ayuda a conocer qué tipo de PLC se está manejando, en este caso podría ser de tipo NPN o PNP por la distribución de sus entradas y salidas, pero en la parte trasera del PLC se encuentra modelo (ver la figura 86).

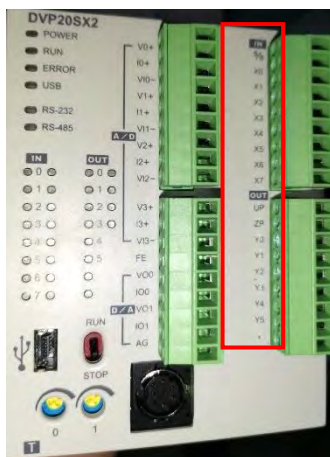


Figura 86 PLC DVP20SX211TT NPN Transistor

A continuación, se presenta el diagrama de conexión del transistor output NPN del modelo SX211T donde:

UP: Es el común y que cuenta con un paro de emergencia utilizando un interruptor externo. En el interior del PLC cuenta con un nodo donde hay fusibles de 5~10A en la terminal compartiendo los contactos de salida para proteger el circuito de salida.

ZP: Es el positivo de la fuente de alimentación DC (5V-30V)

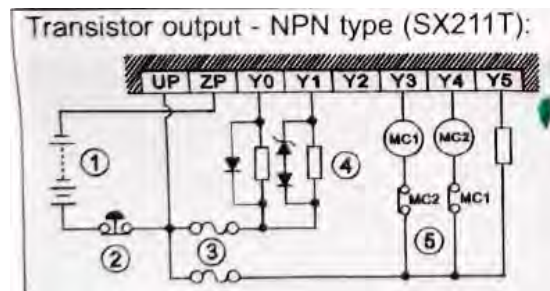


Figura 87 Conexión de UP y ZP del PLC SX211T

A continuación, se presenta el diagrama de conexión del relevador con la salida del PLC Y0 (ver la figura 88).

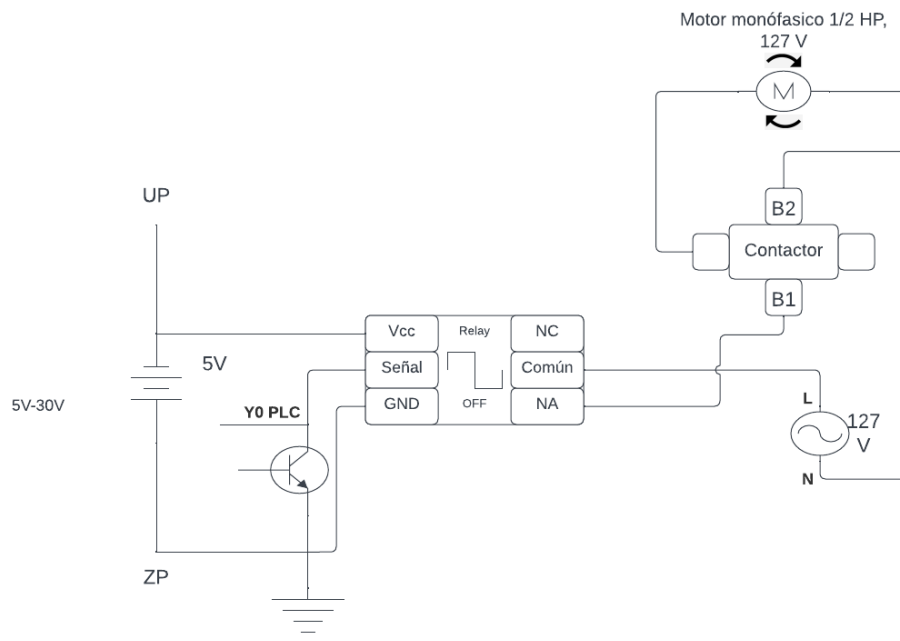


Figura 88 Diagrama de conexión del Relay, contactor y motor monofásico

Ahora se presenta el diagrama de conexión del sensor de temperatura LM35 al PLC (ver la figura 89)

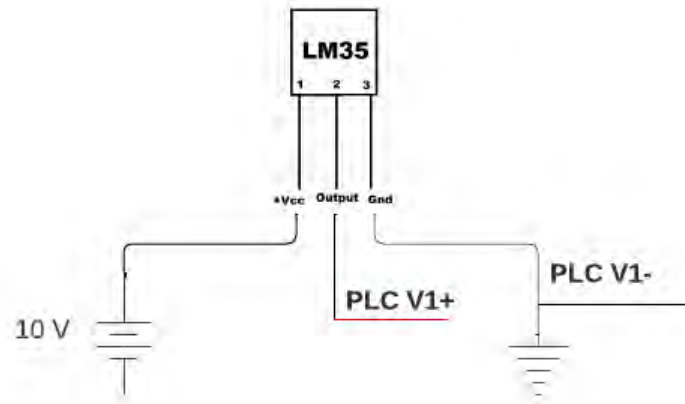


Figura 89 Diagrama de conexión del sensor de temperatura LM35

Por último, se presenta el diagrama de conexión del sensor de corriente LA 25-NP al PLC (ver la figura 90)

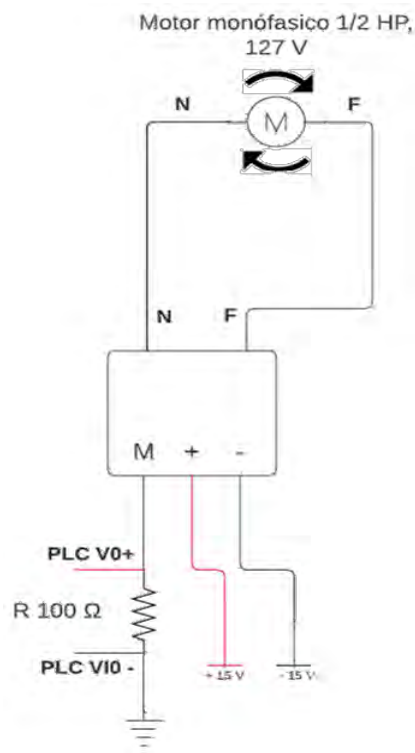


Figura 90 Diagrama de conexión del sensor de corriente LA 25-NP

El sistema SCADA queda instalada de la siguiente manera (ver la figura 91).

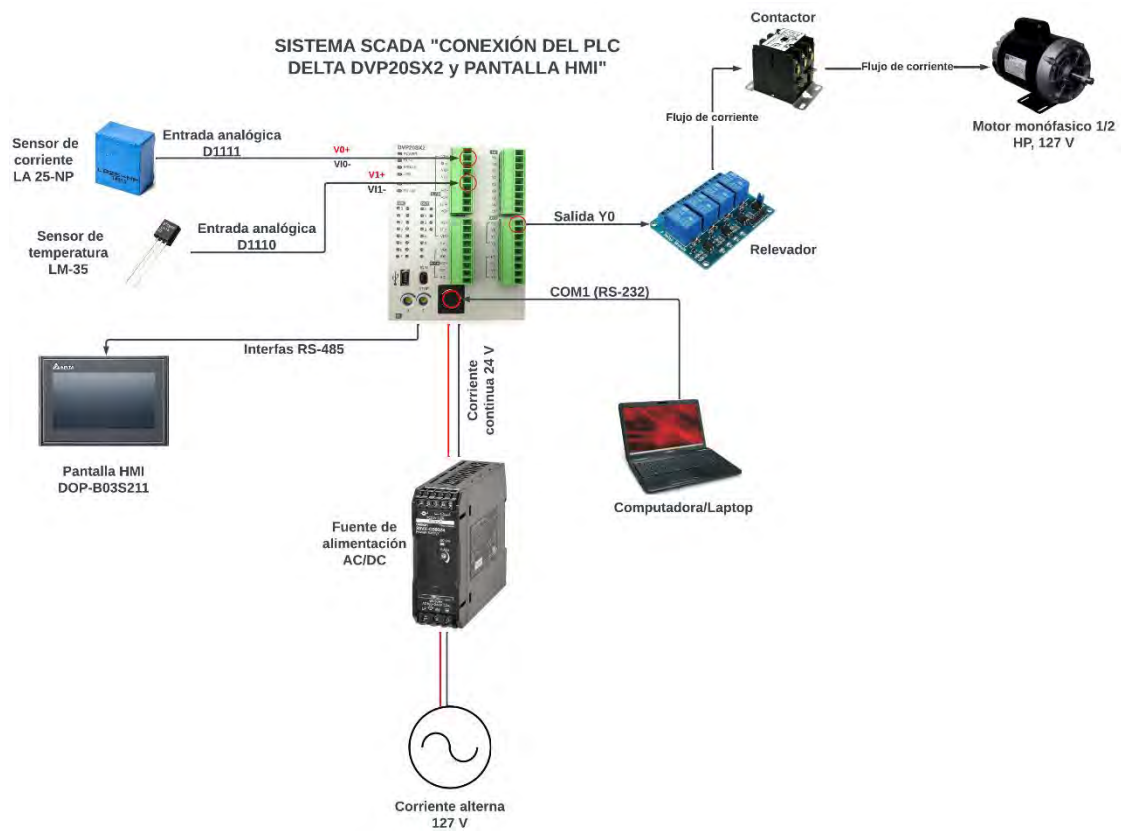


Figura 91 Sistema Scada Conexiones de los sensores, actuadores, pantalla HMI, fuente de alimentación AC/DC y comunicación RS-232

Una vez conectado se copia el programa en ISPSoft de la siguiente manera. En el software COMMGR se agrega el PLC (ver la figura 91). En este punto ya debe estar conectado el PLC al PC.

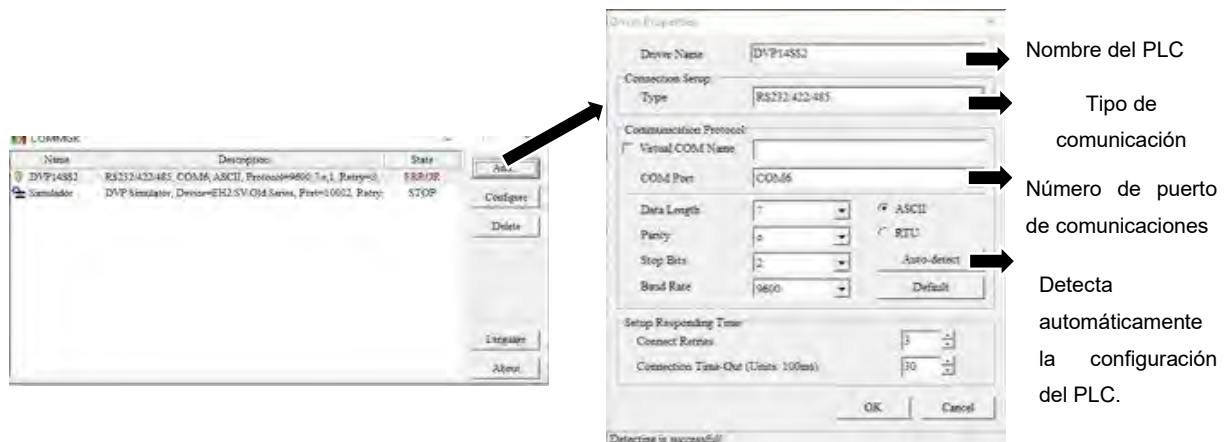


Figura 92 Paso para copilar el programa

Finalmente, en el programa copilamos con el siguiente botón y automáticamente empezara a transferir la programación de la PC al PLC (ver la figura 93).

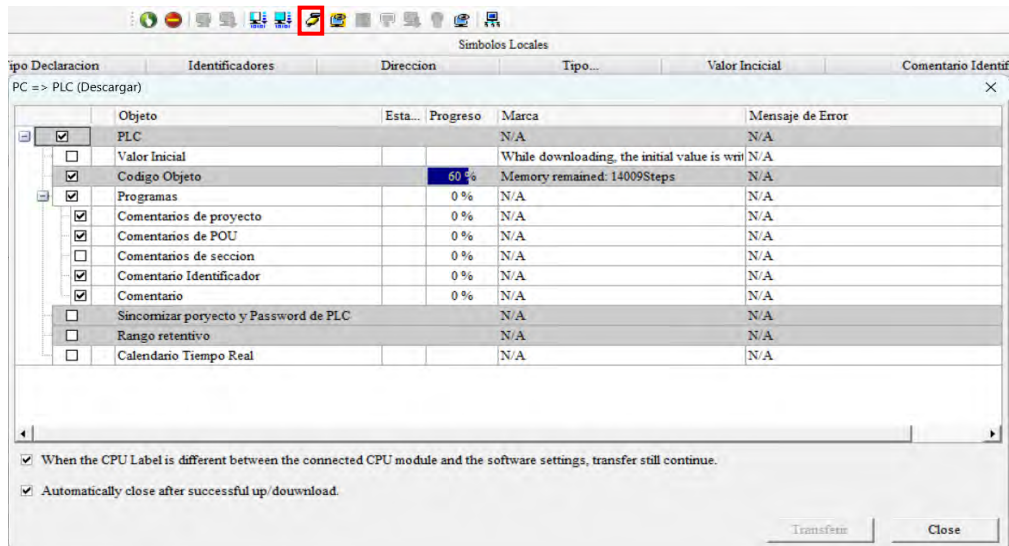


Figura 93 copilando el programa

Y para copilar la interfaz en DOPSoft solamente es seleccionar el siguiente botón (ver la figura 94)

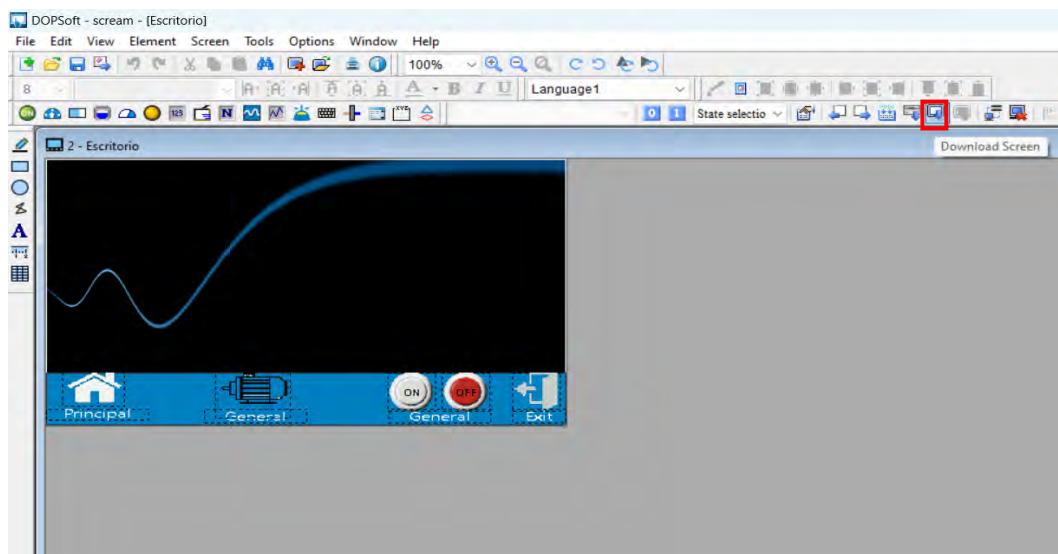


Figura 94 Copilando la interfaz para la pantalla HMI

5.2 Resultados

Instalación de todos los equipos utilizados (ver figura 95)



Figura 95 Visualización del prototipo

Conexiones de los sensores analógicos y la salida a relevador (ver la figura 96).



Figura 96 Conexiones físicas

Visualización de la pantalla al momento de presionar el botón de paro de emergencia únicamente se desactivará al ingresar el Password de primer nivel (ver la figura 97).



Figura 97 Visualización de la pantalla al desactivar con Password

Medición de corriente con un amperímetro de gancho en la fase del motor monofásico (ver la figura 98). Las mediciones de corriente sí coinciden con la de la pantalla HMI.



Figura 98 Medición de corriente con amperímetro de gancho y comparando la medición del PLC en la pantalla HMI.

Medición de temperatura con un infrarrojo Fluke a comparación con la lectura del sensor LM35 en la pantalla HMI (ver la figura 99).



Figura 99 Medición de temperatura Fluke y LM35 en la pantalla HMI

CONCLUSIÓN

Este tema ha servido para conocer como en nuestro país la automatización ha logrado tener un gran impacto en las grandes industrias. La Universidad Autónoma del Estado de Quintana Roo está cada vez incursionando en las tecnologías de la automatización, para que los alumnos conozcan y puedan crear sistemas que facilite y mejore la calidad de vida de las personas dentro de la industria o institución.

La implementación de un sistema SCADA conlleva una aplicación de diversos conocimientos adquiridos durante la formación como ingeniero en Sistema de Energía. Se involucran áreas de electrónica, electricidad, maquinas eléctricas, programación entre otras. El objetivo de esta tesis fue implementar un sistema SCADA para control y monitoreo de las variables eléctricas y de temperatura de un motor monofásico usando PLC y pantalla HMI.

A lo largo de este trabajo se desarrolló y documentó todos los procedimientos para la implementación de este sistema, con la finalidad de facilitar a los alumnos y a los ingenieros que se vean interesados en equipos DELTA. Esto sienta las bases para la programación de los dos softwares: ISPSOft y DOPSOft descritos en el capítulo tres y cuatro.

Por lo tanto, en el capítulo cinco da respuesta a los alcances propuestos, logrando una programación y un interfaz adecuada para el monitoreo en tiempo real (corriente y temperatura) de un motor monofásico. Sin embargo, este proyecto puede ser mejorado con los sensores adecuados que se utilizan en las industrias, de igual forma agregando mas dispositivos de control como variador de frecuencia. Además, con la placa del motor se podría monitorear las siguientes variables; potencia activa (kW), reactiva (kVAr) y aparente (kVA), etc.

Bibliografía

- Aguilera Martinez, P. (2002). *Definición de un autómata programable*. San Nicolas De Los Garza: Universidad Autonoma de Nuevo Leo.
- Ba Villarreal, A. A. (2021). *Desarrollo y diseño de una interfaz para red de PLC's basada en internet de las cosas*. Chetumal: Universidad de Quintana Roo.
- Cerrada, M. (2011). Concepción del supervisor para ISA 95. *Diagnóstico de fallas basado en modelos: Una solución factible para el desarrollo de aplicaciones SCADA en tiempo real*, 6.
- CIBERCHA. (5 de Abril de 2022). *Ciber Chambas*. Obtenido de Instalar Software DOPSOFT 2.0 para programar las HMI de la marca Delta: <https://www.ciberchambas.com/instalar-dopsoft2-hmi-delta/>
- Corona Ramírez, L. G., Abarca Jiménez, G. S., & Mares Carreño, J. (2019). *Sensores y Actuadores aplicaciones con Arduino*. México: Patria.
- Daneri, P. (2008). *PLC automatización y control industrial*. Buenos Aires: Hispano Americana HASA.
- DELTA. (2011). *DVP-ES2/EX2/SS2/SA2/SX2 Manual de Operación*.
- DELTA. (2018). *Delta DIAView SCADA System User Manual*. Medellin Colombia: DELTA.
- DELTA. (2023). *Controladores lógicos programables-Serie DVP*. Obtenido de PLC- Controladores lógicos programables: <https://www.deltaww.com/en-US/products/060301/ALL/>
- Eduardo, I. C. (2012). Sistemas de automatización. En I. C. Eduardo, *Sistemas de Automatización* (pág. 8). Santa Clara: Feijóo.
- Emilio, G. M. (2020). Los fundamentos modernos de la Automática. En G. M. Emilio, *Automatización de Procesos Industriales. Robótica y Automática*. (págs. 28-29). Valencia: Universitat Politècnica de València.
- Farnell. (2023). *Resumen del producto*. Obtenido de LA 25-NP: <https://es.farnell.com/lem/la-25-np/current-transducer-25a-pcb/dp/1617404#:~:text=El%20LA%2025->

NP%20de%20LEM%20es%20un%20transductor,externa%20y%20capacidad%20de%20soportar%20sobrecarga%20de%20corriente.

JM Industrial. (2023). *¿Cómo funciona?* Obtenido de Termocupla: <https://www.jmi.com.mx/termocupla>

L. Blevins, T., K. McMillan, G., K. Wojsznis, W., & W. Brown, M. (2008). *¿Qué Es la pirámide de la automatización?* En M. D. Lozano Lozano, & R. A. Zamora Muñoz, *Tecnologías y herramientas de ingeniería asociadas a los niveles superiores de la pirámide de la automatización*. (pág. 22). Bolívia: Universidad tecnológica de Bolívar facultad de ingeniería electrónica .

Lozano Lozano, M. D., & Zamora Muñoz, R. A. (2008). *Tecnologías y herramientas de ingeniería asociadas a los niveles superiores de la pirámide de la automatización*. Bolívia: Universidad tecnológica de Bolívar facultad de ingeniería electronica cartagena de indias D.T. y C.

Peña Palacios, C. A., & Lechuga Sierra, E. J. (2008). *Sensor de temperatura*. Bolivar: Universidad tecnologica de Bolivar facultad de ingenieria electrica y electronica .

Pérez López, E. (2015). *Los sistemas SCAD en la automatización industrial*. Recinto Grecia: Universidad de Costa Rica .

PROCOEN. (27 de 02 de 2018). *Sensor de corriente de núcleo cerrado*. Obtenido de Sensores de corriente, características que debe conocer: <https://procoen.com/sensores-de-corriente/>

Rodriguez de Avila, J. E. (2012). *Buenas Prácticas para diseño de HMI de alto rendimiento*. Colombio: Universidad tecnologica de Bolivar.

Rodríguez Penin, A. (2007). *Sistemas SCADA -2a edición*. Barcelona: Marcombo.

Rodriguez Penin, A. (2007). *Sistemas SCADA-Guía práctica*. España: Marcombo.

Rodríguez Penin, A. (2012). *Sistema SCADA 3a edición*. Barcelona: Marcombo.

Santacruz Ron, L. P. (2017). *Diseño e implementación de sistemas colectores de pelusa y horómetros para una máquina de coser industrial overlock y una recubridora*. Escuela Politécnica Nacional.

- SDI. (2022). *Sensores de corriente*. Obtenido de ¿Cómo funcionan los sensores?: <https://sdindustrial.com.mx/blog/sensores/>
- SICMA21. (11 de 10 de 2021). *Qué es un HMI y sus características*. Obtenido de Qué es un HMI y cómo funciona: <https://www.sicma21.com/que-es-un-hmi-y-como-funciona/>
- Texas Instruments. (2017). LM35 Precision Centigrade Temperature Sensors. *Datasheet*, 1.
- WEG AUTRIAL. (22 de Agosto de 2019). *Aplicaciones de Scada*. Obtenido de Qué es SCADA, aplicaciones y beneficios: <https://www.autrial.es/n-que-es-scada-aplicaciones-y-beneficios-691-13849-es.html>
- WITORG. (2023). *La pirámide de la automatización* . Obtenido de Pirámide de la automatización e industria 4.0: <https://www.witorg.org/piramide-de-la-automatizacion-e-industria-4-0/>

Anexos 1 Mapa de memoria para los SX2

| | | | |
|----------|-------------------|--|--|
| X | Entradas | X0~X377, sistema numérico octal, 256 puntos máximo | |
| Y | Salidas | Y0~Y377, sistema numérico octal, 256 puntos máximo | |
| M | Relé Auxiliar | General | M0~M511 M768~M999 M2000~M2047 |
| | | Enclavado | M512~M767 M2048~M4095 |
| | | Especial | M1000~M1999 |
| S | Punto escalonado | Punto Escalonado inicial | S0~S9 |
| | | Retorno a punto cero | S10~S19 |
| | | Enclavado | S20~S127 |
| | | General | S128~S911 |
| | | Alarma | S912~S1023 |
| T | Temporizador | 100ms (M1028 = ON, T64~T126: 10ms), | T0~T126 T128~T183 |
| | | | T184~T199 para subrutinas |
| | | | T250~T255 (acumulable) |
| | | 10ms (M1038 = ON, T200~T245: 1ms), | T200~T239 |
| | | | T240~T245 (acumulable) |
| | | 1ms | T127 T246~T249 (acumulable) |
| C | Contador | Conteo progresivo de 16 bits | C0~C111 C128~C199 |
| | | | C112~C127 |
| | | Conteo progresivo/regresivo de 32 bits | C200~C223 |
| | | | C224~C231 |
| D | Registro de datos | General | D0~D407, 408 palabras D600~D999, 400 palabras D3920~D9999, 6080 palabras |
| | | Enclavado | D408~D599, 192 palabras D2000~D3919, 1920 palabras |

Anexos 2 Lista numérica de instrucciones

Control de bucle

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|---|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 00 | CJ | - | ✓ | Salto condicional | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 01 | CALL | - | ✓ | Llamar subrutina | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 02 | SRET | - | - | Retorno a subrutina | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 03 | IRET | - | - | Retorno a interrupción | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 04 | EI | - | - | Habilitar interrupción | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 05 | DI | - | - | Deshabilitar interrupción | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 06 | FEND | - | - | El fin del programa principal (primer fin) | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 07 | WDT | - | ✓ | Actualizar temporizador guardián | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 08 | FOR | - | - | Inicio de un bucle For-Next | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 09 | NEXT | - | - | Fin de un bucle For-Next | ✓ | ✓ | ✓ | ✓ | 1 | - |

Comparación de transmisión

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|---------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 10 | CMP | DCMP | ✓ | Comparar | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 11 | ZCP | DZCP | ✓ | Comparar zona | ✓ | ✓ | ✓ | ✓ | 9 | 17 |
| 12 | MOV | DMOV | ✓ | Mover | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 13 | SMOV | - | ✓ | Cambiar movimiento | ✓ | ✓ | ✓ | ✓ | 11 | - |
| 14 | CML | DCML | ✓ | Complementar | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 15 | BMOV | - | ✓ | Bloquear movimiento | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 16 | FMOV | DFMOV | ✓ | Llenar movimiento | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 17 | XCH | DXCH | ✓ | Intercambiar | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 18 | BCD | DBCD | ✓ | Convertir BIN a BCD | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 19 | BIN | DBIN | ✓ | Convertir BCD a BIN | ✓ | ✓ | ✓ | ✓ | 5 | 9 |

Cuatro operaciones aritméticas

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|----------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 20 | ADD | DADD | ✓ | Adición | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 21 | SUB | DSUB | ✓ | Sustracción | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 22 | MUL | DMUL | ✓ | Multipliación | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 23 | DIV | DDIV | ✓ | División | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 24 | INC | DINC | ✓ | Incremento | ✓ | ✓ | ✓ | ✓ | 3 | 5 |
| 25 | DEC | DDEC | ✓ | Decremento | ✓ | ✓ | ✓ | ✓ | 3 | 5 |
| 26 | WAND | DAND | ✓ | Palabra lógica AND | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 27 | WOR | DOR | ✓ | Palabra lógica OR | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 28 | WXOR | DXOR | ✓ | Lógico XOR | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 29 | NEG | DNEG | ✓ | Complemento a 2 (negación) | ✓ | ✓ | ✓ | ✓ | 3 | 5 |

Rotación y desplazamiento

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|----------------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 30 | ROR | DROR | ✓ | Girar a la derecha | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 31 | ROL | DROL | ✓ | Rotar a la izquierda | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 32 | RCR | DRCR | ✓ | Girar a la derecha con acarreo | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 33 | RCL | DRCL | ✓ | Girar a la izquierda con acarreo | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 34 | SFTR | - | ✓ | Bit Cambia a la derecha | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 35 | SFTL | - | ✓ | Bit cambia a la izquierda | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 36 | WSFR | - | ✓ | Palabra cambia a la derecha | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 37 | WSFL | - | ✓ | Palabra cambia a la izquierda | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 38 | SFWR | - | ✓ | Cambiar registro a escribir | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 39 | SFRD | - | ✓ | Cambiar registro a leer | ✓ | ✓ | ✓ | ✓ | 7 | - |

Procesamiento de datos

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|--------------------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 40 | ZRST | - | ✓ | Zona restablecida | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 41 | DECO | - | ✓ | Decodificar | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 42 | ENCO | - | ✓ | Codificar | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 43 | SUM | DSUM | ✓ | Suma de bits activos | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 44 | BON | DBON | ✓ | Verificar estado de bit especificado | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 45 | MEAN | DMEAN | ✓ | Media | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 46 | ANS | - | - | Establecer anunciador temporizado | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 47 | ANR | - | ✓ | Restablecer anunciador | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 48 | SQR | DSQR | ✓ | Raíz cuadrada | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 49 | FLT | DFLT | ✓ | Punto flotante | ✓ | ✓ | ✓ | ✓ | 5 | 9 |

Procesamiento de alta velocidad

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|--|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 50 | REF | - | ✓ | Actualizar | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 51 | REFF | - | ✓ | Actualizar y ajustar filtro | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 52 | MTR | - | - | Matriz de entrada | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 53 | - | DHSCS | - | Contador de alta velocidad SET | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 54 | - | DHSCR | - | Restablecer contador de alta velocidad | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 55 | - | DHSZ | - | Comparar zona de alta velocidad | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 56 | SPD | - | - | Detección de velocidad | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 57 | PLSY | DPLSY | - | Salida de pulso | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 58 | PWM | - | - | Modulación de ancho de pulso | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 59 | PLSR | DPLSR | - | Rampa de pulso | ✓ | ✓ | ✓ | ✓ | 9 | 17 |

Instrucciones útiles

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|--------------------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 60 | IST | - | - | Estado inicial | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 61 | SER | DSER | ✓ | Buscar una pila de datos | - | ✓ | ✓ | ✓ | 9 | 17 |
| 62 | ABSD | DABSD | - | Secuenciador de tambor absoluto | - | ✓ | ✓ | ✓ | 9 | 17 |
| 63 | INCD | - | - | Secuenciador de tambor incremental | - | ✓ | ✓ | ✓ | 9 | - |
| 64 | TTMR | - | - | Temporizador de entrenamiento | - | ✓ | ✓ | ✓ | 5 | - |
| 65 | STMR | - | - | Temporizador especial | - | ✓ | ✓ | ✓ | 7 | - |
| 66 | ALT | - | ✓ | Estado alterno | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 67 | RAMP | DRAMP | - | Valor variable de rampa | - | ✓ | ✓ | ✓ | 9 | 17 |
| 68 | DTM | - | ✓ | Transformación de datos y movimiento | - | ✓ | ✓ | ✓ | 9 | - |
| 69 | SORT | DSORT | - | Clasificación de datos | - | ✓ | ✓ | ✓ | 11 | 21 |

Pantalla externa de entrada/salida

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|-----------------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 70 | TKY | DTKY | - | Entrada de 10 teclas | - | ✓ | ✓ | ✓ | 7 | 13 |
| 71 | HKY | DHKY | - | Entrada de teclas hexadecimales | - | ✓ | ✓ | ✓ | 9 | 17 |
| 72 | DSW | - | - | Interruptor DIP | - | ✓ | ✓ | ✓ | 9 | - |
| 73 | SEGD | - | ✓ | Decodificador de 7 segmentos | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 74 | SEGL | - | - | 7 segmentos con enclavamiento | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 75 | ARWS | - | - | Interruptor de flecha | - | ✓ | ✓ | ✓ | 9 | - |
| 76 | ASC | - | - | Conversión de código ASCII | - | ✓ | ✓ | ✓ | 11 | - |
| 77 | PR | - | - | Imprimir (salida de código ASCII) | - | ✓ | ✓ | ✓ | 5 | - |

Entrada/salida en serie

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|---|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 78 | FROM | DFROM | ✓ | Leer datos CR de módulos especiales | ✓ | ✓ | ✓ | ✓ | 9 | 17 |
| 79 | TO | DTO | ✓ | Escribir datos CR en módulos especiales | ✓ | ✓ | ✓ | ✓ | 9 | 17 |
| 80 | RS | - | - | Comunicación en serie | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 81 | PRUN | DPRUN | ✓ | Ejecución en paralelo | - | ✓ | ✓ | ✓ | 5 | 9 |
| 82 | ASCII | - | ✓ | Convertir HEX a ASCII | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 83 | HEX | - | ✓ | Convertir ASCII a HEX | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 84 | CCD | - | ✓ | Verificar código | - | ✓ | ✓ | ✓ | 7 | - |
| 85 | VRRD | - | ✓ | Volumen leído | - | - | ✓ | ✓ | 5 | - |
| 86 | VRSC | - | ✓ | Escala de volumen leído | - | - | ✓ | ✓ | 5 | - |
| 87 | ABS | DABS | ✓ | Valor absoluto | ✓ | ✓ | ✓ | ✓ | 3 | 5 |
| 88 | PID | DPID | - | Control PID | ✓ | ✓ | ✓ | ✓ | 9 | 17 |

Instrucciones básicas

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|--|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 89 | PLS | - | - | Salida de flanco ascendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 90 | LDP | - | - | Operación de detección de flanco ascendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 91 | LDF | - | - | Operación de detección de flanco descendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 92 | ANDP | - | - | Conexión en serie de flanco ascendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 93 | ANDF | - | - | Conexión en serie de flanco descendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 94 | ORP | - | - | Conexión en paralelo de flanco ascendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 95 | ORF | - | - | Conexión en paralelo de flanco descendente | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 96 | TMR | - | - | Temporizador | ✓ | ✓ | ✓ | ✓ | 4 | - |
| 97 | CNT | DCNT | - | Contador | ✓ | ✓ | ✓ | ✓ | 4 | 6 |
| 98 | INV | - | - | Operación inversa | ✓ | ✓ | ✓ | ✓ | 1 | - |
| 99 | PLF | - | - | Salida de flanco descendente | ✓ | ✓ | ✓ | ✓ | 3 | - |

Instrucciones de comunicación

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|--------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 100 | MODRD | - | - | Leer datos de Modbus | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 101 | MODWR | - | - | Escribir datos de Modbus | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 102 | FWD | - | - | Operación directa de VFD | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 103 | REV | - | - | Operación inversa de VFD | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 104 | STOP | - | - | Detener VFD | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 105 | RDST | - | - | Leer estado VFD | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 106 | RSTEF | - | - | Restablecer VFD anormal | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 107 | LRC | - | ✓ | Verificación de LRC | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 108 | CRC | - | ✓ | Verificación de CRC | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 150 | MODRW | - | - | Leer/escribir MODBUS | ✓ | ✓ | ✓ | ✓ | 11 | - |
| 206 | ASDRW | - | - | Unidad de servo R/W ASDA | - | ✓ | ✓ | ✓ | 7 | - |

Operación de punto flotante

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|---|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 110 | - | DECMP | ✓ | Comparar punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 111 | - | DEZCP | ✓ | Comparar zona de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 112 | | DMOVR | ✓ | Mover datos de punto flotante | ✓ | ✓ | ✓ | ✓ | | 9 |
| 116 | - | DRAD | ✓ | Grado → Radián | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 117 | - | DDEG | ✓ | Radián → Grado | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 118 | - | DEBCD | ✓ | Conversión de flotante a científica | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 119 | - | DEBIN | ✓ | Conversión de científica a flotante | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 120 | - | DEADD | ✓ | Suma de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 121 | - | DESUB | ✓ | Resta de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 122 | - | DEMUL | ✓ | Multiplicación de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 123 | - | DEDIV | ✓ | División de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 124 | - | DEXP | ✓ | Operación de exponente flotante | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 125 | - | DLN | ✓ | Operación de logaritmo natural flotante | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 126 | - | DLOG | ✓ | Operación de logaritmo flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 127 | - | DESQR | ✓ | Raíz cuadrada de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 128 | - | DPOW | ✓ | Operación de potencia de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 129 | INT | DINT | ✓ | Flotante a entero | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 130 | - | DSIN | ✓ | Seno | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 131 | - | DCOS | ✓ | Coseno | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 132 | - | DTAN | ✓ | Tangente | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 133 | - | DASIN | ✓ | Seno inverso | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 134 | - | DACOS | ✓ | Coseno inverso | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 135 | - | DATAN | ✓ | Tangente inversa | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 172 | - | DADDR | ✓ | Suma de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 173 | - | DSUBR | ✓ | Resta de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 174 | - | DMULR | ✓ | Multiplicación de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 175 | - | DDIVR | ✓ | División de punto flotante | ✓ | ✓ | ✓ | ✓ | - | 13 |

Instrucción adicional

| API | Mnemónico | | PULSE | Función | Aplicable a | | | ESCALONES | | |
|-----|-----------|---------|-------|--|-------------|-----|-----|-----------|---------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 143 | DELAY | - | ✓ | Retardo | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 144 | GPWM | - | - | Salida PWM general | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 147 | SWAP | DSWAP | ✓ | Intercambio de byte | ✓ | ✓ | ✓ | ✓ | 3 | 5 |
| 154 | RAND | DRAND | ✓ | Número aleatorio | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 168 | MVM | DMVM | ✓ | Enmascarar y combinar bits designados | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 176 | MMOV | - | ✓ | Conversión de 16 bits→32 bits | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 177 | GPS | - | - | Recepción de datos GPS | ✓ | ✓ | ✓ | ✓ | 5 | - |
| 178 | - | DSPA | - | Posicionamiento de celda solar | ✓ | ✓ | ✓ | ✓ | - | 9 |
| 179 | WSUM | DWSUM | ✓ | Suma de múltiples dispositivos | ✓ | ✓ | ✓ | ✓ | 7 | 13 |
| 202 | SCAL | - | ✓ | Cálculo de valor proporcional | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 203 | SCLP | DSCLP | ✓ | Cálculo de valor proporcional de parámetro | ✓ | ✓ | ✓ | ✓ | 9 | 13 |
| 205 | CMPT | - | ✓ | Comparación de tabla | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 207 | CSFO | - | - | Detectar velocidad y salida proporcional | ✓ | ✓ | ✓ | ✓ | 7 | - |

Control de posicionamiento

| API | Mnemónico | | PULSE | Función | Aplicable a | | | ESCALONES | | |
|-----|-----------|---------|-------|--|-------------|-----|-----|-----------|---------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 155 | - | DABSR | - | Posición absoluta leída | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 156 | - | DZRN | - | Retorno a cero | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 157 | - | DPLSV | - | Salida de pulso de velocidad ajustable | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 158 | - | DDRVI | - | Control de posición relativa | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 159 | - | DDRVA | - | Control de posición absoluta | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 191 | - | DPPMR | - | Movimiento relativo de punto a punto de 2 ejes | ✓ | - | ✓ | ✓ | - | 17 |
| 192 | - | DPPMA | - | Movimiento absoluto de punto a punto de 2 ejes | ✓ | - | ✓ | ✓ | - | 17 |
| 193 | - | DCIMR | - | Interpolación inversa de posición relativa de 2 ejes | ✓ | - | ✓ | ✓ | - | 17 |
| 194 | - | DCIMA | - | Interpolación inversa de posición absoluta de 2 ejes | ✓ | - | ✓ | ✓ | - | 17 |
| 195 | - | DPTPO | - | Salida de pulso de un solo eje conformea la tabla | ✓ | ✓ | ✓ | ✓ | - | 13 |
| 197 | - | DCLLM | - | Control de posición de cierre de bucle | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 198 | - | DVSP0 | - | Salida de pulso de velocidad variable | ✓ | ✓ | ✓ | ✓ | - | 17 |
| 199 | - | DICF | ✓ | Cambiar frecuencia inmediatamente | ✓ | ✓ | ✓ | ✓ | - | 13 |

Calendario de tiempo real

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|-------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 160 | TCMP | - | ✓ | Comparar datos horarios | ✓ | ✓ | ✓ | ✓ | 11 | - |
| 161 | TZCP | - | ✓ | Comparar zona horaria | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 162 | TADD | - | ✓ | Suma de datos horarios | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 163 | TSUB | - | ✓ | Resta de datos horarios | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 166 | TRD | - | ✓ | Leer datos horarios | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 167 | TWR | - | ✓ | Escribir datos horarios | ✓ | ✓ | ✓ | ✓ | 3 | - |
| 169 | HOUR | DHOUR | - | Medidor de horas | ✓ | ✓ | ✓ | ✓ | 7 | 13 |

Código gris

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|-------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 170 | GRY | DGRY | ✓ | BIN → Código gris | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 171 | GBIN | DGBIN | ✓ | Código gris → BIN | ✓ | ✓ | ✓ | ✓ | 5 | 9 |

Operación de matriz

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|-----------------------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 180 | MAND | - | ✓ | Matriz AND | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 181 | MOR | - | ✓ | Matriz OR | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 182 | MXOR | - | ✓ | Matriz XOR | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 183 | MXNR | - | ✓ | Matriz XNR | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 184 | MINV | - | ✓ | Matriz inversa | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 185 | MCMP | - | ✓ | Comparación de matriz | ✓ | ✓ | ✓ | ✓ | 9 | - |
| 186 | MBRD | - | ✓ | Leer bit de matriz | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 187 | MBWR | - | ✓ | Escribir bit de matriz | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 188 | MBS | - | ✓ | Cambiar bit de matriz | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 189 | MBR | - | ✓ | Rotar bit de matriz | ✓ | ✓ | ✓ | ✓ | 7 | - |
| 190 | MBC | - | ✓ | Conteo de estado de bit de matriz | ✓ | ✓ | ✓ | ✓ | 7 | - |

Operación lógica tipo contacto

| API | Mnemónico | | PULSE | Función | Aplicable a | | | | ESCALONES | |
|-----|-----------|---------|-------|----------------------|-------------|-----|-----|-----|-----------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 215 | LD& | DLD& | - | $S_1 \text{ y } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 216 | LD | DLD | - | $S_1 \text{ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 217 | LD^ | DLD^ | - | $S_1 \text{ ^ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 218 | AND& | DAND& | - | $S_1 \text{ y } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 219 | AND | DAND | - | $S_1 \text{ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 220 | AND^ | DAND^ | - | $S_1 \text{ ^ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 221 | OR& | DOR& | - | $S_1 \text{ y } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 222 | OR | DOR | - | $S_1 \text{ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 223 | OR^ | DOR^ | - | $S_1 \text{ ^ } S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |

Comparación tipo contacto

| API | Mnemónico | | PULSE | Función | Aplicable a | | | ESCALONES | | |
|-----|-----------|---------|-------|----------------|-------------|-----|-----|-----------|---------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 224 | LD= | DLD= | - | $S_1 = S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 225 | LD> | DLD> | - | $S_1 > S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 226 | LD< | DLD< | - | $S_1 < S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 228 | LD<> | DLD<> | - | $S_1 \neq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 229 | LD<= | DLD<= | - | $S_1 \leq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 230 | LD>= | DLD>= | - | $S_1 \geq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 232 | AND= | DAND= | - | $S_1 = S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 233 | AND> | DAND> | - | $S_1 > S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 234 | AND< | DAND< | - | $S_1 < S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 236 | AND<> | DAND<> | - | $S_1 \neq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 237 | AND<= | DAND<= | - | $S_1 \leq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 238 | AND>= | DAND>= | - | $S_1 \geq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 240 | OR= | DOR= | - | $S_1 = S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 241 | OR> | DOR> | - | $S_1 > S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 242 | OR< | DOR< | - | $S_1 < S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 244 | OR<> | DOR<> | - | $S_1 \neq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 245 | OR<= | DOR<= | - | $S_1 \leq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 246 | OR>= | DOR>= | - | $S_1 \geq S_2$ | ✓ | ✓ | ✓ | ✓ | 5 | 9 |

Control de bit específico

| API | Mnemónico | | PULSE | Función | Aplicable a | | | ESCALONES | | |
|-----|-----------|---------|-------|---|-------------|-----|-----|-----------|---------|---------|
| | 16 bits | 32 bits | | | ES2 EX2 | SS2 | SA2 | SX2 | 16 bits | 32 bits |
| 266 | BOUT | DBOUT | - | Bit especificado de salida de una palabra | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 267 | BSET | DBSET | - | Establecer ON el bit específico de una palabra | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 268 | BRST | DBRST | - | Restablecer bit especificado de una palabra | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 269 | BLD | DBLD | - | Cargar contacto NO por bit especificado | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 270 | BLDI | DBLDI | - | Cargar contacto NC por medio de bit específico | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 271 | BAND | DBAND | - | Conectar contacto NO en serie por bit especificado | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 272 | BANI | DBANI | - | Conectar contacto NC por bit especificado | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 273 | BOR | DBOR | - | Conectar contacto NO en paralelo por bit especificado | ✓ | ✓ | ✓ | ✓ | 5 | 9 |
| 274 | BORI | DBORI | - | Conectar contacto NC en paralelo por bit especificado | ✓ | ✓ | ✓ | ✓ | 5 | 9 |