RESIDENCIA PROFESIONAL.

CFE Gerencia de Transmisión del Sureste.

Departamento de Automatización Y Control

Proyecto:

Sistema de monitoreo de las subestaciones zona sureste mediante Incontrol.

Asesor interno:

ing. Odilio Magdaleno Orozco.

Asesor externo:

Ing. Víctor Manuel Flores Carrera.

Residente:

Jacob Gómez José.

Número De Control: 05270271.

Capitulo 1. Generalidades.

Capitulo 2. Fundamentos Teóricos.

Capitulo 3. Desarrollo del Proyecto.

Capitulo 1. Generalidades.

1.1 Introducción.

En estos tiempos donde la necesidad de la adquisición de información de forma verídica, constante y en tiempo real, ha dado lugar al diseño de nuevas tecnologías para cumplir con la normatividad mundial con la finalidad de tener un mejor control de lo que se está realizando en toda la infraestructura que se tiene, y de esta manera tomar decisiones más precisas en tiempo y forma.

La gerencia regional de transmisión sureste, tiene en cuenta esta necesidad, y por eso se planteara el desarrollo del proyecto en base al sistema SCADA, el cual se explicara en capítulos siguientes.

El Proyecto lleva distintas fases y dividido en varios capítulos que son los siguientes:

- Capítulo 1: Se detalla la Descripción de la empresa sus actividades y necesidades. Se Señala el problema y su Planteamiento.
- Capítulo 2: En este capítulo se encuentra el fundamento teórico, de donde se baso el proyecto.
- Capítulo 3: En este apartado se encuentran las propuestas del proyecto, basándose en los resultados de los capítulos Anteriores. Por último informamos sobre las conclusiones y recomendaciones que ofrecemos a la empresa.

1.2 Información general de la institución o empresa donde se desarrollo el proyecto.

En 1937, México tenía 18.3 millones de habitantes; de los cuales, únicamente siete millones (38%) contaban con servicio de energía eléctrica, proporcionado con serias dificultades por tres empresas privadas. La oferta no satisfacía la demanda, las interrupciones de luz eran constantes y las tarifas muy elevadas. Además, esas empresas se enfocaban a los mercados urbanos más redituables, sin contemplar en sus planes de expansión a las poblaciones rurales, donde habitaba más de 62% de la población. Para dar respuesta a esas situaciones que no permitían el desarrollo económico del país, el Gobierno federal decidió crear, el 14 de agosto de 1937, la Comisión Federal de Electricidad, que en una primera etapa se dio a la tarea de construir plantas generadoras para satisfacer la demanda.

Y con ello beneficiar a más mexicanos mediante el bombeo de agua de riego, el arrastre y la molienda; pero sobre todo, con alumbrado público y para casas habitación.

Los primeros proyectos de CFE se emprendieron en Teloloapan, Guerrero; Pátzcuaro, Michoacán; Suchiate y Xía, en Oaxaca, y Ures y Altar, en Sonora. En 1938, la empresa tenía apenas una capacidad de 64 Kw., misma que, en ocho años, aumentó hasta alcanzar 45,594 kW. Entonces, las compañías privadas dejaron de invertir y nuestra empresa se vio obligada a generar energía para que éstas la revendieran.

En 1960, de los 2,308 MW de capacidad instalada en el país, CFE aportaba 54%; la Mexican Light, 25%; la American and Foreign, 12%, y el resto de las compañías, 9%. Sin embargo, a pesar de los esfuerzos de generación y electrificación, para esas fechas apenas 44% de la población contaba con electricidad. Tal situación del Sector Eléctrico Mexicano motivó al entonces Presidente Adolfo López Mateos a nacionalizar la industria eléctrica, el 27 de septiembre de 1960.

A partir de entonces, se comenzó a integrar el Sistema Eléctrico Nacional, extendiendo la cobertura del suministro y acelerándola.

Industrialización del país. Para ello, el Estado mexicano adquirió los bienes e instalaciones de las compañías privadas, mismas que operaban con serias deficiencias, por la falta de inversión y los problemas laborales.

Para 1961, la capacidad total instalada en el país ascendía a 3,250 MW. CFE vendía 25% de la energía que producía y su participación en la propiedad de centrales generadoras de electricidad pasó de cero a 54%. En poco más de 20 años, nuestra empresa había cumplido uno de sus más importantes cometidos: ser la entidad rectora en la generación de energía eléctrica. En esa década, la inversión pública se destinó en más de 50% a obras de

infraestructura. Con parte de estos recursos se construyeron importantes centros generadores, entre ellos los de Infiernillo y Temascal.

En esos años se instalaron plantas generadoras por el equivalente a 1.4 veces lo hecho hasta entonces, alcanzando, en 1971, una capacidad instalada de 7,874 MW. Al finalizar los 70, se superó el reto de sostener el mismo ritmo de crecimiento, al instalarse entre 1970 y 1980 centrales generadoras por el equivalente a 1.6 veces, para llegar a una capacidad instalada de 17,360 MW. En la década de los 80, el crecimiento fue menos espectacular, principalmente por la disminución en la asignación de recursos. No obstante, en 1991 la capacidad instalada ascendía a 26,797 MW. Actualmente, la capacidad instalada en el país es de 46,672 MW*, de los cuales 47.55% corresponde a generación termoeléctrica de CFE; 19.85% a *productores independientes de energía (PIE); 22.04% a hidroelectricidad; 5.57% a centrales carboeléctricas; 2.06% a geotérmica; 2.92% a nucleoeléctrica, y 0.005% a eoloeléctrica.

Debe señalarse que, en los inicios de la industria eléctrica mexicana operaban varios sistemas aislados, con características técnicas diferentes; llegando a coexistir casi 30 voltajes de distribución, siete de alta tensión para líneas de transmisión y dos frecuencias eléctricas de 50 y 60 hertz. Ello dificultaba el suministro de electricidad a todo el país, por lo que CFE definió y unificó los

Criterios técnicos y económicos del Sistema Eléctrico Nacional, normalizando los voltajes de operación, con la finalidad de estandarizar los equipos, reducir sus costos y los tiempos de fabricación, almacenaje e inventariado. Luego, unificó la frecuencia a 60 hertz en todo el país e integró los sistemas de transmisión, en el Sistema Interconectado Nacional.

Otro rubro con logros contundentes, se refiere a la red de transmisión de electricidad, el cual se compone actualmente de: 46,688 kilómetros de líneas de 400, 230 y 161 kV; 327 subestaciones de potencia con una capacidad de 135,238 MVA, y 46,633 kilómetros de líneas de subtransmisión de 138 kV y tensiones menores. Por su parte, el sistema de distribución (que también estaba en ceros en 1937) cuenta actualmente con 1,545 subestaciones con 40,719 MVA de capacidad; 6,775 circuitos de distribución con una longitud de 368,405 kilómetros; 982,702 transformadores de distribución con una capacidad de 32,189 MVA; 235,951 kilómetros de líneas secundarias de baja tensión y 600,663 kilómetros de acometidas.

El día de hoy, 127,621 localidades tienen electricidad y sus habitantes reciben una atención más rápida y cómoda en las 951 oficinas de atención al público y los

1,884 cajeros CFE mático, en los que se puede pagar el recibo de luz a cualquier hora, los 365 días del año.

*Incluye 19 Centrales de productores independientes de energía (PIE) con una capacidad total de 9,266 MW, las cuales se incluyen en el apartado de centrales generadoras.

1.3 Área específica relacionada directamente con el proyecto.

Descripción del Área de Control.

Funciones de la subgerencia de control e informática:

-Reemplazar y ampliar la vida útil del equipo de Control instalados en el ámbito geográfico de la Gerencia Regional de Transmisión Sureste.

-Proponer las normas, métodos, políticas y criterios orientados al mantenimiento y reposición del equipo de Control instalado en el ámbito geográfico de Gerencia Regional de Transmisión Sureste.

-Proponer con las subÀreas de Transmisión y Transformación el registro y análisis de fallas mediante controles estadísticos para apoyo en la toma de decisiones.

-Coordinar el reemplazo y ampliar la vida útil de los equipos de control instalados en el ámbito de la GRTSE.

-Proponer las normas, métodos, políticas y criterios orientados al mantenimiento y reposición del equipo de control instalado en el ámbito de la GRTSE.

-Proponer con las subÀreas el registro y análisis de fallas mediante controles estadísticos, para apoyo en la toma de decisiones.

-Establecer y coordinar procedimientos de trabajo, que permitan eficientar el mantenimiento del equipo de control supervisorio, esquemas de automatización y sistemas de informática; así como incrementar el índice de disponibilidad en estos sistemas.

-Controlar, evaluar y promover el cumplimiento de los programas de trabajo, de medidas correctivas, objetivos negociados y presupuestos, así como vigilar la normatividad establecida en la especialidad.

-Diseñar, administrar y operar el sistema de información y control local de estación.

-Diseñar, administrar y operar los sistemas de información y redes.

1.3.1 Ubicación.

La oficina de la GRTSE esta Ubicada en Carretera panamericana No. 5675 int. 500 m., Colonia Plan de Ayala, Tuxtla Gutiérrez Chiapas México.

1.3.2 Misión y Visión

Misión.

Asegurar la disponibilidad de la red eléctrica de potencia mediante una eficiente planeación y ejecución del mantenimiento y modernización, satisfaciendo las expectativas de nuestros clientes, respetando el medio ambiente y fomentando una mejor calidad de vida a nuestros trabajadores.

Visión.

Ser una organización productiva, rentable y humana con desarrollo sostenido y sustentable, caracterizada por su actitud de servicio, mejora continua, liderazgo tecnológico, certificada en sus procesos y con un desempeño comparable con las mejores del mundo.

1.4 Antecedentes

En la gerencia regional de transmisión sureste tiene a su cargo 5 subareas que dependen de ella, La GRTS se encarga de gestionar y tener un control de sus actividades de las 5 subareas que son S.E. Tuxtla, S.E. Malpaso S.E. Villahermosa, S.E. Tapachula, S.E. Istmo, que estas 5 se encargan del mantenimiento de las líneas de transmisión de sus subestaciones correspondientes, estas subestaciones y actividades se explicaran detalladamente en el capítulo 1.

En resumen es de suma importancia que la Gerencia Regional de Transmisión Sureste tenga en tiempo real la información de cómo operan las demás subareas en torno a producción, y transmisión de Energía eléctrica.

En Comisión federal de electricidad, se cuenta con un proyecto piloto de monitoreo de información en tiempo real, pero aun posee ciertas fallas de conexión de información, fallas en las DNP's que causan cortes en el flujo de información.

Este proyecto se encarga de utilizar los conocimientos adquiridos en la carrera de Ingeniería Electrónica utilizando simbología en Relay Ladder Logic y orientándolo a la programación de diagramas de Verdad SCADA y structure Text, combinando con HTML y PHP, Integrando todos estos conocimientos y tecnologías podremos proponer el desarrollo de un sistema más robusto de adquisición de Información.

1.5 Planteamiento del problema.

Comisión federal de Electricidad (CFE) desarrolla continuamente un control y monitoreo de las subestaciones a fines de tener un mejor funcionamiento.
Existen adversidades y fallas; meteorológica, sociales, ambientales y de sistemas, que ocasionan pérdidas fundamentales para la empresa y afectan a todos los interesados.
Una línea de transmisión está expuesta a diversos acontecimientos que pueden provocar fallas y dan como consecuencias las interrupciones de estas mismas para su distribución: como lo son inundaciones, incendios, vandalismo, tormentas eléctricas, o incidentes con la naturaleza.
El sistema de operación a grandes rasgos de una subestaciones consta de dispositivos como lo son: las cuchillas, interruptores, protecciones que integra una bahías, cada una de esta última está comprendida con dichos dispositivos.
La bahía se encuentra conectada a las líneas de buss (400 kW,230KW,115KW) dependiendo en que proceso de transformación se encuentre.

1.5.1 Alcances Y Limitaciones

Alcances.

- Monitoreo en tiempo real.
- Prevención temprana para cualquier falla.
- Reducción de paros.
- Reducción de accidentes.
- Reducción de costos de operación.

Limitaciones.

- Falta de control de operación.
- Errores de flujo de información por causa de falla de UTR (Unidad Terminal Remota).
- Errores de fallas mecánicas.

1.6 Nombre del proyecto.

Sistema de monitoreo e información de fallas en las subestaciones, zona sureste, mediante Incontrol.

1.7 Objetivos generales y específicos.

1.7.1 Objetivo General Del Proyecto.

Desarrollar y diseñar con RLL (Lógica De Escalera De Relevadores) y Structure Text(Estructura De Texto) dentro del sistema Scada (SAD400) mensajes de alarma para el mejoramiento en las condiciones de seguridad y monitoreo en las subestaciones para resolver más eficientemente fallas que tenga como consecuencia la interrupción de este servicio.

1.7.2 Objetivos Específicos.

Los objetivos a realizarse mediante el proyecto consisten en:

- Monitoreo en tiempo real para identificar fallas.
- Reducción de costos de operación y posibles accidentes.
- Fiabilidad de información.
- Informar de fallas a los encargados de las subestaciones de donde se
- Origina la falla por medio de mensajes de texto a su celular
- Especificando que cuchilla se abrió o cerró.

Cabe mencionar que el sofware InControl fue adquierido por la empreza para cumplir dichos objetivos especificos.

1.8 Justificación Del Proyecto.

Comisión Federal de Electricidad (Gerencia de transmisión del sureste) es una empresa que proporciona servicio confiable, segura, oportuna, con la mejor eficiencia posible.

Debido a lo que implica la empresa es necesario un monitoreo en tiempo real para la adquisición de información de manera oportuna y fiable.
Con respecto al estado de las subestaciones:

- Comportamiento de interruptores y cuchillas.
- Fallos y alarmas de las protecciones.
- Cantidad de KW generada, etc.

Y es de suma importancia informar a las personas encargadas de la subestación donde se origino la falla y corregir a la brevedad posible evitando fallas del servicio, y no genere costos de operación, ni accidentes.

Capitulo 2. Fundamentos Teóricos.

2.1 Wonderware.

Wonderware es el líder en el mercado en operaciones de tiempo real del software de administrador. El software de Wonderware ofrece importantes reducciones de costos asociados con el diseño, construcción, despliegue y mantenimiento de aplicaciones seguras y estandarizadas para la fabricación y operaciones de infraestructura. Nuestras soluciones permiten a las compañías sincronizar sus operaciones de producción y de industria con los objetivos de negocio, la obtención de la velocidad y flexibilidad para alcanzar una rentabilidad sostenida.

Wonderware Soluciones de Industrias de Energía Eléctrica

Con una amplia base instalada en la generación, transmisión y distribución de aplicaciones, Wonderware ofrece una refrescante alternativa a los tradicionales sistemas SCADA a través de la plataforma del sistema de Wonderware construido en ArchestrA Tecnología.

Reestructuración de la industria, los costes energéticos, la creciente demanda, la integración de la generación distribuida y otros factores a cabo servicios de electricidad en un entorno complejo dinámico que exige un flujo continuo de información en una solución escalable, confiable y seguro.

La aplicación de una sola automatizada subestación (SA) del sistema SCADA / GMS, SCADA / EMS, SCADA / DMS o buscar la máxima inteligente de Smart Grid, el software de Wonderware proporciona la plataforma que permite la integración de aplicaciones para crear un amplio entorno real de las operaciones de gestión del tiempo.

2.1.1 InTouch.

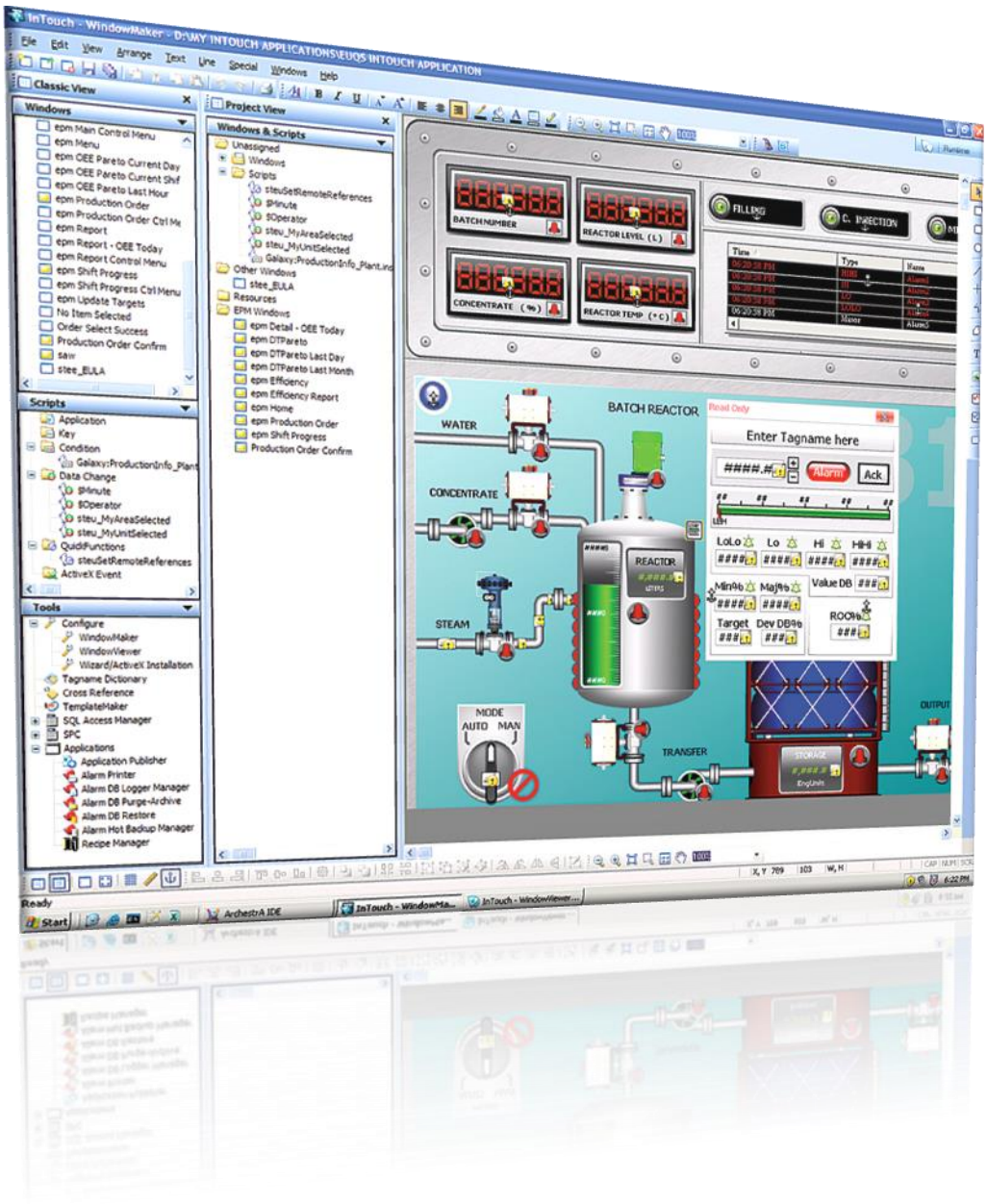


Figura 2.1 Intouch.

2.1.1.1 Describcion

El software de InTouch proporciona una visualización gráfica que tiene sus operaciones de gestión, control y optimización a un nivel completamente nuevo. La reputación de InTouch HMI está por encima de todos los demás. Lo que la industria ahora sabe como Human Machine Interface (HMI) .

Todo esto conduce a las normas impulsadas por bien diseñado sistemas que maximizan la productividad, optimizar la eficiencia de los usuarios, aumentar la calidad, y menor desarrollo, mantenimiento y gastos operacionales para ayudar hacer que la empresa sea mas eficaz.

2.1.1.2 Beneficios

Verdaderamente la legendaria facilidad de uso que permite a los desarrolladores y operadores de forma rápida y sencilla a ser más productivos

La integración de dispositivos sin igual y la conectividad a prácticamente todos los dispositivos y sistema.

Impresionante representación gráfica visual y la interacción con su operación trae la información correcta a las personas adecuadas en el momento adecuado

Historia de la ruta de migración sin interrupciones versión de software que significa que sus aplicaciones HMI de inversión está protegida.

2.1.1.3 Entre las principales funcionalidades

Resolución de gráficos independientes y símbolos visuales inteligentes que traen sus instalaciones a la vida en tu pantalla de la computadora

Las secuencias de comandos sofisticado para ampliar y personalizar las aplicaciones para sus necesidades específicas.

En tiempo real distribuidos alarmante, con vistas históricas para el análisis de
Built-in, en tiempo real y tendencias históricas

Controles de Microsoft ActiveX y. NET controles de integración

2.1.2 InControl.

Figura 2.2 Incontrol

2.1.2.1 Describcion,

InControl™ 7.11 es un componente de control, de arquitectura abierta, que le permite diseñar, crear, probar y ejecutar programas para controlar sus procesos más rápidos. El software está basado en los sistemas operativos Microsoft® Windows NT® Workstation, Windows® 2000 Professional y Windows XP.

InControl ofrece una alternativa de software a los PLCs, más poderosa y a menor costo. Ejecutándose en PCs y sistemas operativos estándar, InControl ofrece un paquete de control más robusto, bastamente superior en conectividad y con sofisticadas capacidades lógicas para manejar complejos

procesos discretos y por lotes. También ofrece mayor capacidad si se compara con los micro-PLCs, y tiene un menor costo punto por punto que un PLC de gama media. InControl contiene editores que manejan Lógica de Escalera IEC61131-3 (RLL), Diagramas Secuénciales (SFC) y Texto Estructurado (ST). También puede usar una variedad de sistemas populares y de bajo costo de proveedores de I/O (Opto22, Grayhill, Sixnet, etc.). InControl usa las últimas tecnologías distribuidas de Microsoft, como controles ActiveX y COM+, para asegurar un diseño robusto y flexible. Estas características mejoran la funcionalidad y robustez del producto, mientras permiten una integración transparente con una amplia gama de soluciones de hardware y software de diferentes fabricantes.

Al momento de conectarse a los productos de la familia FactorySuite® 2000, InControl cuenta con un avanzado sistema de scripting y un avanzado motor de ejecución en tiempo real, además de poderosas capacidades de comunicaciones para una sencilla pero versátil integración de información. InControl ofrece toda la funcionalidad de un cliente y servidor OPC, un diseño modular y la interfaz hacia el protocolo SuiteLink, exclusivo de Wonderware, lo cual le concede a este producto una excepcional escalabilidad y flexibilidad. InControl ofrece rendimiento sin precedentes utilizando Microsoft NT y Windows 2000 en tiempo real de las capacidades. InControl puede ejecutar un PID en bucle de menos de 10 microsegundos, o 1000 líneas de la lógica o en un ST RLL programa en menos más de 1 milisegundo en la mayoría de los PC estándar.

2.1.2.2 InControl 7,11 ™ Tiempo Real Sistema de Control

Wonderware ® InControl 7,11 ™ es una aplicación en tiempo real del motor que se adapte a una amplia gama de procesos de fabricación que requieren alta velocidad, procesamiento de datos deterministas y lógica. Se ejecuta en un sistema abierto, InControl ofrece un robusto paquete de control que proporciona conectividad y sofisticadas capacidades de la lógica de ejecución para el manejo de scripting complejo, discreto y tramitar las solicitudes. InControl 7,11 expande sus capacidades de control con el apoyo de Microsoft ® Windows NT ® y Windows ® 2000 basado en motores de tiempo de ejecución; actoryObjects ActiveX basado en controles de usuario y la costumbre. InControl del IEC61131-3 compatible con los editores, amplia gama de E / S interfaces e integrado OPC y SuiteLink ™ comunicaciones le permiten una solución a medida para satisfacer su aplicación exacta de requisitos.

Además proporciona InControl interfaces para una amplia variedad de abrir sistemas de E / S, y SuiteLink Servidores OPC. Cuando está conectado a Wonderware FactorySuite ® ®, añade InControl operativos avanzados y de información en tiempo real la capacidad de FactorySuite. Con su diseño modular, además de su OPC y SuiteLink comunicaciones, InControl ofrece un cierto grado de flexibilidad y escalabilidad que es inigualable por cualquier otro PLC o blanda lógica de paquete.

2.1.2.2.1 Aplicaciones.

InControl proporciona un control integrado y la solución de secuencias de comandos que reemplaza propiedad de los sistemas de control con arquitectura abierta de control basados en NT, proporcionando un menor costo de control con la arquitectura integrada de la conectividad, la transformación de gran alcance capacidad, y fácil capacidad de expansión. OEM pueden beneficiarse especialmente de la conectividad para las interfaces de comunicación y múltiples sistemas de E / S, editores y flexibles capacidades de ActiveX. Proceso de solicitudes beneficiarse de los motores en tiempo de ejecución distribuidos; poderoso control PID, herramientas de simulación y de alta E / S cuenta. Para aplicaciones de control de máquina incrustables InControl proporciona, en tiempo real discreto y flexible de control de la tecnología ActiveX.

2.1.2.3 Características Y Beneficios

Microsoft Windows 2000 y basados en NT InControl 7,11 nativa se basa en Microsoft NT, teniendo el máximo provecho de todos los tiempo real y extensibilidad NT proporciona capacidades. InControl apoya distribuido control a través de DCOM, con par-a-par comunicación incorporado en el producto.

2.1.2.4 Arquitectura Abierta

InControl 7,11 se puede utilizar en cualquier plataforma que apoya el Microsoft Windows NT y Windows 2000 sistemas operativos, incluidos los de panel plano industrial de trabajo, servidores SMP y abierto industriales controladores.

2.1.2.5 Fábrica de Objetos

Fábrica de Objeto PID, Plomo / Lag, Derivada, analógico de alarma, Deadtime Delay, integrador, limitador, Ratio de la estación, ICAccess para VB y las interfaces de puerto serie V.2

2.1.2.6 Comunicaciones Flexibles

Soporta una amplia gama de comunicaciones de red incluyendo SuiteLink cliente / servidor y SuiteLink OPC cliente / servidor OPC I / O support Apoya populares de E / S de dispositivos interfaces para Open Interfaces de red, así como legado de sistemas de E / S.

2.1.2.7 Normas Internacionales

Cumple con las normas IEC 61131-3, OPC normas Abrir el dispositivo y las normas de interfaz de red.

2.1.2.8 Características En Línea

Soporta una variedad de la supervisión en línea y edición Monitor de procesos, incluyendo el estatuto, la Fuerza de E / S, Edición en línea, de flujo de potencia Destacando, Simulación y Depuración. Fábrica de objetos de ActiveX extensible Apoyo Crear sus propios algoritmos en Visual Basic o C + + y convocatoria de InControl como un objeto ActiveX.

Ahora InControl apoya en los acontecimientos y objetos ActiveX puede ejecutar objetos ActiveX asincrónicamente.

2.1.2.9 ST Editor de texto

ST proporciona editor independiente de secuencias de comandos, y ahora ha mayor en línea de herramientas de depuración. Definidos por el usuario, Funciones y Bloques de Función Crear funciones definidas por el usuario y en función de los bloques RLL o ST idiomas. Estas funciones o de funciones Bloques pueden ser instanciado en RLL y ST programas.

2.1.2.10 Capacidades de Control Distribuido

Conectar a cualquier mando a distancia del motor de tiempo de ejecución, editar y descargar programas y supervisar el estado de todos estación central de desarrollo.

2.1.2.11 La capacidad de simulación

Proporciona herramientas para la simulación del mundo real a los procesos depurar y optimizar aplicaciones. Disponibles para usar como una aplicación independiente para supervisar variables y InControl depurar aplicaciones.

2.1.2.12 Integración Con Wonderware FactorySuite Componentes

InControl está fuertemente integrado con Wonderware ® InTouch ® y el resto de FactorySuite, proporcionando un poder sin precedentes y la productividad de la industria mundo. InControl 7.11 es un componente básico de tecnología de la FactorySuite. Es el componente de control que ejecuta en tiempo real de la lógica de control, se conecta a la fábrica piso de E / S, proporciona la capacidad de simulación y es una base de datos FactorySuite servidor. Es un motor de scripting para InTouch, motor de la lógica fase InBatch ™, y los datos servidor para IndustrialSQL Server ™ o SuiteVoyager™.

La integración comienza con la instalación. A continuación, a un tagName común navegador que, con mando a distancia InTouch de referencia, puede definir una única base de datos en InControl y fácilmente cualquier referencia o todas las etiquetas de InTouch. SuiteLink y el cliente OPC y servidor proporciona la conectividad sin fisuras entre-suite comunicación. Por último, tenemos un poderoso conjunto de herramientas InTouch en que te permite controlar fácilmente el motor de tiempo de ejecución de InControl y proporcionar acceso directo a InControl programas, proyectos y

tagnames. InControl del tiempo real lógica de ejecución capacidad de elogios por FactorySuite ofreciendo un conjunto completo de herramientas para el desarrollo basado en servidor de secuencias de comandos o aplicaciones de gran thinclient basado en FactorySuite proyectos.

2.1.2.13 Conectividad

OCX de serie, DeviceNet (Tecnologías de la SS), Profibus (SS Tecnologías), Optomux (Opto 22), Ethernet (Automatización directa), Abrir Modbus (Master Funcionalidad a través de TCP / IP, ASCII y RTU), Interbus-S G4 (Phoenix Contacto), AB1784KTX (Allen Bradley), Genius GE (GE), Profibus (sinérgicos Micro Systems), GE 90/30 (GE Fanuc), SDS de Honeywell, OPC Cliente, y SuiteLink Cliente.

2.1.2.14 Requisitos del sistema

Para Windows 2000 y NT y el desarrollo en tiempo de ejecución sistemas basados en el procesador Pentium sistema (Pentium 233 MHz y hasta procesadores recomendados), Soporta Sistemas SMP, con 32 MB de RAM (64 MB recomendado si se ejecuta conjuntamente con InTouch), requiere 50 MB de espacio en disco duro (instalación típica) y 25 MB (compacto de instalación).

Sólo requiere de tiempo de ejecución de la instalación menos de dos MB de espacio en disco duro que permite InControl runtimes para ser instalado en aplicaciones embebidas utilizando la memoria flash Sistemas operativos compatibles Tanto para el entorno de desarrollo y tiempo de ejecución motor, InControl 7,11 ejecutará en Windows NT 4.0 SP6a y Windows 2000 (versión 1).

2.2 Lógica De Escalera (Relay Ladder Logic(RLL))

Es una filosofía de la lógica de dibujo esquemas eléctricos. Ahora es un lenguaje gráfico muy popular para la programación de controladores lógicos programables (PLC). Fue originalmente inventado para describir la lógica hecho de relés. El nombre se basa en la observación de que los programas en esta lengua se asemejan a las escaleras de mano, con dos rieles verticales y horizontales de una serie de peldaños entre ellos.

Un programa en la lógica de escalera, también llamado diagrama de una escalera, es similar a un esquema de un conjunto de relé circuitos. Un argumento que la ayuda inicial de la adopción de la lógica de escalera es que una amplia gama de ingenieros y técnicos puedan comprender y utilizar sin mucho entrenamiento adicional, debido a la semejanza con sistemas de hardware familiar. (Este argumento se ha vuelto menos relevante dado que la mayoría de los programadores de la lógica de escalera tienen un fondo de software más convencionales en los lenguajes de programación, en la práctica y aplicación de la lógica de escalera tienen características. Tales como la ejecución secuencial y el apoyo para el control de flujo, características que hacen que la analogía con el hardware algo impreciso.) Lógica de escalera es

ampliamente utilizada para programar PLC, donde el control secuencial de un proceso o una operación de fabricación es obligatorio.

Como controladores lógicos programables se hizo más sofisticada, también ha sido utilizado en muy complejos sistemas de automatización. Los fabricantes de controladores lógicos programables en general, también asociados los sistemas de programación lógica de escalera. Normalmente, la lógica de escalera de dos fabricantes de idiomas no será completamente compatible; lógica de escalera es mejor pensar como un conjunto de lenguajes de programación estrechamente relacionados en lugar de un idioma (el IEC 61131-3 norma ha contribuido a reducir las diferencias innecesarias, pero la traducción de los programas entre los sistemas todavía requiere un trabajo importante). Incluso diferentes modelos de controlador programable dentro de la misma familia pueden tener diferentes escalas de notación de tal manera que los programas no pueden ser fácilmente intercambiados entre los modelos. Lógica de escalera puede ser pensada como un lenguaje basado en normas, en lugar de una lengua de procedimiento. Un "puesto" en la escala representa una regla. Cuando se aplique con relés y otros dispositivos electromecánicos, las diversas normas "ejecutar" simultánea e inmediatamente. Cuando se aplique en un controlador lógico programable, las normas son generalmente ejecutan en forma secuencial por el software, en un bucle continuo (exploración). Al ejecutar el bucle con la suficiente rapidez, por lo general, muchas veces por segundo, el efecto de la ejecución inmediata y simultánea es relativamente logrado hasta dentro de la tolerancia del tiempo necesario para ejecutar cada peldaño en el "bucle" (la "exploración").

Es algo similar a otras lenguas basado en normas, como las hojas de cálculo o de SQL. Sin embargo, el uso adecuado de los controladores programables requiere la comprensión de las limitaciones de la orden de ejecución de los peldaños.

2.3 Gráfico De Función Secuencial (Sequential Function Chart(SFC))

Es un lenguaje de programación gráfica utilizado para PLCs. Es uno de los cinco idiomas definidos por IEC 61131-3 estándar. El SFC se define en la norma IEC 848, "Preparación de la función de los sistemas de control de las cartas", y se basó en Grafcet (basada en binario redes de Petri. Se puede utilizar para programar los procesos que pueden dividirse en etapas.

Principales componentes de SFC son:

- Medidas, junto con las acciones
- Asociados con las transiciones condición lógica
- Dirigido vínculos entre los pasos y transiciones

Pasos en un diagrama de SFC puede ser activa o inactiva. Acciones sólo se ejecutan para medidas activas. Un paso puede ser activo a uno de dos motivos:

(1) Es un primer paso de la forma especificada por el programador (2) Se activa durante un ciclo de exploración y no se desactiva, ya Medidas cuando se activan todos los pasos anteriores y que se activa la conexión de la transición es superable (es decir, su condición de asociado es cierto).

Cuando una transición se pasa, todos los pasos anteriores se desactivan a la vez y después de todos los pasos a continuación se activan a la vez.

Acciones asociadas con las medidas pueden ser de varios tipos, los más relevantes se continua (N), Set (S) y Reset (R). Aparte de los evidentes efectos de Set y Reset, un N acción asegura que su objetivo es establecer la variable a 1, siempre y cuando el paso está activa. Una SFC norma establece que si dos medidas tienen una acción de N en el mismo objetivo, la variable nunca se debe restablecer a 0. También es posible insertar LD (Ladder Diagram), acciones dentro de un programa SFC (y esta es la manera estándar, por ejemplo, para trabajar sobre variables de tipo entero).

SFC es un lenguaje inherentemente paralelo en que el control de los flujos múltiples (POUS en el nivel del lenguaje) puede ser activo a la vez.

Extensiones no estándar de la lengua incluyen macroactions: es decir, las acciones dentro de un programa de unidad que influyen en el estado de la unidad de otro programa. Los más relevantes, tales macroaction es "forzar", en la que uno puede decidir el POU medidas activas de otro POU.

2.4 Textos Estructurados (ST.- Structured Text)

Es un lenguaje de marcas ligero creado para escribir textos de manera cómoda y rápida. Tiene la principal ventaja de que ese texto puede usarse para generar documentos equivalentes en HTML, TeX, docbook u otros lenguajes.

Actualmente se usa más reStructuredText, que es una revisión que mejora y amplía StructuredText.

Texto estructurado es uno de los 5 idiomas soportados por la IEC 61131-3 estándar. Está diseñado para los controladores de lógica programable (PLC). Se trata de un alto nivel de idioma que es el bloque estructurado y sintácticamente asemeja Pascal. Todos los idiomas comparten elementos comunes IEC61131. Las variables y llamadas a funciones son definidos por los elementos comunes a fin de diferentes idiomas se pueden utilizar en el mismo programa.

Instrucciones Complejas y anidadas que están soportadas por ST:

- Iteration loops (REPEAT-UNTIL; WHILE-DO)

- Conditional execution (IF-THEN-ELSE; CASE)
- Functions (SQRT(), SIN())

2.5 NORMA IEC 61131-3 ESTÁNDAR

En la actualidad aún siguen persistiendo sistemas de control específicos del fabricante, con programación dependiente y conexión compleja entre distintos sistemas de control. Esto significa para el usuario costos elevados, escasa flexibilidad y falta de normalización en las soluciones al control industrial.
IEC 61131 es el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar. Esta norma se divide en cinco partes:

- Parte 1: Vista general.
- Parte 2: Hardware.
- Parte 3: Lenguaje de programación.
- Parte 4: Guías de usuario.
- Parte 5: Comunicación.

IEC 61131-3 pretende es la base real para estandarizar los lenguajes de programación en la automatización industrial, haciendo el trabajo independiente de cualquier compañía. Hay muchas maneras de describir el trabajo desarrollado en la tercera parte de esta norma, indicaremos algunas de ellas son:

- IEC 61131-3 es el resultado del gran esfuerzo realizado por 7 multinacionales a los que se añaden muchos años de experiencia en el campo de la automatización industrial.
- Incluye 200 páginas de texto aproximadamente, con más de 60 tablas.
- IEC 61131-3 son las especificaciones de la sintaxis y semántica de un lenguaje de programación, incluyendo el modelo de software y la estructura del lenguaje.

Otra visión distinta es dividir el estándar en dos partes: (ver figura 2.3):

- Elementos comunes.
- Lenguajes de programación.

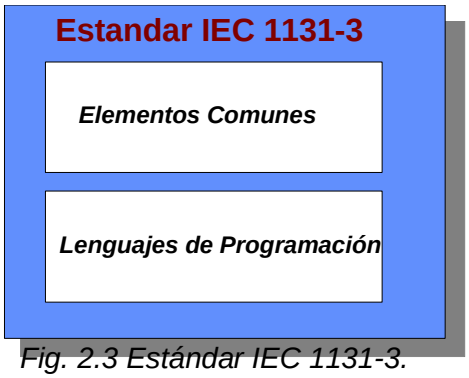


Fig. 2.3 Estándar IEC 1131-3.

2.5.1 Elementos Comunes

2.5.1.1 Tipos de datos

Dentro de los elementos comunes, se definen los tipos de datos. Los tipos de datos previenen de errores en una fase inicial, como por ejemplo la división de un dato tipo fecha por un número entero. Los tipos comunes de datos son: variables booleanas, número entero, número real, byte y palabra, pero también fechas, horas del día y cadenas (strings). Basado en estos tipos de datos, el usuario puede definir sus propios tipos de datos, conocidos como tipos de datos derivados. De este modo, se puede definir por ejemplo un canal de entrada analógica como un tipo de dato.

2.5.1.2 Variables

Las variables permiten identificar los objetos de datos cuyos contenidos pueden cambiar, por ejemplo, los datos asociados a entradas, salidas o a la memoria del autómata programable. Una variable se puede declarar como uno de los tipos de datos elementales definidos o como uno de los tipos de datos derivados. De este modo se crea un alto nivel de independencia con el hardware, favoreciendo la reusabilidad del software. La extensión de las variables está normalmente limitada a la unidad de organización en la cual han sido declaradas como locales. Esto significa que sus nombres pueden ser reutilizados en otras partes sin conflictos, eliminando una frecuente fuente de errores. Si las variables deben tener una extensión global, han de ser declaradas como globales utilizando la palabra reservada VAR_GLOBAL. Pueden ser asignados parámetros y valores iniciales que se restablecen al inicio, para obtener la configuración inicial correcta.

2.5.1.3 Configuración, recursos y tareas

Para entender esto mejor, vamos a ver el modelo de software, que define IEC 61131-3 (ver figura 2.4).

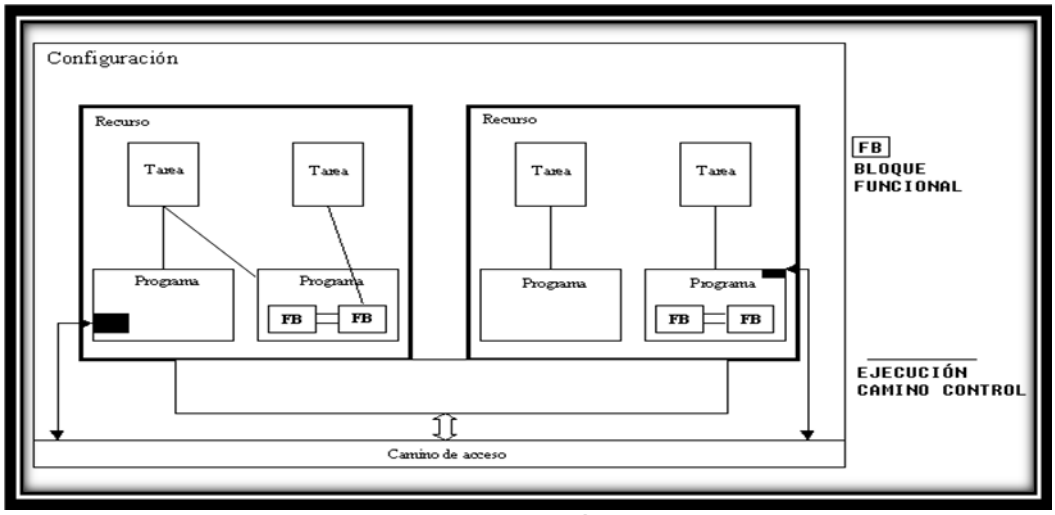


Fig. 2.4 Configuración del IEC 1131-3.

Al más alto nivel, el elemento software requerido para solucionar un problema de control particular puede ser formulado como una configuración. Una configuración es específica para un tipo de sistema de control, incluyendo las características del hardware: procesadores, direccionamiento de la memoria para los canales de I/O y otras capacidades del sistema.

Dentro de una configuración, se pueden definir uno o más recursos. Se puede entender el recurso como un procesador capaz de ejecutar programas IEC.

Con un recurso, pueden estar definidas una o más tareas. Las tareas controlan la ejecución de un conjunto de programas y/o bloques de función. Cada una de ellos puede ser ejecutada periódicamente o por una señal de disparo especificada, como el cambio de estado de una variable.

Los programas están diseñados a partir de un diferente número de elementos de software, escrito en algunos de los distintos lenguajes definidos en IEC 61131-3. Típicamente, un programa es una interacción de Funciones y Bloques Funcionales, con capacidad para intercambiar datos. Funciones y bloques funcionales son las partes básicas de construcción de un programa, que contienen una declaración de datos y variables y un conjunto de instrucciones. Comparado esto con un PLC convencional, éste contiene un solo recurso, ejecutando una tarea que controla un único programa de manera cíclica. IEC 61131-3 incluye la posibilidad de disponer de estructuras más complejas.

El futuro que incluye multi-procesamiento y gestión de programas por eventos ¡Y no está muy lejos!, observar simplemente las características de los sistemas distribuidos o los sistemas de control de tiempo real. IEC 61131-3 está disponible para un amplio rango de aplicaciones, sin tener que conocer otros lenguajes de programación adicionales.

2.5.2. Unidades de Organización de Programa

Dentro de IEC 1131-3, los programas, bloques Funcionales y funciones se denominan Unidades de Organización de Programas, POU´s.

2.5.2.1 Funciones

IEC 61131-3 especifica funciones estándar y funciones definidas por usuario. Las funciones estándar son por ejemplo ADD (suma), ABS (valor absoluto), SQRT (raíz cuadrada), SIN (seno), y COS (coseno). Las funciones definidas por usuario, una vez implementadas pueden ser usadas indefinidamente en cualquier POU.
Las funciones no pueden contener ninguna información de estado interno, es decir, que la invocación de una función con los mismos argumentos (parámetros de entrada) debe suministrar siempre el mismo valor (salida).

2.5.2.2 Bloques Funcionales, FB´s

Los bloques funcionales son los equivalentes de los circuitos integrados, IC´s, que representan funciones de control especializadas. Los FB´s contienen tanto datos como instrucciones, y además pueden guardar los valores de las variables (que es una de las diferencias con las funciones). Tienen un interfaz de entradas y salidas bien definido y un código interno oculto, como un circuito integrado o una caja negra. De este modo, establecen una clara separación entre los diferentes niveles de programadores, o el personal de mantenimiento. Un lazo de control de temperatura, PID, es un excelente ejemplo de bloque funcional. Una vez definido, puede ser usado una y otra vez, en el mismo programa, en diferentes programas o en distintos proyectos. Esto lo hace altamente reutilizable.

Los bloques funcionales pueden ser escritos por el usuario en alguno de los lenguajes de la norma IEC, pero también existen FB´s estándar (biestables, detección de flancos, contadores, temporizadores, etc.). Existe la posibilidad de ser llamados múltiples veces creando copias del bloque funcional que se denominan instancias. Cada instancia llevará asociado un identificador y una estructura de datos que contenga sus variables de salida e internas.

2.5.2.3 Gráfico Funcional Secuencial (SFC (Figura 2.5))

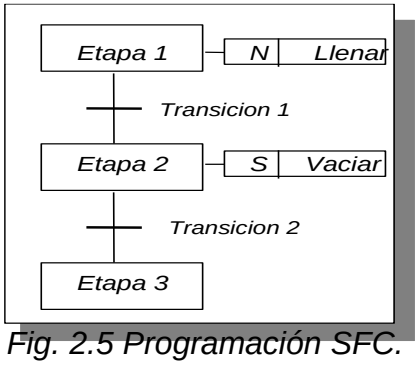


Fig. 2.5 Programación SFC.

SFC describe gráficamente el comportamiento secuencial de un programa de control. Esta definición deriva de las Redes de Petri y Grafcet (IEC 848), con las modificaciones adecuadas para convertir las representaciones de una norma de documentación en un conjunto de elementos de control de ejecución para una POU de un autómata programable.

SFC ayuda a estructurar la organización interna de un programa, y a descomponer un problema en partes manejables, manteniendo simultáneamente una visión global. Los elementos del SFC proporcionan un medio para subdividir una POU de un autómata programable en un conjunto de etapas y transiciones interconectadas por medio de enlaces directos. Cada etapa lleva asociados un conjunto bloques de acción y a cada transición va asociada una condición de transición que cuando se cumple, causa la desactivación de la etapa anterior a la transición y la activación de la siguiente. Los bloques de acción permiten realizar el control del proceso. Cada elemento puede ser programado en alguno de los lenguajes IEC, incluyéndose el propio

SFC. Dado que los elementos del SFC requieren almacenar información, las únicas POU´s que se pueden estructurar utilizando estos elementos son los bloques funcionales y los programas.

Se pueden usar secuencias alternativas y paralelas, comúnmente utilizadas en muchas aplicaciones. Debido a su estructura general, de sencilla comprensión, SFC permite la transmisión de información entre distintas personas con distintos niveles de preparación y responsabilidad dentro de la empresa.

2.6 Lenguajes de Programación

Se definen cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos). Una vez aprendidos se podrá usar una amplia variedad de sistemas basados en esta norma. Los lenguajes consisten en dos tipos literal y gráfico (Figura 2.6):

Literales:
- Lista de instrucciones (IL).
- Texto estructurado (ST).

Gráficos:
- Diagrama de contactos (LD).
- Diagrama de bloques funcionales (FBD).

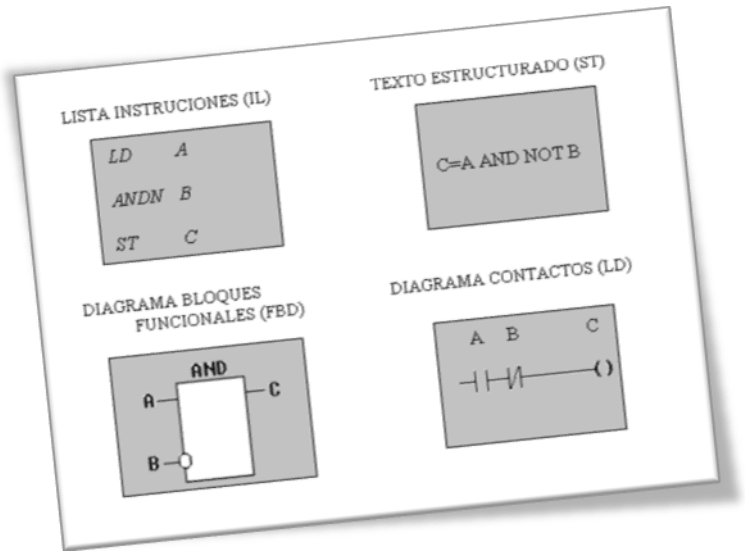


Fig. 2.6 Lenguajes de programación.

En la figura superior, los cuatro programas describen la misma acción. La elección del lenguaje de programación depende de:

- los conocimientos del programador,
- el problema a tratar,
- el nivel de descripción del proceso,
- la estructura del sistema de control,
- la coordinación con otras personas o departamentos.

Los cuatros lenguajes están interrelacionados y permiten su empleo para resolver conjuntamente un problema común según la experiencia del usuario.

El Diagrama de contactos (LD) tiene sus orígenes en los Estados Unidos. Está basado en la presentación gráfica de la lógica de relés. Lista de Instrucciones (IL) es el modelo de lenguaje ensamblador basado un acumulador simple; procede del alemán ´Anweisungliste, AWL.

El Diagramas de Bloques Funcionales (FBD) es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa. El lenguaje Texto estructurado (ST) es un lenguaje de alto nivel con orígenes en el Ada, Pascal y ´C´; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (REPEAT-UNTIL; WHILE-DO), ejecución condicional (IF-THEN-ELSE; CASE), funciones (SQRT, SIN, etc.).

2.7 Top-down vs. Bottom-up

La norma también permite dos formas de desarrollar tu programa de control (ver figura 6): de arriba a abajo (Top-down) y de abajo a arriba (bottom-up).

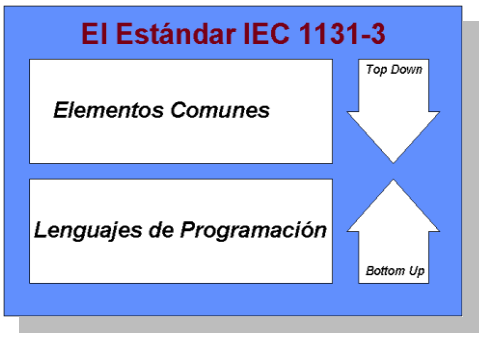


Fig. 2.7 Top-down vs. Bottom-up

Puedes especificar inicialmente la aplicación completa y dividirla en partes, declarar las variables y demás. También puedes comenzar la programación desde abajo, por ejemplo, por medio de funciones y bloque funcionales. Por cualquiera de los caminos que elijas, IEC 61131-3 te ayudará durante todo el proceso.

2.7.1 Implementaciones

Cumplir todos los requerimientos de la norma IEC 61131-3 no es fácil, por eso se permiten implementaciones parciales en varios aspectos. Esto hace referencia al número de lenguajes que soportan las herramientas de desarrollo disponibles, y al número de funciones y de bloques funcionales.

Con ello se deja libertad al suministrador, pero el usuario debe tener cuidado durante el proceso de selección de la herramienta adecuada. Incluso una

actualización del software puede dar lugar a un nivel muy alto de trabajo durante la implementación.

Muchos entornos de programación IEC actuales ofrecen aquello que se espera a nivel de interfase de usuario: uso de ratón, menús desplegables, pantallas de programación gráfica, múltiples ventanas, ayuda en línea, verificación durante el diseño, etc. Debe hacerse notar que estos detalles no están especificados en la norma por lo que es una de las partes donde los proveedores pueden diferenciarse. Las implicaciones técnicas de la norma IEC 61131-3 son altas, dejando bastante espacio para el crecimiento y la diferenciación. Esto la hace adecuada para entrar óptimamente en el próximo siglo.

La norma IEC 61131-3 tendrá un gran impacto en el mundo del control industrial y éste no se restringe al mercado convencional de los PLC´s. Ahora mismo, se pueden ver adoptadas en aplicaciones para control de movimiento, sistemas distribuidos y sistemas de control basados en PC (SoftPLC), incluyendo los paquetes SCADA. Y las áreas de su utilización siguen creciendo. El uso de IEC 61131-3 proporciona numerosos beneficios para usuarios/programadores.

Los beneficios de la adopción de este estándar son varios, dependiendo de las áreas de aplicación: control de procesos, integrador de sistemas, educación, programación, mantenimiento, instalación, etc. Vamos a nombrar sólo algunos de estos beneficios:

1. Se reduce el gasto en recursos humanos, formación, mantenimiento y consultoría.
2. Evita las fuentes habituales de problemas por el alto nivel de flexibilidad y reusabilidad del software.
3. Las técnicas de programación son utilizables en amplios sectores (control industrial en general).
4. Combinan adecuadamente diferentes elementos que pueden provenir de diferentes fabricantes, programas, proyectos.
5. Incrementa la conectividad y comunicación entre los distintos departamentos y compañías.

El estándar IEC 61131-3 es una realidad en papel. Ahora los usuarios que aprecian los beneficios del estándar deben demandar productos que cubran sus necesidades, de modo que las empresas proveedoras puedan amortizar los gastos de desarrollo de las herramientas adecuadas.

2.8 Subestación Eléctrica

Una subestación eléctrica es usada para la transformación de la tensión de la energía eléctrica. El componente principal (y más caro) de una subestación eléctrica es el transformador. Las subestaciones eléctricas elevadoras se ubican en las inmediaciones de las centrales eléctricas para aumentar la tensión de salida de sus generadores. En España los niveles de tensión normalizados más habituales son 15, 20, 66, 132, 220 y 400 kV. De ellos, los dos últimos corresponden a la red de transporte (gestionada y operada por RED ELÉCTRICA) y el resto son de la red de distribución. Cerca de las poblaciones y de los consumidores, se encuentran las subestaciones eléctricas reductoras que reducen el nivel de tensión para que sea apto para su uso por medianos consumidores (fábricas, centros comerciales, hospitales, etc).

Dicha reducción tiene lugar entre tensiones de transporte (400 o 220kV) a tensiones de distribución. Repartidos en el interior de las ciudades existen centros de transformación (CT's) que bajan la tensión a 400V en trifásica (tres fases y neutro), la cual es apropiada para su distribución a pequeños consumidores, entre los que se encuentra el consumo doméstico. Para este tipo de consumo se utiliza en cada vivienda una fase y el neutro, por lo que la tensión que se mide con un polímetro es de 230 V. La razón técnica para realizar esta operación es la conveniencia de transportar la energía eléctrica a larga distancia a tensiones elevadas para reducir las pérdidas resistivas por efecto Joule ($P = I^2 \cdot R$), que dependen de la intensidad de corriente.

Las líneas de la subestación eléctrica están protegidas por equipos principalmente con dos principios de funcionamiento: diferencial de línea y distancia. En el primer caso se compara la intensidad de ambos extremos de la línea en cada instante y se comprueba que coincidan, mientras que en el segundo se obtiene la impedancia de la línea realizando el cociente entre tensión e intensidad para verificar que se encuentre entre unos valores predeterminados. También poseen aparatos de maniobra tanto en carga (interruptores) como sin carga (seccionadores) y de medida (transformadores de intensidad y de tensión). Así mismo es necesario establecer comunicaciones entre las subestaciones que se encuentran en los extremos de las líneas, y ésta puede realizarse bien mediante fibra óptica, comunicaciones en alta frecuencia a través de la misma línea (onda portadora) o por un enlace de radio. Para proteger líneas de media tensión (<66 kV.) frente a caídas de rayos durante tormentas eléctricas y prevenir que se vean afectadas por averías en CT's de clientes se instalan fusibles (comúnmente llamados XS) de manera que éstos sean los elementos que se deterioren en caso de sobreintensidades.

2.8.1 Subestaciones Blindadas

Existe otra alternativa a las Subestaciones Eléctricas convencionales descritas anteriormente. Éstas son las llamadas Subestaciones Eléctricas Blindadas. La Subestación eléctrica blindada más usual es la GIS, Gas Insulated Switchgear. En ellas el fluido que trabaja como aislante es el gas SF6, hexafluoruro de azufre.Éste gas es usado en la mayoría de interruptores de subestaciones

eléctricas convencionales por sus adecuadas características para la eliminación del arco eléctrico. Una Subestación GIS esta construida a base de conductos dentro de los cuales se encuentra el gas SF6.A su vez, dentro de este conductor, además del propio gas, se encuentra el propio elemento conductor eléctrico, que normalmente es un embarrado de cobre bañado en plata.

En la parte GIS de conductos se encuentran los principales elementos de una subestación (a excepción de los transformadores de potencia), tales como los Interruptores, seccionadores, autoválvulas y transformadores de medida. Estos conductos pueden ser instalados outdoor o indoor. Así, es común la instalación de éstos en un edificio (tipo nave industrial) construido al efecto. Por lo tanto, la disposición típica de una subestación GIS es en cuestión la misma que una subestación convencional, salvo que la aparamenta de maniobra y medida (interruptores, seccionadores, transformadores de medida, etc) no está instalada en un parque intemperie, si no que se encuentra en los conductores aislados en SF6. Son numerosos los paises que en la actualidad están instalando éste tipo de Subestación Eléctrica, porque admite un alto grado de tensión de trabajo en un reducido espacio, tienen un mantenimiento muy reducido, y son muy aptas para lugares con ambientes pulvíjenos.

2.8.2 Cuchilla Desconectadoras

Las cuchillas desconectadoras (llamados también Seccionadores) son interruptores de una subestación o circuitos eléctricos que protegen a una subestación de cargas eléctricas demasiado elevadas. Son muy utilizadas en las centrales de transformación de energía eléctrica de cada ciudad; Consta de las siguientes partes:

1. Contacto fijo. Diseñado para trabajo rudo, con recubrimiento de plata.

2. Multicontacto móvil. Localizado en el extremo de las cuchillas, con recubrimiento de plata y muelles de respaldo que proporcionan cuatro puntos de contacto independientes para óptimo comportamiento y presión de contacto.

3. Cámara interruptiva. Asegura la interrupción sin arco externo. Las levas de las cuchillas y de la cámara interruptiva están diseñadas para eliminar cualquier posibilidad de flameo externo.

4. Cuchillas. Fabricadas con doble solera de cobre. La forma de su ensamble proporciona una mayor rigidez y alineación permanente, para asegurar una operación confiable.

5. Contacto de bisagra. Sus botones de contacto troquelado y plateados en la cara interna de las cuchillas, en unión con un gozne plateado giratorio y un resorte de presión de acero inoxidable, conforman un diseño que permite combinar óptimamente la presión de contacto,

evitando puntos calientes pero facilitando la operación y estabilidad de las cuchillas.

6. Aisladores tipo estación. De porcelana, dependiendo del tipo de seccionador varía el número de campanas.

7. Base acanalada. De acero galvanizado de longitud variable, con varios agujeros y ranuras para instalarse en cualquier estructura.

8. Cojinete. De acero, con buje de bronce que proporciona una operación suave. No requiere mantenimiento y resiste la corrosión.

9. Mecanismo de operación. Permite una amplia selección de arreglos de montaje para diferentes estructuras.

La maniobra de operación con estas cuchillas implica abrir antes los interruptores que las cuchillas en el caso de desconexión. Y cerrar antes las cuchillas y después los interruptores en el caso de conexión. Esto es debido a que los seccionadores son un tipo de aparamenta eléctrica más de seguridad, que de corte propiamente dicho, pues su objetivo es proporcionar una seguridad visual de desconexión real ante operaciones que requieren desconexión. De esta forma, un operario trabajando puede ver visualmente que la desconexión se ha llevado a cabo, y que no sufrirá ninguna clase de daños, aunque exista un fallo en los interruptores, y que las cuchillas pueden tener peligro de arco eléctrico mientras que los interruptores, no.

2.8.3 Interruptor

Un interruptor es un dispositivo para cambiar el curso de un circuito. El modelo prototípico es un dispositivo mecánico (por ejemplo un interruptor de ferrocarril) que puede ser desconectado de un curso y unido (conectado) al otro. El término "el interruptor" se refiere típicamente a la electricidad o a circuitos electrónicos. En usos donde requieren múltiples opciones de conmutación (p.ej., un teléfono), con el tiempo han sido remplazados por las variantes electrónicas que pueden ser controladas y automatizadas. Un disyuntor o interruptor automático magneto-térmico es un aparato capaz de interrumpir o abrir un circuito eléctrico cuando la intensidad de la corriente eléctrica que por él circula excede de un determinado valor o, en el que se ha producido un cortocircuito, con el objetivo de no causar daños a los equipos eléctricos.

A diferencia de los fusibles, que deben ser reemplazados tras un único uso, el disyuntor puede ser rearmado una vez localizado y reparado el daño que causó el disparo o desactivación automática. Se fabrican disyuntores de diferentes tamaños y características lo cual hace que sea ampliamente utilizado en viviendas, industrias y comercios.

2.8.4 Protección Sistema De Potencia

Las Protecciones de los sistemas de potencia son una parte integral de estos, tienen como tarea evitar la destrucción de un conjunto de equipos o dispositivos interconectados en una tarea común por causa de una falla que podría iniciarse de manera simple y después extenderse sin control en forma encadenada. El sistema de protecciones debe aislar la parte donde se ha producido la falla buscando perturbar lo menos posible la red.

2.8.4.1 Protecciones de líneas de transmisión

La Línea de Transmisión (LT) es el elemento del sistema eléctrico de potencia destinado a transportar la energía, desde su generación hasta el punto de distribución para su consumo, por lo que se considera como el elemento más importante en el suministro de energía eléctrica. Y forma parte de la Red de transporte de energía eléctrica. El esquema de protección de una LT está formado por una protección primaria y protecciones de respaldo, siendo la primaria de alta velocidad y las de respaldo con acción retardada.

El objeto de la característica de alta velocidad de la protección primaria es debido a que ésta debe actuar en la menor cantidad de tiempo posible tratando de aislar la falla del sistema, las de respaldo son de acción retardada, ya que tienen que esperar a que la protección primaria actúe, si no es así lo harán éstas otras. Esto no significa que las de respaldo solo actuarán en caso de que la primaria no actúe.
La gran desventaja es que la protección de respaldo aisla una sección de mayor dimensión que la primaria. Existen varios factores que afectan el diseño y operación de un SP en Líneas de Transmisión, los cuales son: configuración de la red y niveles de tensión, entre otros. Los esquemas de protección que se pueden utilizar en una LT, son: Protección contra sobre corriente (PSC), Protección de distancia (PD), Protección de hilo piloto (PHP), y la protección híbrida (PH). Las protecciones que se aplican a las líneas de transmisión se dividen en dos grupos principales, el de protecciones primarias y el de protecciones de respaldo como se describen a continuación:

1. Primaria
    a. Diferencial con hilo piloto
    b. Comparación de fase con onda portadora (carrier), o hilo piloto con tonos de audio.
    c. Comparación direccional con relevadores de distancia y onda portadora, o hilo piloto con tonos de audio.

2. Respaldo
    a. Distancia
    b. Sobre corriente direccional de fases y tierra

2.9 Esquema Unifilar

Un esquema unifilar es una representación gráfica de una instalación eléctrica o de parte de ella. El esquema unifilar se distingue de otros tipos de esquemas eléctricos en que el conjunto de conductores de un circuito se representa mediante una única línea, independientemente de la cantidad de dichos conductores. Típicamente el esquema unifilar tiene una estructura de árbol.

2.9.1 Elementos típicos en un esquema unifilar

La siguiente es una relación no exhaustiva de elementos gráficos que se suelen encontrar en un esquema unifilar.

2.9.2 Cuadros eléctricos

Todos los componentes que se encuentran en el interior de un mismo cuadro eléctrico se representan en el interior de un polígono (probablemente un rectángulo). Este polígono representa al cuadro eléctrico y se suele dibujar con una línea discontínua. Además, es conveniente que una etiqueta identifique a qué cuadro hace referencia cada polígono.

2.9.3 Circuito

Un circuito es una rama del esquema unifilar con dos extremos. El extremo superior puede ser el inicio del esquema unifilar o estar conectadas a otro circuito aguas arriba. El extremo inferior puede estar conectado a uno o más circuitos aguas abajo, o a un receptor.

2.9.4 Número y características de los conductores

El número de conductores de un circuito se representa mediante unos trazos oblicuos, y paralelos entre sí, que se dibujan sobre la línea. Sólamente se representan los conductores activos (no el de tierra), por lo que es habitual encontrar dos, tres o cuatro trazos, para circuitos monofásicos, trifásicos sin neutro y trifásicos con neutro, respectivamente. Junto a cada rama se indican las características del conductor, como número de conductores, sección, material, aislamiento, canalización, etc.

2.9.5 Aparamenta de protección o maniobra

En algunas ramas del esquema unifilar es posible encontrar aparamenta de protección o de maniobra como, por ejemplo, interruptores diferenciales, magnetotérmicos o relés.

2.9.6 Receptores

Las ramas inferiores del esquema unifilar alimentan a receptores eléctricos, tales como lámparas, tomas de corriente, motores, etc. Cada grupo de receptores iguales en un mismo circuito se representa mediante un único símbolo. Debajo del símbolo del receptor se indican algunos datos de interés, como la designación del receptor, la cantidad, la potencia de cálculo de la línea, la longitud máxima o la caída de tensión en el punto más alejado de la línea.

2.10 Compuertas Lógicas

Las computadoras digitales utilizan el sistema de números binarios, que tiene dos dígitos 0 y 1. Un dígito binario se denomina un bit. La información está representada en las computadoras digitales en grupos de bits. Utilizando diversas técnicas de codificación los grupos de bits pueden hacerse que representen no solamente números binarios sino también otros símbolos discretos cualesquiera, tales como dígitos decimales o letras de alfabeto. Utilizando arreglos binarios y diversas técnicas de codificación, los dígitos binarios o grupos de bits pueden utilizarse para desarrollar conjuntos completos de instrucciones para realizar diversos tipos de cálculos.

La información binaria se representa en un sistema digital por cantidades físicas denominadas señales, Las señales eléctricas tales como voltajes existen a través del sistema digital en cualquiera de dos valores reconocibles y representan una variable binaria igual a 1 o 0. Por ejemplo, un sistema digital particular puede emplear una señal de 3 volts para representar el binario "1" y 0.5 volts para el binario "0".

La siguiente ilustración muestra un ejemplo de una señal binaria (figura 2.8).

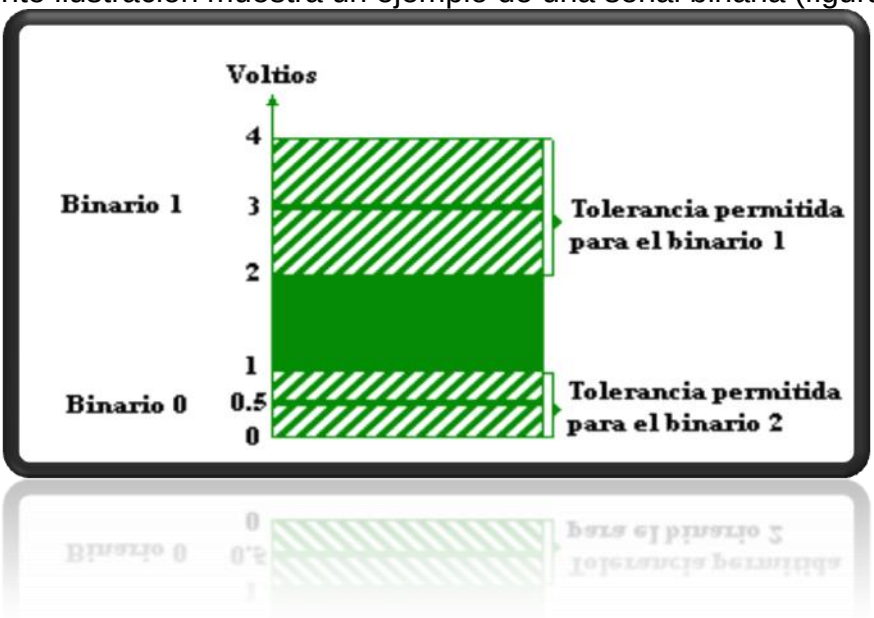


Fig. 2.8 Tolerancias permitidas para los binarios.

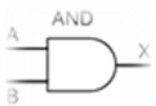
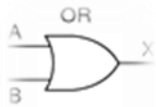
Como se muestra en la figura, cada valor binario tiene una desviación aceptable del valor nominal. La región intermedia entre las dos regiones permitidas se cruza solamente durante la transición de estado.

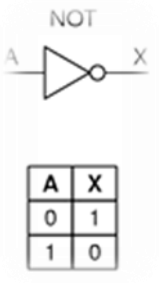
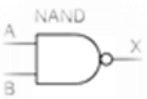
Los terminales de entrada de un circuito digital aceptan señales binarias dentro de las tolerancias permitidas y los circuitos responden en los terminales de salida con señales binarias que caen dentro de las tolerancias permitidas. La lógica binaria tiene que ver con variables binarias y con operaciones que toman un sentido lógico. La manipulación de información binaria se hace por circuitos lógicos que se denominan Compuertas.

Las compuertas son bloques del hardware que producen señales en binario 1 ó 0 cuando se satisfacen los requisitos de entrada lógica. Las diversas compuertas lógicas se encuentran comúnmente en sistemas de computadoras digitales. Cada compuerta tiene un símbolo gráfico diferente y su operación puede describirse por medio de una función algebraica. Las relaciones entrada - salida de las variables binarias para cada compuerta pueden representarse en forma tabular en una tabla de verdad.

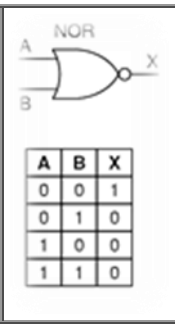
A continuación se detallan los nombres, símbolos, gráficos, funciones algebraicas, y tablas de verdad de las compuertas más usadas (Tabla 2.1).

Tabla 2.1 Compuertas Lógicas

Compuerta AND: Cada compuerta tiene dos variables de entrada designadas por A y B y una salida binaria designada por x. La compuerta AND produce la multiplicación lógica AND: esto es: la salida es 1 si la entrada A y la entrada B están ambas en el binario 1: de otra manera, la salida es 0. Estas condiciones también son especificadas en la tabla de verdad para la compuerta AND. La tabla muestra que la salida x es 1 solamente cuando ambas entradas A y B están en 1. El símbolo de operación algebraico de la función AND es el mismo que el símbolo de la multiplicación de la aritmética ordinaria (*). Las compuertas AND pueden tener más de dos entradas y por definición, la salida es 1 si todas las entradas son 1.	 A B X 0 0 0 0 1 0 1 0 0 1 1 1
Compuerta OR: La compuerta OR produce la función sumadora, esto es, la salida es 1 si la entrada A o la entrada B o ambas entradas son 1; de otra manera, la salida es 0. El símbolo algebraico de la función OR (+), es igual a la operación de aritmética de suma. Las compuertas OR pueden tener más de dos entradas y por definición la salida es 1 si cualquier entrada es 1.	 A B X 0 0 0 0 1 1 1 0 1 1 1 1

Compuerta NOT: El circuito NOT es un inversor que invierte el nivel lógico de una señal binaria. Produce el NOT, o función complementaria. El símbolo algebraico utilizado para el complemento es una barra sobra el símbolo de la variable binaria. Si la variable binaria posee un valor 0, la compuerta NOT cambia su estado al valor 1 y viceversa. El círculo pequeño en la salida de un símbolo gráfico de un inversor designa un inversor lógico. Es decir cambia los valores binarios 1 a 0 y viceversa.	 A X 0 1 1 0
Compuerta Separador (yes): Un símbolo triángulo por sí mismo designa un circuito separador, el cual no produce ninguna función lógica particular puesto que el valor binario de la salida es el mismo de la entrada. Este circuito se utiliza simplemente para amplificación de la señal. Por ejemplo, un separador que utiliza 5 volt para el binario 1, producirá una salida de 5 volt cuando la entrada es 5 volt. Sin embargo, la corriente producida a la salida es muy superior a la corriente suministrada a la entrada de la misma. De ésta manera, un separador puede excitar muchas otras compuertas que requieren una cantidad mayor de corriente que de otra manera no se encontraría en la pequeña cantidad de corriente aplicada a la entrada del separador.	 A X 0 0 1 1
Compuerta NAND: Es el complemento de la función AND, como se indica por el símbolo gráfico, que consiste en una compuerta AND seguida por un pequeño círculo (quiere decir que invierte la señal). La designación NAND se deriva de la abreviación NOT - AND. Una designación más adecuada habría sido AND invertido puesto que es la función AND la que se ha invertido. Las compuertas NAND pueden tener más de dos entradas, y la salida es siempre el complemento de la función AND.	 A B X 0 0 1 0 1 1 1 0 1 1 1 0

Compuerta NOR:
La compuerta NOR es el complemento de la compuerta OR y utiliza el símbolo de la compuerta OR seguido de un círculo pequeño (quiere decir que invierte la señal). Las compuertas NOR pueden tener más de dos entradas, y la salida es siempre el complemento de la función OR.



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

2.11 Diagrama De Tiempo

Un diagrama de tiempos o cronograma es una gráfica de formas de onda digitales que muestra la relación temporal entre varias señales, y cómo varía cada señal en relación a las demás. Un cronograma puede contener cualquier número de señales relacionadas entre sí. Examinando un diagrama de tiempos, se puede determinar los estados, nivel alto o nivel bajo, de cada una de las señales en cualquier instante de tiempo especificado, y el instante exacto en que cualquiera de las señales cambia de estado con respecto a las restantes.

El propósito primario del diagrama de tiempos es mostrar los cambios en el estado o la condición de una línea de vida (representando una Instancia de un Clasificador o un Rol de un clasificador) a lo largo del tiempo lineal. El uso más común es mostrar el cambio de estado de un objeto a lo largo del tiempo, en respuesta a los eventos o estímulos aceptados.

Los eventos que se reciben se anotan, a medida que muestran cuándo se desea mostrar el evento que causa el cambio en la condición o en el estado (figura 2.9).

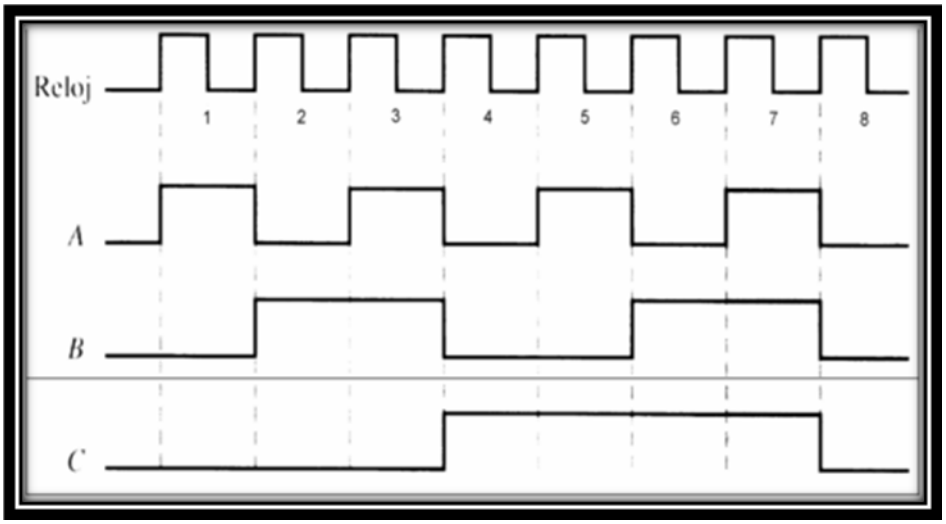


Fig. 2.9 Diagrama de tiempo.

2.11.1 Diagramas de Tiempos UML

En el estándar de Lenguaje de Modelado Unificado de OMG los diagramas de tiempo son una representación especial de interacción que se enfoca en el tiempo de los mensajes enviados entre objetos. Se pueden usar estos diagramas para mostrar restricciones detalladas sobre el tiempo, ó para mostrar los cambios con líneas de vida respecto al tiempo. Los diagramas de tiempo son generalmente utilizados con sistemas en tiempo real o en sistemas embebidos.

2.11.2 Señal Digital

La señal digital es un tipo de señal generada por algún tipo de fenómeno electromagnético en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representan valores discretos, en lugar de valores dentro de un cierto rango. Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada (véase circuito de conmutación).

Los sistemas digitales, como por ejemplo el ordenador, usan lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto, H y otro bajo, L (de High y Low, respectivamente, en inglés). Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria. Si el nivel alto se representa por 1 y el bajo por 0, se habla de lógica positiva y en caso contrario de lógica negativa.

Cabe mencionar que, además de los niveles, en una señal digital están las transiciones de alto a bajo y de bajo a alto, denominadas flanco de subida y de bajada, respectivamente.

En la figura se muestra una señal digital donde se identifican los niveles y los flancos (figura 2.10).

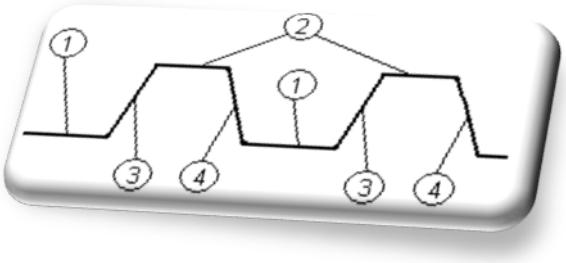


Fig. 2.10 Señal digital.

Señal digital con ruido : 1) Nivel bajo, 2) Nivel alto, 3) Flanco de subida y 4) Flanco de bajada (figura 2.11).

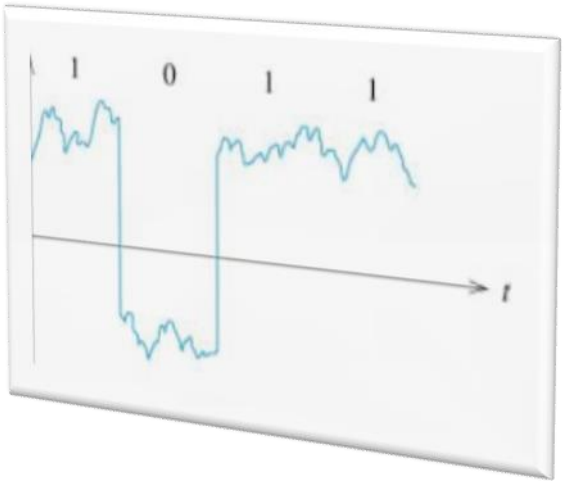


Fig. 2.11 Flanco de bajada.

Es conveniente aclarar que, a pesar de que en los ejemplos señalados el término digital se ha relacionado siempre con dispositivos binarios, no significa que digital y binario sean términos intercambiables. Por ejemplo, si nos fijamos en el código Morse, veremos que en él se utilizan, para el envío de mensajes por telégrafo eléctrico, cinco estados digitales, que son:

> Punto, raya, espacio corto (entre letras), espacio medio (entre palabras) y espacio largo (entre frases)

Referido a un aparato o instrumento de medida, decimos que es digital cuando el resultado de la medida se representa en un visualizador mediante números (dígitos) en lugar de hacerlo mediante la posición de una aguja, o cualquier otro indicador, en una escala.

2.11.3 Ventajas de las señales digitales

1. Ante la atenuación, puede ser amplificada y reconstruida al mismo tiempo, gracias a los sistemas de regeneración de señales.

2. Cuenta con sistemas de detección y corrección de errores, en la recepción.

3. Facilidad para el procesamiento de la señal. Cualquier operación es fácilmente realizable a través de cualquier software de edición o procesamiento de señal.

4. Permite la generación infinita sin pérdidas de calidad. Esta ventaja sólo es aplicable a los formatos de disco óptico; la cinta magnética digital, aunque en menor medida que la analógica (que sólo soporta como mucho 4 o 5 generaciones), también va perdiendo información con la multi-generación.

5. Las señales digitales se ven menos afectadas a causa del ruido ambiental en comparación con las señales analógicas.

2.11.4 Inconvenientes de las señales digitales

1. Necesita una conversión analógica-digital previa y una decodificación posterior en el momento de la recepción.
2. Requiere una sincronización precisa entre los tiempos del reloj del transmisor con respecto a los del receptor.
3. La señal digital requiere mayor ancho de banda que la señal analógica para ser transmitida.
4. En la transformación de una señal analógica a una digital siempre existirá un margen de error ya que una señal analógica continua tiene valores infinitos, y una señal digital actualmente su número de valores es finito.

2.12 Controlador Lógico Programable (PLC.- Programmable Logic Controller)

Los PLC (Programmable Logic Controller en sus siglas en inglés) son dispositivos electrónicos muy usados en Automatización Industrial. PLC = Es un hardware industrial, que se utiliza para la obtención de datos. Una vez obtenidos, los pasa a través de bus (por ejemplo por ethernet) en un servidor. Su historia se remonta a finales de la década de 1960 cuando la industria buscó en las nuevas tecnologías electrónicas una solución más eficiente para reemplazar los sistemas de control basados en circuitos eléctricos con relés, interruptores y otros componentes comúnmente utilizados para el control de los sistemas de lógica combinacional. Hoy en día, los PLC no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como controladores proporcional integral derivativo (PID). Los PLC actuales pueden comunicarse con otros controladores y computadoras en redes de área local, y son una parte fundamental de los modernos sistemas de control distribuido. Existen varios lenguajes de programación, tradicionalmente los más utilizados son el diagrama de escalera (Lenguaje Ladder), preferido por los electricistas, lista de instrucciones y programación por estados, aunque se han incorporado lenguajes más intuitivos que permiten implementar algoritmos complejos mediante simples diagramas de flujo más fáciles de interpretar y mantener.

Un lenguaje más reciente, preferido por los informáticos y electrónicos, es el FBD (en inglés Function Block Diagram) que emplea compuertas lógicas y bloques con distintas funciones conectados entre sí.

En la programación se pueden incluir diferentes tipos de operandos, desde los más simples como lógica booleana, contadores, temporizadores, contactos, bobinas y operadores matemáticos, hasta operaciones más complejas como manejo de tablas (recetas), apuntadores, algoritmos PID y funciones de comunicación multiprotocolos que le permitirían interconectarse con otros dispositivos.

2.12.1 PLC en comparación con otros sistemas de control

Los PLC están adaptados para un amplio rango de tareas de automatización. Estos son típicos en procesos industriales en la manufactura donde el costo de desarrollo y mantenimiento de un sistema de automatización es relativamente alto contra el coste de la automatización, y donde van a existir cambios en el sistema durante toda su vida operacional. Los PLC contienen todo lo necesario para manejar altas cargas de potencia; se requiere poco diseño eléctrico y el problema de diseño se centra en expresar las operaciones y secuencias en la lógica de escalera (o diagramas de funciones). Las aplicaciones de PLC son normalmente hechos a la medida del sistema, por lo que el costo del PLC es bajo comparado con el costo de la contratación del diseñador para un diseño específico que solo se va a usar una sola vez. Por otro lado, en caso de productos de alta producción, los sistemas de control a medida se amortizan por sí solos rápidamente debido al ahorro en los componentes, lo que provoca que pueda ser una buena elección en vez de una solución "genérica". Sin embargo, debe ser notado que algunos PLC ya no tienen un precio alto. Los PLC actuales tienen todas las capacidades por algunos cientos de dólares. Diferentes técnicas son utilizadas para un alto volumen o una simple tarea de automatización, Por ejemplo, una lavadora de uso doméstico puede ser controlada por un temporizador a levas electromecánico costando algunos cuantos dólares en cantidades de producción. Un diseño basado en un microcontrolador puede ser apropiado donde cientos o miles de unidades deben ser producidas y entonces el coste de desarrollo (diseño de fuentes de alimentación y equipo de entradas y salidas) puede ser dividido en muchas ventas, donde el usuario final no tiene necesidad de alterar el control.

Aplicaciones automotrices son un ejemplo, millones de unidades son vendidas cada año, y pocos usuarios finales alteran la programación de estos controladores. (Sin embargo, algunos vehículos especiales como son camiones de pasajeros para tránsito urbano utilizan PLC en vez de controladores de diseño propio, debido a que los volúmenes son pequeños y el desarrollo no sería económico.) Algunos procesos de control complejos, como los que son utilizados en la industria química, pueden requerir algoritmos y características más allá de la capacidad de PLC de alto nivel. Controladores de alta velocidad también requieren de soluciones a medida; por ejemplo, controles para aviones.

Los PLC pueden incluir lógica para implementar bucles analógicos, “proporcional, integral y derivadas” o un controlador PID. Un bucle PID podría ser usado para controlar la temperatura de procesos de fabricación, por ejemplo. Históricamente, los PLC’s fueron configurados generalmente con solo unos pocos bucles de control analógico y en donde los procesos requieren cientos o miles de bucles, un Sistema de Control Distribuido (DCS) se encarga. Sin embargo, los PLC se han vuelto más poderosos, y las diferencias entre las aplicaciones entre DCS y PLC han quedado menos claras.

Resumiendo, los campos de aplicación de un PLC o autómata programable en procesos industriales son: cuando hay un espacio reducido, cuando los procesos de producción son cambiantes periódicamente, cuando hay procesos

secuenciales, cuando la maquinaria de procesos es variable, cuando las instalaciones son de procesos complejos y amplios, cuando el chequeo de programación se centraliza en partes del proceso. Sus aplicaciones generales son las siguientes: maniobra de máquinas, maniobra de instalaciones y señalización y control.

2.12.2 Señales Analógicas y digitales

Las señales digitales o discretas como los interruptores, son simplemente una señal de On/Off (1 ó 0, Verdadero o Falso, respectivamente). Los botones e interruptores son ejemplos de dispositivos que proporcionan una señal discreta. Las señales discretas son enviadas usando la tensión o la intensidad, donde un rango especifico corresponderá al On y otro rango al Off. Un PLC puede utilizar 24V de voltaje continuo en la E/S donde valores superiores a 22V representan un On, y valores inferiores a 2V representan Off. Inicialmente los PLC solo tenían E/S discretas.

Las señales analógicas son como controles de volúmenes, con un rango de valores entre 0 y el tope de escala. Esto es normalmente interpretado con valores enteros por el PLC, con varios rangos de precisión dependiendo del dispositivo o del número de bits disponibles para almacenar los datos. Presión, temperatura, flujo, y peso son normalmente representados por señales analógicas. Las señales analógicas pueden usar tensión o intensidad con una magnitud proporcional al valor de la señal que procesamos. Por ejemplo, una entrada de 4-20 mA o 0-10 V será convertida en enteros comprendidos entre 0-32767.

Las entradas de intensidad son menos sensibles al ruido eléctrico (como por ejemplo el arranque de un motor eléctrico) que las entradas de tensión. Ejemplo: Como ejemplo, las necesidades de una instalación que almacena agua en un tanque. El agua llega al tanque desde otro sistema, y como necesidad a nuestro ejemplo, el sistema debe controlar el nivel del agua del tanque. Usando solo señales digitales, el PLC tiene 2 entradas digitales de dos interruptores del tanque (tanque lleno o tanque vacío). El PLC usa la salida digital para abrir o cerrar una válvula que controla el llenado del tanque.

Si los dos interruptores están apagados o solo el de “tanque vacío” esta encendido, el PLC abrirá la válvula para dejar entrar agua. Si solo el de “tanque lleno” esta encendido, la válvula se cerrara. Si ambos interruptores están encendidos sería una señal de que algo va mal con uno de los dos interruptores, porque el tanque no puede estar lleno y vacío a la vez. El uso de dos interruptores previene situaciones de pánico donde cualquier uso del agua activa la bomba durante un pequeño espacio de tiempo causando que el sistema se desgaste más rápidamente. Así también se evita poner otro PLC para controlar el nivel medio del agua. Un sistema analógico podría usar una báscula que pese el tanque, y una válvula ajustable. El PLC podría usar un PID para controlar la apertura de la válvula. La báscula esta conectada a una entrada analógica y la válvula a una salida analógica. El sistema llena el tanque rápidamente cuando hay poca agua en el tanque. Si el nivel del agua baja rápidamente, la válvula se abrirá todo lo que se pueda, si el caso es que

el nivel del agua esta cerca del tope máximo, la válvula estará poco abierta para que entre el agua lentamente y no se pase de este nivel.

Con este diseño del sistema, la válvula puede desgastarse muy rápidamente, por eso, los técnicos ajustan unos valores que permiten que la válvula solo se abra en unos determinados valores y reduzca su uso. Un sistema real podría combinar ambos diseños, usando entradas digitales para controlar el vaciado y llenado total del tanque y el censor de peso para optimizarlos.

2.12.3 Capacidades E/S en los PLC modulares

Los PLC modulares tienen un limitado número de conexiones para la entrada y la salida. Normalmente, hay disponibles ampliaciones si el modelo base no tiene suficientes puertos E/S. Los PLC con forma de rack tienen módulos con procesadores y con módulos de E/S separados y opcionales, que pueden llegar a ocupar varios racks. A menudo hay miles de entradas y salidas, tanto analógicas como digitales. A veces, se usa un puerto serie especial de E/S que se usa para que algunos racks puedan estar colocados a larga distancia del procesador, reduciendo el coste de cables en grandes empresas. Alguno de los PLC actuales pueden comunicarse mediante un amplio tipo de comunicaciones incluidas RS-485, coaxial, e incluso Ethernet para el control de las entradas salidas con redes a velocidades de 100 Mbps.

Los PLC usados en grandes sistemas de E/S tienen comunicaciones P2P entre los procesadores. Esto permite separar partes de un proceso complejo para tener controles individuales mientras se permita a los subsistemas comunicarse mediante links. Estos links son usados a menudo por dispositivos de Interfaz de usuario (HMI) como keypads o estaciones de trabajo basados en ordenadores personales. El número medio de entradas de un PLC es 3 veces el de salidas, tanto en analógico como en digital. Las entradas “extra” vienen de la necesidad de tener métodos redundantes para controlar apropiadamente los dispositivos, y de necesitar siempre mas controles de entrada para satisfacer la realimentación de los dispositivos conectados.

2.12.4 Programación

Los primeros PLC, en la primera mitad de los 80, eran programados usando sistemas de programación propietarios o terminales de programación especializados, que a menudo tenían teclas de funciones dedicadas que representaban los elementos lógicos de los programas de PLC. Los programas eran guardados en cintas. Más recientemente, los programas PLC son escritos en aplicaciones especiales en un ordenador, y luego son descargados directamente mediante un cable o una red al PLC. Los PLC viejos usan una memoria no volátil (magnetic core memory) pero ahora los programas son guardados en una RAM con batería propia o en otros sistemas de memoria no volátil como las memoria flash. Los primeros PLC fueron diseñados para ser usados por electricistas que podían aprender a programar los PLC en el trabajo. Estos PLC eran programados con “lógica de escalera”("ladder logic").

Los PLC modernos pueden ser programados de muchas formas, desde la lógica de escalera hasta lenguajes de programación tradicionales como el BASIC o C. Otro método es usar la Lógica de Estados (State Logic), un lenguaje de programación de alto nivel diseñado para programas PLC basándose en los diagramas de transición de estados.

Recientemente, el estándar internacional IEC 61131-3 se está volviendo muy popular. IEC 61131-3 define cinco lenguajes de programación para los sistemas de control programables: FBD (Function block diagram), LD (Ladder diagram), ST (Structured text, similar al Lenguaje de programación Pascal), IL (Instruction list) y SFC (Sequential function chart). Mientras que los conceptos fundamentales de la programación del PLC son comunes a todos los fabricantes, las diferencias en el direccionamiento E/S, la organización de la memoria y el conjunto de instrucciones hace que los programas de los PLC nunca se puedan usar entre diversos fabricantes. Incluso dentro de la misma línea de productos de un solo fabricante, diversos modelos pueden no ser directamente compatibles.

La estructura básica de cualquier autómata programable es:

1. Fuente de alimentación: convierte la tensión de la red, 110 ó 220V ac a baja tensión de cc (24V por ejemplo) que es la que se utiliza como tensión de trabajo en los circuitos electrónicos que forma el autómata.

2. CPU: la Unidad Central de Procesos es el auténtico cerebro del sistema. Es el encargado de recibir órdenes del operario a través de la consola de programación y el módulo de entradas. Después las procesa para enviar respuestas al módulo de salidas.

3. Módulo de entradas: aquí se unen eléctricamente los captadores (interruptores, finales de carrera...). La información que recibe la envía al CPU para ser procesada según la programación. Hay 2 tipos de captadores conectables al módulo de entradas: los pasivos y los activos.

4. Módulo de salida: es el encargado de activar y desactivar los actuadores (bobinas de contactores, motores pequeños...). La información enviada por las entradas a la CPU, cuando esta procesada se envía al módulo de salidas para que estas sean activadas (también los actuadores que están conectados a ellas). Hay 3 módulos de salidas según el proceso a controlar por el autómata: relés, triac y transistores.

5. Terminal de programación: la terminal o consola de programación es el que permite comunicar al operario con el sistema. Sus funciones son la transferencia y modificación de programas, la verificación de la programación y la información del funcionamiento de los procesos.

6. Periféricos: ellos no intervienen directamente en el funcionamiento del autómata pero si que facilitan la labor del operario.

2.12.5 Comunicaciones

Las formas como los PLC intercambian datos con otros dispositivos son muy variadas. Típicamente un PLC puede tener integrado puertos de comunicaciones seriales que pueden cumplir con distintos estándares de acuerdo al fabricante. Estos puertos pueden ser de los siguientes tipos:

a. RS-232
b. RS-485
c. RS-422
d. Ethernet

Sobre estos tipos de puertos de hardware las comunicaciones se establecen utilizando algún tipo de protocolo o lenguaje de comunicaciones. En esencia un protocolo de comunicaciones define la manera como los datos son empaquetados para su transmisión y como son codificados. De estos protocolos los más conocidos son:

a. Modbus
b. Bus CAN
c. Profibus
d. Devicenet
e. Controlnet
f. Ethernet I/P

Muchos fabricantes además ofrecen distintas maneras de comunicar sus PLC con el mundo exterior mediante esquemas de hardware y software protegidos por patentes y leyes de derecho de autor.

2.13 SCADA

Es una aplicación de software especialmente diseñada para funcionar sobre ordenadores (computadores) en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos) y controlando el proceso de forma automática desde la pantalla del ordenador.

También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro de la empresa (supervisión, control calidad, control de producción, almacenamiento de datos, etc.).

Comprende todas aquellas soluciones de aplicación para referirse a la captura de información de un proceso o planta, no necesariamente industrial, para que, con esta información, sea posible realizar una serie de análisis o estudios con los que se pueden obtener valiosos indicadores

Que permitan una retroalimentación sobre un operador o sobre el propio proceso, tales como:

- Indicadores sin retroalimentación inherente (no afectan al proceso, sólo al operador):
    - Estado actual del proceso. Valores instantáneos;
    - Desviación o deriva del proceso. Evolución histórica y acumulada;
- Indicadores con retroalimentación inherente (afectan al proceso, después al operador):
    - Generación de alarmas;
    - HMI Human Machine Interface (Interfaces hombre-máquina);
    - Toma de decisiones:
        - Mediante operatoria humana;
        - Automática (mediante la utilización de sistemas basados en el conocimiento o sistemas expertos).

2.13.1 Esquema De Un Sistema Típico

Este esquema es un ejemplo de la aplicación del sistema SCADA en áreas industriales (figura 2.12).

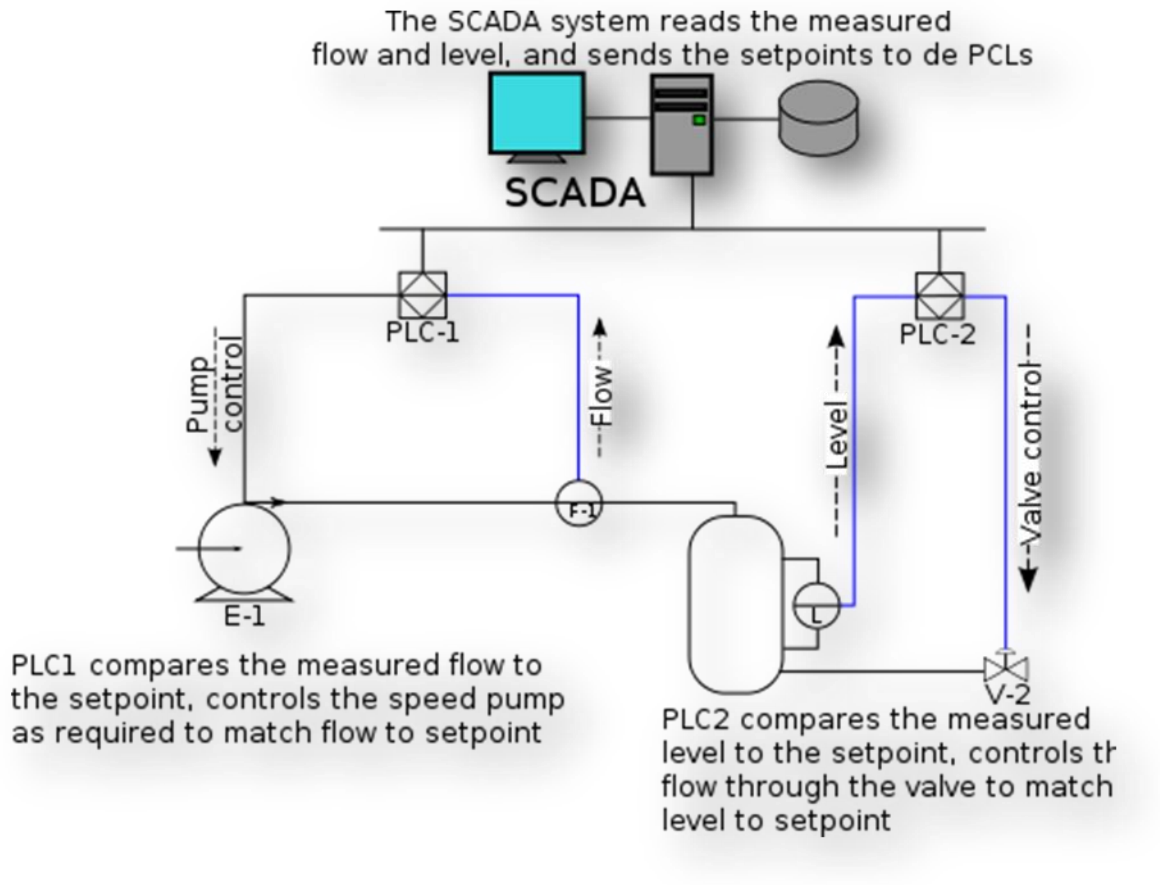


Fig 2.11 Sistema SCADA.

Estas áreas pueden ser:

- Monitorear procesos químicos, físicos o de transporte en sistemas de suministro de agua, para controlar la generación y distribución de energía eléctrica, de gas o en oleoductos y otros procesos de distribución.
- Gestión de la producción (facilita la programación de la fabricación).
- Mantenimiento (proporciona magnitudes de interés tales para evaluar y determinar modos de fallo, MTBF, índices de Fiabilidad, entre otros).
- Control de Calidad (proporciona de manera automatizada los datos necesarios para calcular índices de estabilidad de la producción CP y CPk, tolerancias, índice de piezas NOK/OK, etc.
- Administración (actualmente pueden enlazarse estos datos del SCADA con un servidor ERP (Enterprise Resource Planning o sistema de planificación de recursos empresariales), e integrarse como un módulo más).
- Tratamiento histórico de información (mediante su incorporación en bases de datos).

2.13.2 Definiciones del Sistema

Supervisión: acto de observar el trabajo o tareas de otro (individuo o máquina) que puede no conocer el tema en profundidad, supervisar no significa el control sobre el otro, sino el guiarlo en un contexto de trabajo, profesional o personal, es decir con fines correctivos y/o de modificación.

Automática: ciencia tecnológica que busca la incorporación de elementos de ejecución autónoma que emulan el comportamiento humano o incluso superior.

Principales familias: autómatas, robots, controles de movimiento, adquisición de datos, visión artificial, etc.

PLC: Programmable Logic Controller, Controlador Lógico Programable.
PAC: Programmable Automation Controller, Controlador de Automatización Programable.

Un sistema SCADA incluye un hardware de señal de entrada y salida, controladores, interfaz hombre-máquina (HMI), redes, comunicaciones, base de datos y software.

El término SCADA usualmente se refiere a un sistema central que monitorea y controla un sitio completo o una parte de un sitio que nos interesa controlar (el control puede ser sobre máquinas en general, depósitos, bombas, etc.) o finalmente un sistema que se extiende sobre una gran distancia (kilómetros / millas). La mayor parte del control del sitio es en realidad realizada automáticamente por una Unidad Terminal Remota (UTR), por un Controlador Lógico Programable (PLC) y más actualmente por un Controlador Automático Programable (PAC). Las funciones de control del servidor están casi siempre restringidas a reajustes básicos del sitio o capacidades de nivel de supervisión.

Por ejemplo un PLC puede controlar el flujo de agua fría a través de un proceso, pero un sistema SCADA puede permitirle a un operador cambiar el punto de consigna (set point) de control para el flujo, y permitirá grabar y mostrar cualquier condición de alarma como la pérdida de un flujo o una alta temperatura. La realimentación del lazo de control es cerrada a través del RTU o el PLC; el sistema SCADA monitorea el desempeño general de dicho lazo. El sistema SCADA también puede mostrar gráficas con históricos, tablas con alarmas y eventos, permisos y accesos de los usuarios. Necesidades de la supervisión de procesos:

- Limitaciones de la visualización de los sistemas de adquisición y control.

- Control software. Cierre de lazo del control.

- Recoger, almacenar y visualizar la información.

2.13.3 Interfaz Humano – Máquina

Una interfaz Hombre - Máquina o HMI ("Human Machine Interface") es el aparato que presenta los datos a un operador (humano) y a través del cual éste controla el proceso. Los sistemas HMI podemos pensarlos como una "ventana de un proceso". Esta ventana puede estar en dispositivos especiales como paneles de operador o en un ordenador. Los sistemas HMI en ordenadores se los conoce también como software HMI o de monitoreo y control de supervisión. Las señales del proceso son conducidas al HMI por medio de dispositivos como tarjetas de entrada/salida en el ordenador, PLC's (Controladores lógicos programables), PACs (Controlador de automatización programable), RTU (Unidades remotas de I/O) o DRIVER's (Variadores de velocidad de motores). Todos estos dispositivos deben tener una comunicación que entienda el HMI. La industria de HMI nació esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLCs y otros mecanismos de control.

Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuyen a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual, los sistemas SCADA lo hacen de manera automática. Históricamente los PLC no tienen una manera estándar de presentar la información al operador. La obtención de los datos por el sistema SCADA parte desde el PLC o desde otros controladores y se realiza por medio de algún tipo de red, posteriormente esta información es combinada y formateada. Un HMI puede tener también vínculos con una base de datos para proporcionar las tendencias, los datos de diagnóstico y manejo de la información así como un cronograma de procedimientos de mantenimiento, información logística, esquemas detallados para un censor o máquina en particular, incluso sistemas expertos con guía de resolución de problemas. Desde cerca de 1998, virtualmente todos los productores principales de PLC ofrecen integración con sistemas HMI/SCADA, muchos de ellos usan protocolos de comunicaciones abiertos y no propietarios.

Numerosos paquetes de HMI/SCADA de terceros ofrecen compatibilidad incorporada con la mayoría de PLCs, incluyendo la entrada al mercado de ingenieros mecánicos, eléctricos y técnicos para configurar estas interfaces por sí mismos, sin la necesidad de un programa hecho a medida escrito por un desarrollador de software. SCADA es popular debido a esta compatibilidad y seguridad. Ésta se usa desde aplicaciones pequeñas, como controladores de temperatura en un espacio, hasta aplicaciones muy grandes como el control de plantas nucleares.

2.13.4 Soluciones de Hardware

La solución de SCADA a menudo tiene componentes de sistemas de control distribuido, DCS (Distribuited Control System). El uso de RTUs o PLCs o últimamente PACs sin involucrar computadoras maestras está aumentando, los cuales son autónomos ejecutando procesos de lógica simple.

Frecuentemente se usa un lenguaje de programación funcional para crear programas que corran en estos RTUs y PLCs, siempre siguiendo los estándares de la norma IEC 61131-3. La complejidad y la naturaleza de este tipo de programación hace que los programadores necesiten cierta especialización y conocimiento sobre los actuadores que van a programar. Aunque la programación de estos elementos es ligeramente distinta a la programación tradicional, también se usan lenguajes que establecen procedimientos, como pueden ser FORTRAN, C o Ada95. Esto les permite a los ingenieros de sistemas SCADA implementar programas para ser ejecutados en RTUs o un PLCs.

2.13.5 Componentes del sistema

Los tres componentes de un sistema SCADA son:

1. Múltiples Unidades de Terminal Remota (también conocida como UTR, RTU o Estaciones Externas).
2. Estación Maestra y Computador con HMI.
3. Infraestructura de Comunicación.

2.13.6 Unidad de Terminal Remota (UTR)

La UTR se conecta al equipo físicamente y lee los datos de estado como los estados abierto/cerrado desde una válvula o un interruptor, lee las medidas como presión, flujo, voltaje o corriente. Por el equipo el UTR puede enviar señales que pueden controlarlo: abrirlo, cerrarlo, intercambiar la válvula o configurar la velocidad de la bomba, ponerla en marcha, pararla.

La UTR puede leer el estado de los datos digitales o medidas de datos analógicos y envía comandos digitales de salida o puntos de ajuste analógicos. Una de las partes más importantes de la implementación de SCADA son las alarmas. Una alarma es un punto de estado digital que tiene cada valor NORMAL o ALARMA. La alarma se puede crear en cada paso que los requerimientos lo necesiten. Un ejemplo de un alarma es la luz de "tanque de combustible vacío"del automóvil.

El operador de SCADA pone atención a la parte del sistema que lo requiera, por la alarma. Pueden enviarse por correo electrónico o mensajes de texto con la activación de una alarma, alertando al administrador o incluso al operador de SCADA.

2.13.7 Estación Maestra

El término "Estación Maestra" se refiere a los servidores y el software responsable para comunicarse con el equipo del campo (UTRs, PLCs, etc) en estos se encuentra el software HMI corriendo para las estaciones de trabajo en el cuarto de control, o en cualquier otro lado.

En un sistema SCADA pequeño, la estación maestra puede estar en un solo computador, A gran escala, en los sistemas SCADA la estación maestra puede incluir muchos servidores, aplicaciones de software distribuido, y sitios de recuperación de desastres.

El sistema SCADA usualmente presenta la información al personal operativo de manera gráfica, en forma de un diagrama de representación. Esto significa que el operador puede ver un esquema que representa la planta que está siendo controlada. Por ejemplo un dibujo de una bomba conectada a la tubería puede mostrar al operador cuanto fluido esta siendo bombeado desde la bomba a través de la tubería en un momento dado o bien el nivel de líquido de un tanque o si la válvula esta abierta o cerrada. Los diagramas de representación puede consistir en gráficos de líneas y símbolos esquemáticos para representar los elementos del proceso, o pueden consistir en fotografías digitales de los equipos sobre los cuales se animan las secuencias.

Los bloques software de un SCADA (módulos), permiten actividades de adquisición, supervisión y control; Características: - Configuración: permite definir el entorno de trabajo del SCADA, adaptándolo a la aplicación particular que se desea desarrollar. - Interfaz gráfico del operador: proporciona al operador las funciones de control y supervisión de la planta.

El proceso se representa mediante sinópticos gráficos almacenados en el ordenador de proceso y generados desde el editor incorporado en el SCADA o importados desde otra aplicación durante la configuración del paquete. - Módulo de proceso: ejecuta las acciones de mando preprogramadas a partir de los valores actuales de variables leídas. - Gestión y archivo de datos: almacenamiento y procesado ordenado de datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos. - Comunicaciones: transferencia de información entre la planta y la arquitectura hardware que soporta el SCADA, y también entre ésta y el resto de elementos informáticos de gestión.

El paquete HMI para el sistema SCADA típicamente incluye un programa de dibujo con el cual los operadores o el personal de mantenimiento del sistema pueden cambiar la apariencia de la interfaz.

Estas representaciones pueden ser tan simples como unas luces de tráfico en pantalla, las cuales representan el estado actual de un campo en el tráfico actual, o tan complejas como una pantalla de multiproyector representando posiciones de todos los elevadores en un rascacielos o todos los trenes de una vía férrea. Plataformas abiertas como Linux que no eran ampliamente usados inicialmente, se usan debido al ambiente de desarrollo altamente dinámico y porque un cliente que tiene la capacidad de acomodarse en el campo del

hardware y mecanismos a ser controlados que usualmente se venden UNIX o con licencias OpenVMS. Hoy todos los grandes sistemas son usados en los servidores de la estación maestra así como en las estaciones de trabajo HMI.

2.13.8 Filosofía Operacional

En vez de confiar en la intervención del operador o en la automatización de la estación maestra los RTU pueden ahora ser requeridos para operar ellos mismos, realizando su propio control sobre todo por temas de seguridad. El software de la estación maestra requiere hacer más análisis de datos antes de ser presentados a los operadores, incluyendo análisis históricos y análisis asociados con los requerimientos de la industria particular. Los requerimientos de seguridad están siendo aplicados en los sistemas como un todo e incluso el software de la estación maestra debe implementar los estándares más fuertes de seguridad en ciertos mercados.

Para algunas instalaciones, los costos que pueden derivar de los fallos de un sistema de control es extremadamente alto, es posible incluso haya riesgo de herir las personas. El hardware del sistema SCADA es generalmente lo suficientemente robusto para resistir condiciones de temperatura, humedad, vibración y voltajes extremos pero en estas instalaciones es común aumentar la fiabilidad mediante hardware redundante y varios canales de comunicación. Una parte que falla puede ser fácilmente identificada y su funcionalidad puede ser automáticamente desarrollada por un hardware de backup. Una parte que falle puede ser reemplazada sin interrumpir el proceso. La confianza en cada sistema puede ser calculado estadísticamente y este estado es el significado de tiempo medio entre fallos, el cual es una variable que acumula tiempos entre fallas. El resultado calculado significa que el tiempo medio entre fallos de sistemas de alta fiabilidad puede ser de siglos.

2.13.9 Infraestructura y Métodos de Comunicación

Los sistemas SCADA tienen tradicionalmente una combinación de radios y señales directas seriales o conexiones de módem para conocer los requerimientos de comunicaciones, incluso Ethernet e IP sobre SONET (fibra óptica) es también frecuentemente usada en sitios muy grandes como ferrocarriles y estaciones de energía eléctrica. Es más, los métodos de conexión entre sistemas puede incluso que sea a través de comunicación wireless (por ejemplo si queremos enviar la señal a una PDA, a un teléfono móvil,...) y así no tener que emplear cables.

Para que la instalación de un SCADA sea perfectamente aprovechada, debe de cumplir varios objetivos:

1. Deben ser sistemas de arquitectura abierta (capaces de adaptarse según las necesidades de la empresa).
2. Deben comunicarse con facilidad al usuario con el equipo de planta y resto de la empresa (redes locales y de gestión).

3. Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware. También tienen que ser de utilización fácil.

2.13.10 Aplicaciones SCADA

- Acimut Scada Monitoriza - Creación de proyectos SCADA funcionales mediante "pinchar y arrastrar"
- FreeSCADA - Aplicación "Open source" para proyectos SCADA
- Likindoy Profesional free GPL Scada system - Centrologic
- FAST/TOOLS - Yokogawa FAST/TOOLS SCADA

Una de las soluciones en el control SCADA es utilizando un programa para monitorizar, controlar y automatizar señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos. Uno de los programas más utilizados para este fin es el LabView (National Instruments).

2.14 Ping

La utilidad ping comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco (ambos definidos en el protocolo de red ICMP) para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP. Muchas veces se utiliza para medir la latencia o tiempo que tardan en comunicarse dos puntos remotos, y por ello, se utiliza entre los aficionados a los juegos en red el término PING para referirse al lag o latencia de su conexión. Existe otro tipo, Ping ATM, que se utiliza en las redes ATM (como puede ser una simple ADSL instalada en casa) y, en este caso, las tramas que se transmiten son ATM (nivel 2 del modelo OSI). Este tipo de paquetes se envían para probar si los enlaces ATM están correctamente definidos.

2.15 Tag

Una etiqueta o baliza (términos a veces reemplazados por el anglicismo Tag) es una marca con tipo que delimita una región en los lenguajes basados en XML. También puede referirse a un conjunto de juegos informáticos interactivos que se añade a un elemento de los datos para identificarlo (Oxford English Dictionary). Esto ocurre, por ejemplo, en los archivos MP3 que guardan información sobre una canción así como sobre el artista que la ha cantado o compuesto. Como ocurre en otros casos, a menudo se emplea la palabra inglesa (Tag) a pesar de que «etiqueta» o «baliza» son perfectamente adecuadas. Con la llegada de la World Wide Web ha habido una invasión de tags. La Web se basa en el HTML, o «lenguaje de marcado de hipertexto», que está basado en el uso de etiquetas. Las etiquetas (entre otras muchas cosas) le dicen al programa visualizador de páginas Web (o navegador) en qué juego de caracteres está la página, de qué tipo es cada uno de los fragmentos de texto que contiene (por ejemplo, encabezamiento, texto normal, etc.), si

están alineados a un lado o centrados, en qué tipo de letra está el texto (cursiva, negrita, etc.), si hay tablas, de qué anchura son etc. Dicho de otro modo: las balizas dan al navegador las instrucciones necesarias para que presente la página en pantalla.

2.16 Protocolos De Comunicación

Se conoce como protocolo de comunicaciones a un conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre sistemas.
El protocolo DNP (Distributed Network Protocol), originalmente desarrollado por Westronic Inc. en 1990, actualmente GE Energy Services, documentado y puesto al público en 1993, es un protocolo basado en los estándares de comunicación IEC 870-5 diseñado para la industria en aplicaciones de telecontrol, especialmente enfocado hacia el sector eléctrico por la precisión y calidad de la información que transporta. Es un protocolo de comunicaciones abierto y no propietario diseñado basándose en un modelo que incluye tres de las capas del modelo OSI (Open Systems Interconnections), denominado EPA (Enhanced Performance Architecture): Capa de Aplicación, Capa de Enlace de Datos y Capa Física.

El DNP 3.0 es muy eficiente por ser un protocolo de capas, mientras que asegura alta integridad de datos. Es adecuado para aplicaciones en el ambiente SCADA completo: RTU-IED, Maestra–Remota, punto-punto y aplicaciones de red.

Características del DNP 3.0:

a. Pueden existir más de 65000 dispositivos con direcciones diferentes en un mismo enlace.
b. Permite mensajes en “Broadcast”.
c. Confirmaciones al nivel de la Capa de Enlace y/o Capa de Aplicación garantizando así alta integridad en la información.
d. Solicitudes y respuestas con múltiples tipos de datos en un solo mensaje, y permite objetos definidos por el usuario incluyendo la transferencia de archivos.
e. Segmentación de los mensajes en múltiples tramas para garantizar una excelente detección de errores y recuperación de tramas con errores.
f. Puede incluir solo datos que hallan cambiado en el mensaje de respuesta (Reporte por excepción).
g. Asigna prioridades a un grupo de datos (clases), y los solicita periódicamente basándose en las mismas.
h. Los dispositivos esclavos pueden enviar respuestas sin solicitud (Respuestas no Solicitadas).
i. Soporta sincronización temporal con un formato de tiempo estándar.

Esta es una capacidad que tiene el protocolo DNP 3.0, que permite a los dispositivos esclavos respondan a los maestros sin que éstos los interroguen. Por lo general se usa esta característica para que los dispositivos esclavos

reporten los eventos ya sean las alarmas, secuencia de eventos y/o cambios en las mediciones sin necesidad de preguntar por ellos.

El criterio para que un dispositivo esclavo reporte Respuestas no Solicitadas se basa en dos parámetros, configurables en todo dispositivo que se comunique en DNP 3.0 y que soporte esta propiedad:

a. HOLD COUNT: este parámetro configura un número determinado de eventos o cambios que tienen que ocurrir para que el dispositivo tome la decisión de enviar una Respuesta no Solicitada reportando dichos eventos.

b. HOLD TIME: este parámetro configura el tiempo máximo que debe pasar hasta que el dispositivo envíe una Respuesta no Solicitada. Con este parámetro se evita el caso en que ocurran eventos en el dispositivo pero que no superen en número al HOLD COUNT, entonces el dispositivo espera el HOLD TIME para enviar los eventos que tiene almacenado.

Capitulo 3. Desarrollo del Proyecto.

El proyecto se desarrollo en un laboratorio virtual, gracias a la tecnología de un software el cual simula los cambios como las cuchillas, interruptores y protecciones que existen en las subestaciones comportándose esta como una subestación en tiempo real.

Empezaremos con el desarrollo del proyecto en el laboratorio virtual ya que es nuestra programación elemental y será heredado a nuestro programa real, entendemos por programa real a la implementación del código en las lógicas y desarrollo de las subestaciones.

Aplicaremos nuestro desarrollo lógico para cada una de las bahías de las subestaciones zona sureste de México. Teniendo en cuenta que el desarrollo será ameno debido a la similitud con la de nuestro desarrollo en el laboratorio.

3.1 Unifilar de la subestación virtual(Laboratorio).

Se realizo el diseño del unifilar de la subestación virtual (figura 3.1) en el software InTouch desarrollados por los elementos del departamento, el cual nos ayudara al diseño de la lógica de compuertas.

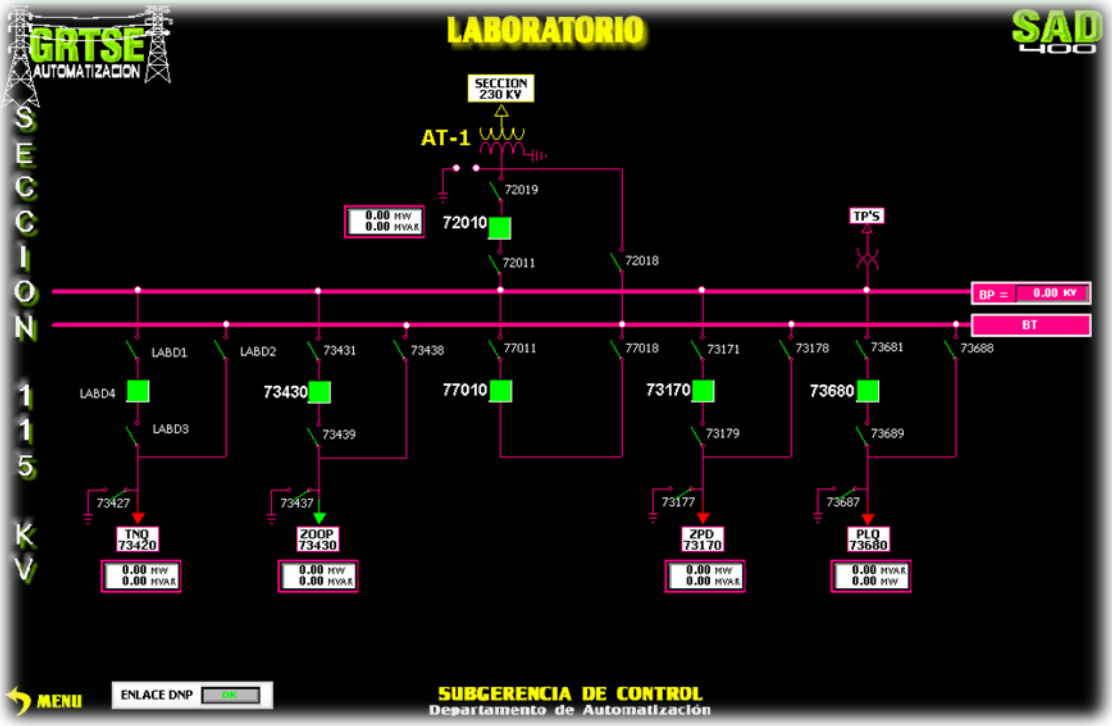


Figura 3.1 Unifilar del laboratorio.

3.2 Diseño de la lógica de compuertas de la subestación virtual.

Gracias al unifilar diseñado se pudo realizar la lógica de compuertas, la cual tendrá una gran utilidad para la programación en KOP o RLL en InControl 7.11 (figura 3.2)

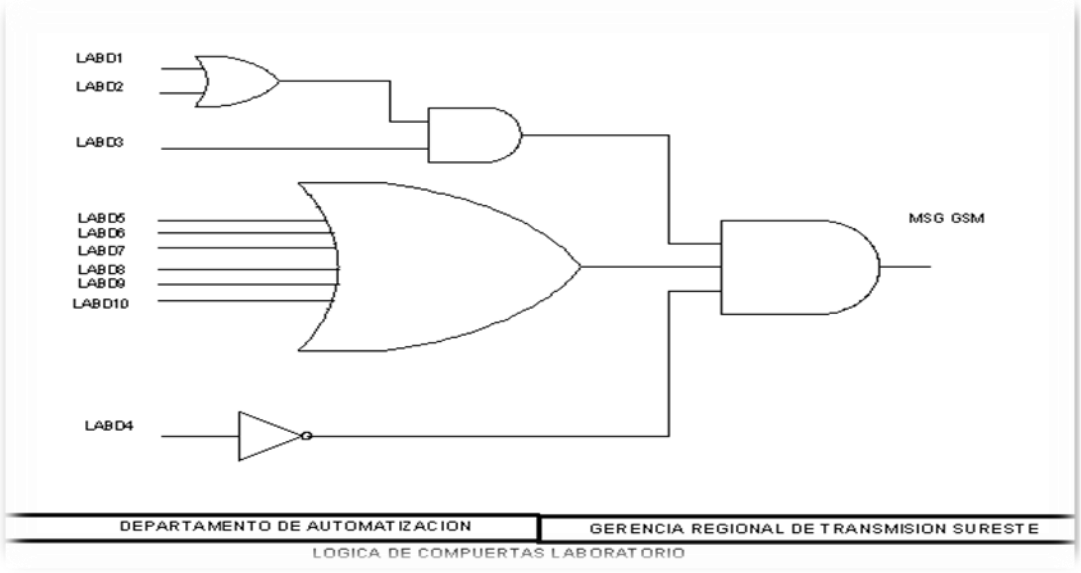


Figura 3.2 Lógica de compuerta del laboratorio.

3.3 Diagramas De Tiempos.

- Los diagramas de tiempos son importantes porque ellos nos darán la pauta del mensaje y cuando deben de salir, a continuación se explicaran cuando deben de salir los mensajes.
- En este diagrama de tiempo simulamos un interruptor que se encuentra en 1 pero en un tiempo se abre el interruptor, cuando se abre el interruptor checara un minuto hacia tras para checar si existió una protección que estuvo en 1, si la encuentra mandara un mensaje al celular como disparo, pero cuando el interruptor se abra y no encuentre una protección que estuvo activada dentro del minuto no mandara ningún mensaje de disparo (figura 3.3).
- Cuando el interruptor se cierra volverá a checar una protección hacia tras si encuentra una protección que estuvo en 1 dentro del minuto mandara un mensaje de interruptor cerrado, pero si no encuentra ninguna protección que estuvo en 1 dentro del minuto no mandara el mensaje de cerrado (figura 3.3).

Nota: el programa realizado checa 1 minuto hacia tras y hacia delante para que no haya ningún problema más adelante.

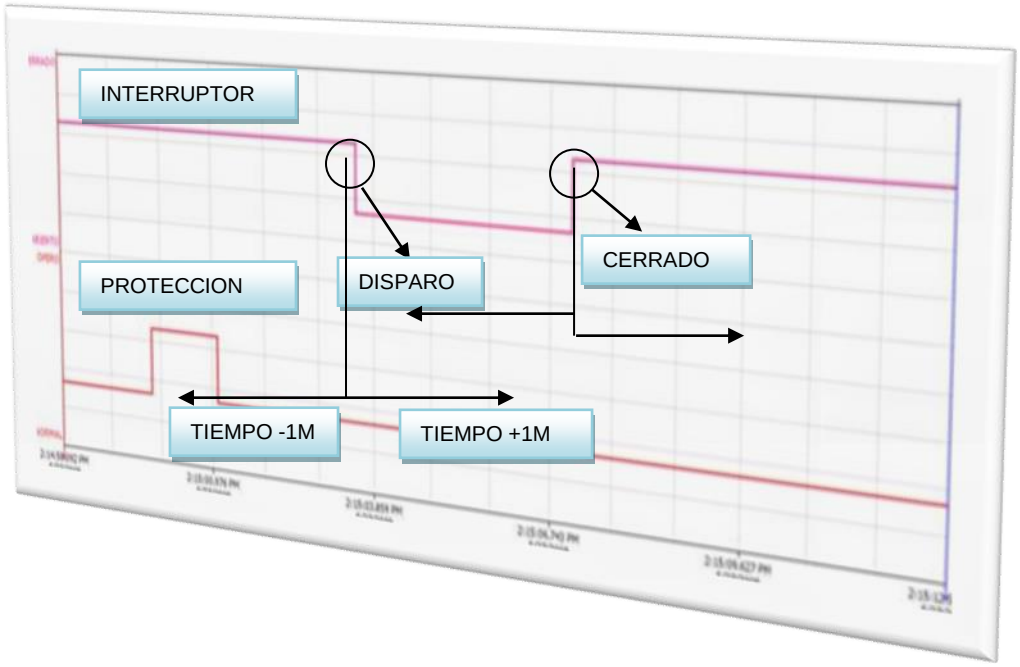


Fig. 3.3 Comportamiento del interruptor en diagrama de tiempo.

- En este diagrama de tiempos es muy versátil porque se utilizan 5 protecciones activadas para un solo interruptor, en las subestaciones se utilizan más de 10 interruptores, cada interruptor puede tener infinidades de protecciones nos debe de generar mensajes de disparo y mensajes de cerrado la cual llevara una etiqueta de tiempo diferente, ya antes mencionado como checar un disparo o un cerrado (figura 3.4).
- Este caso es cuando el interruptor checa un minuto hacia tras si se activo antes una protección (figura 3.4).

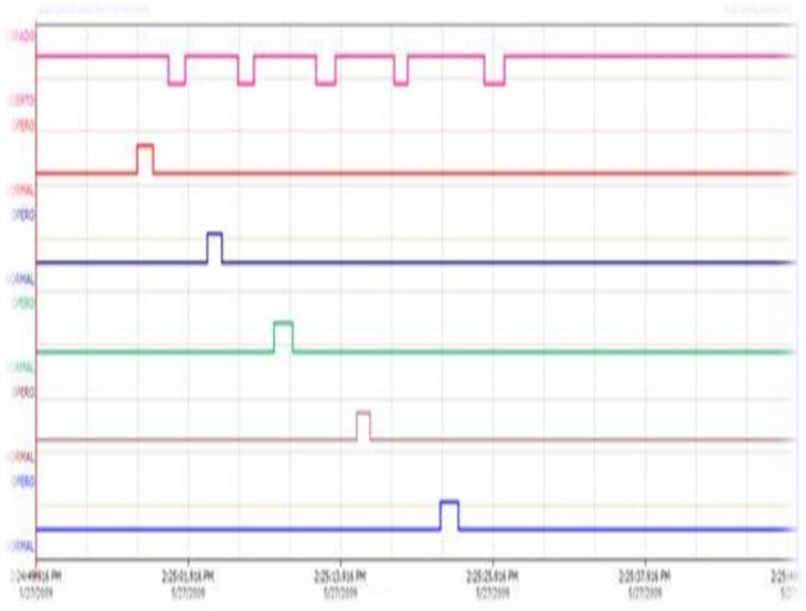


Fig. 3.4 Activación de las protecciones antes que el evento del interruptor.

- Este caso también mandara mensajes de disparo y de cerrado ahora el interruptor checa un minuto hacia delante para checar si se activo antes una protección (figura 3.5).

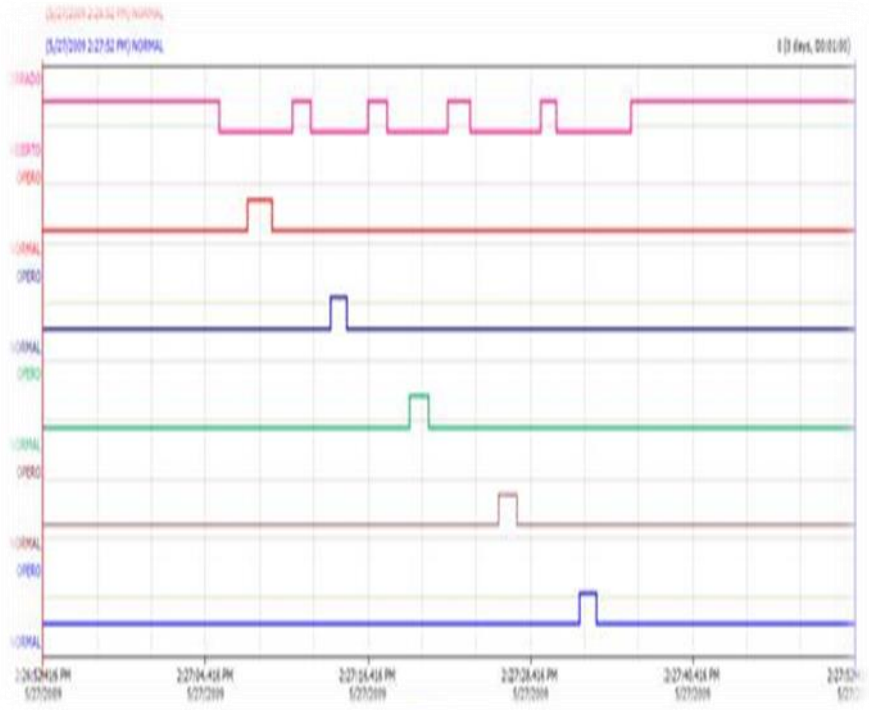


Fig. 3.5 Activación de las protecciones desues que el evento del interruptor.

- En condiciones normales el interruptor se mantiene cerrado (figura 3.6).

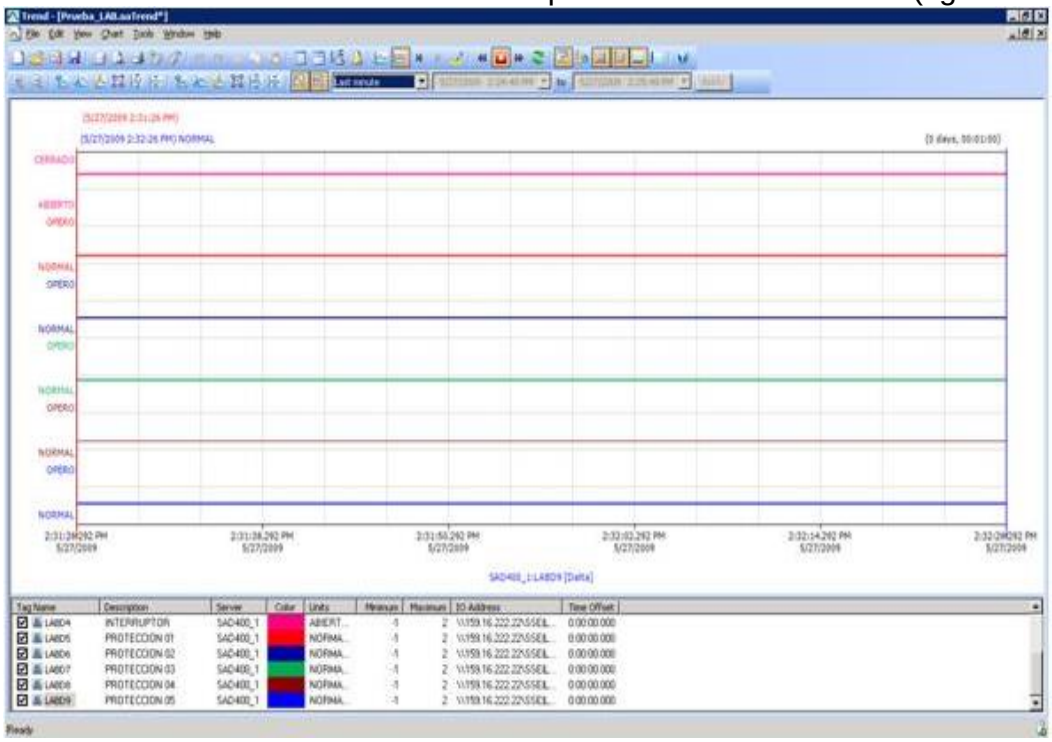


Fig. 3.6 Estado normal de las bahías.

- Cuando hay mantenimiento en los interruptores o maniobras se abre el interruptor y se queda operando la protección, ahí no mandara ningún mensaje el programa.
- Esta imagen corresponde a la subestación virtual que se utilizado para los disparos y cerrados, la cual contienen Tag para cuchillas, protecciones y interruptor, la cual nos ayudo a simular lo que hace una subestación (figura 3.7).

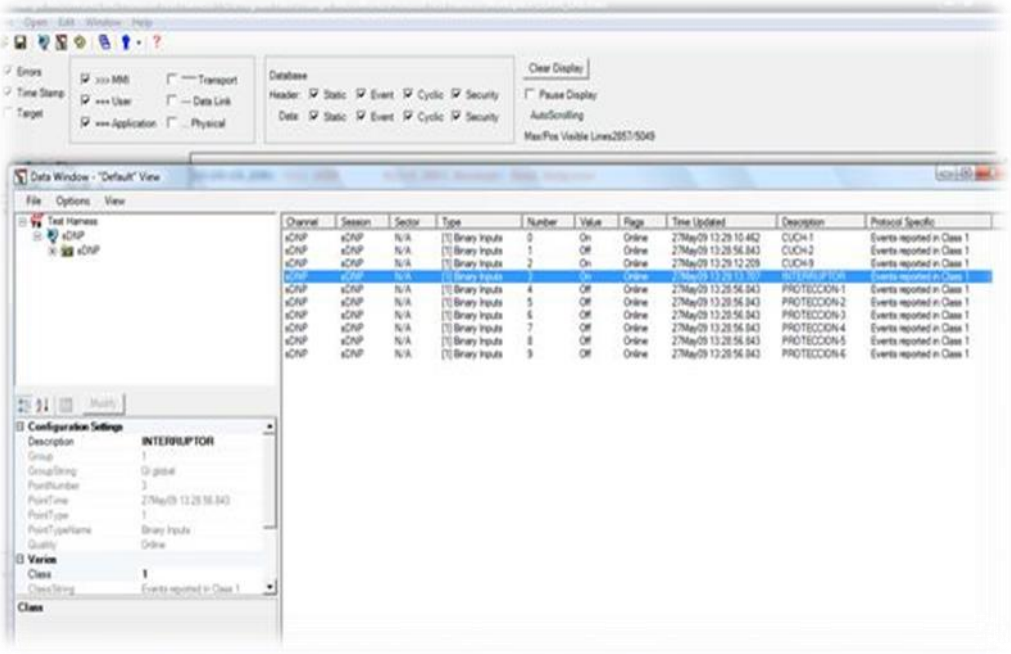
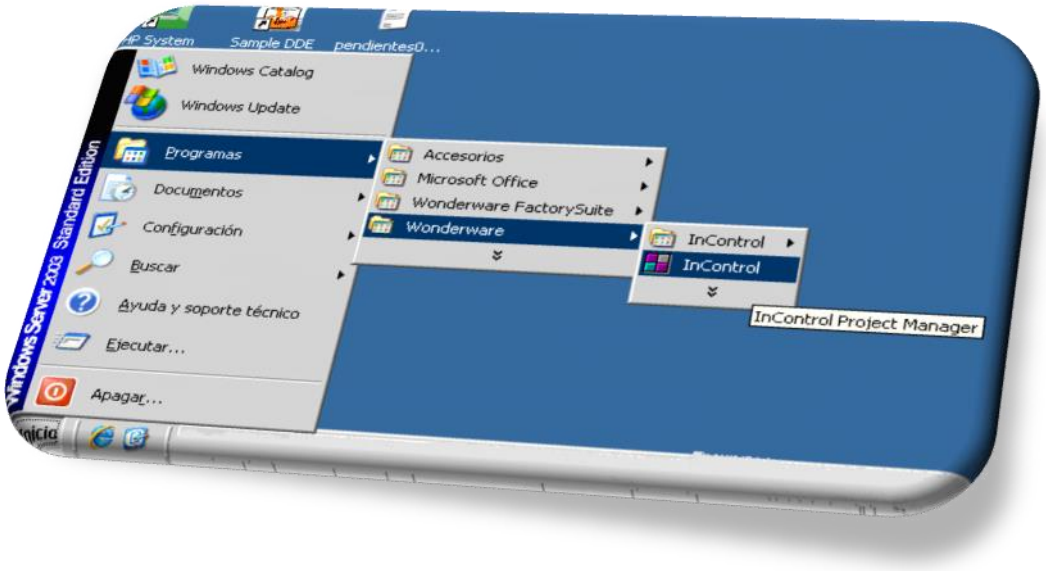


Fig. 3.7 Simulador de las subestaciones

3.4 Ejecución Del Programa Incontrol

1.- Abrir *Menú /Inicio/Programas/Wonderware/ InContro (figura*



3.9).

Fig. 3.8 ejecucion de Incontrol

3.5 Creación de un proyecto en Incontrol.

Dar clic en *New:* 

el cual se encuentra en la parte superior del cuadro de diálogo.

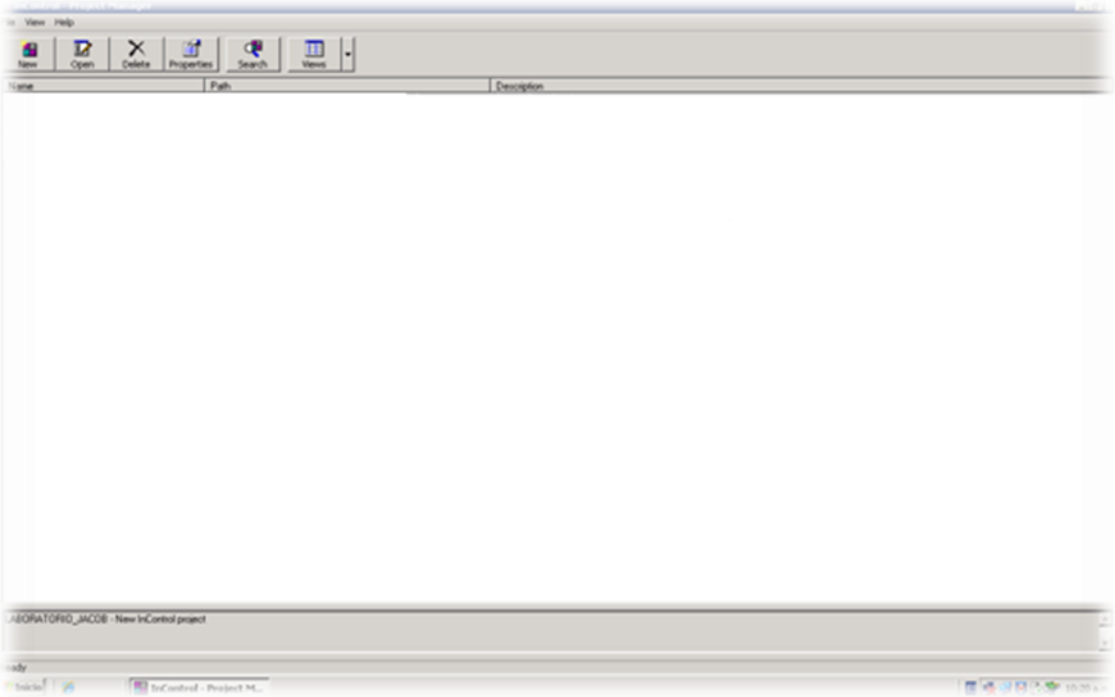


Fig.3.9 Directorio de Incontrol.

Crear un programa asignándole un nombre en el directorio por ejemplo:

C:\PROYECTOS\INCONTROL\LABORATORIO_JACOB

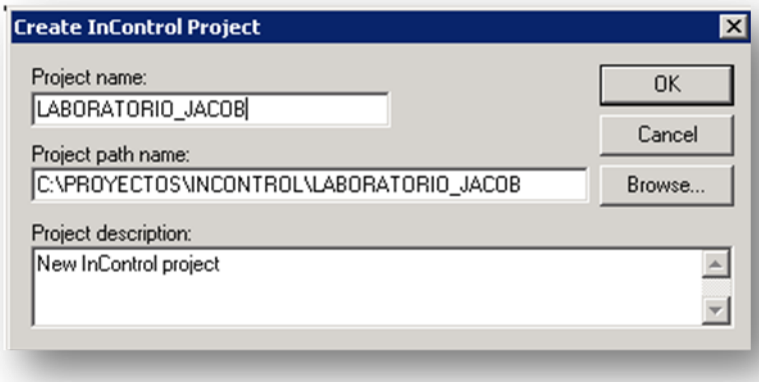


Fig.3.10 Créate Incontrol Project.

En el cuadro de diálogo de *InControl Project-Manager* aparecerá el proyecto creado, seleccionar y darle doble clic para abrirlo

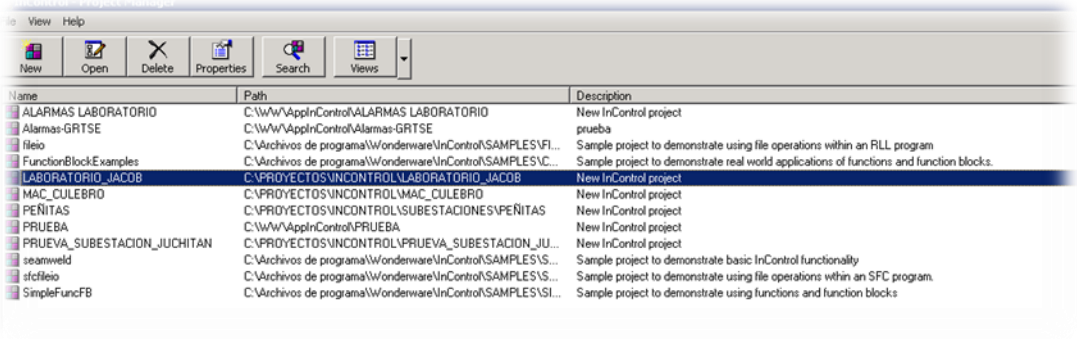


Fig.3.11 Proyecto creado en el directorio.

En el programa InControl aparecerá una pestaña llamado *Project*, si no está habilitado este panel hacer clic en menú *View*, y seleccionar *Project*

En el panel *Project* abrimos la pestaña *Project View y* damos clic derecho en el botón de *I/O* (entradas/salidas), después dar clic en *New I/O* para agregar las Tag por las cuales estarán conectados el servidor y el software.

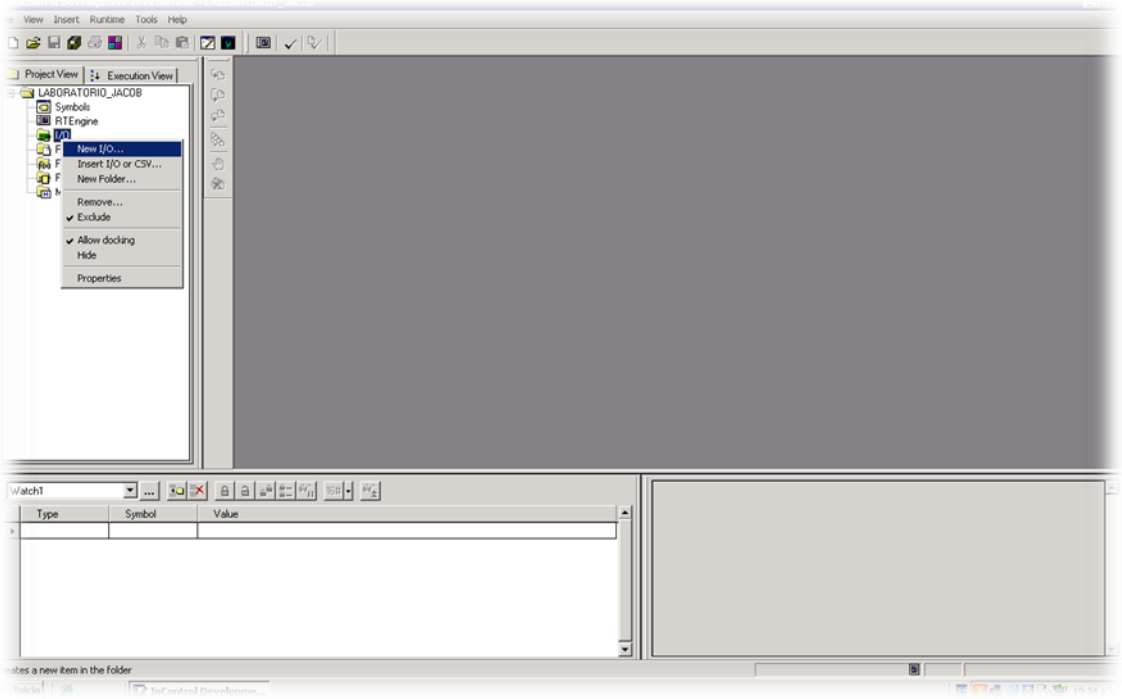


Fig.3.12 Pantalla principal de Incontrol.

Aparecerá un cuadro de diálogo llamado (*Fig.3.14*) *New* en el cual aparecen los métodos de Comunicación, para este caso se selecciona *Suite Link versión 2 (Wonderware)*.

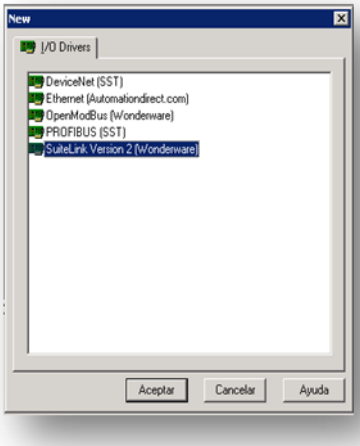


Fig.3.13 Tipo de coneccion.

Comunicación a través de SuiteLinck

A diferencia de otras E / S conductores, el conductor SuiteLink no requiere ningún hardware especial para el instalación. La figura siguiente muestra ejemplos de las conexiones de SuiteLink que usted puede hacer mientras InControl funciona como un cliente.

1.- InControl, un servidor en un ordenador, se comunica con InControl ejecuta como un cliente en otro equipo.

2.- InControl, un cliente en un equipo, se comunica con InTouch funcionando como un servidor en otro equipo.

3.- InControl, que funciona como un cliente, se comunica con InTouch ejecuta como un servidor en el mismo equipo.

4.- InControl, que funciona como un cliente, se comunica con cualquier Wonderware I / O servidor que se ejecuta en el mismo equipo.

Aparecerá un cuadro de diálogo llamado *SuiteLink2 – Edit/Create Driver* con la configuración del nombre del Driver: *SLC1*

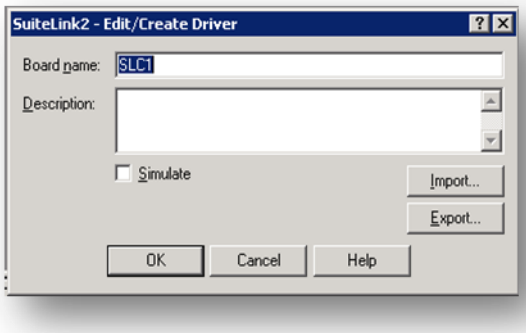


Fig.3.14 Create Driver.

En el cuadro de diálogo *SuiteLink2 Driver*, dar clic en el ícono driver *SLC1* y después dar clic en el botón *Add.*

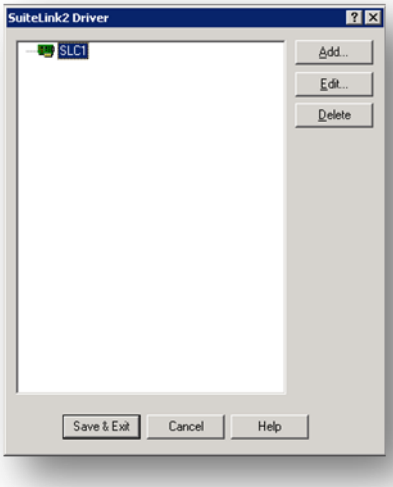


Fig.3.15 Driver creado.

Aparecerá un cuadro de diálogo llamado *SuiteLink2- Edit/Create Topic,(Fig.3.15)* el cual colocaremos los siguientes datos: *Al ver colocado estos datos dar clic en* ***OK.***

Seleccionar el Topic:

SSE_LAB(159.16.222.22\ SSE │LAB) y darle clic en *Add*, el cual creará una tag y colocar los datos correspondientes a las tags como aparece en la imagen figura siguiente.

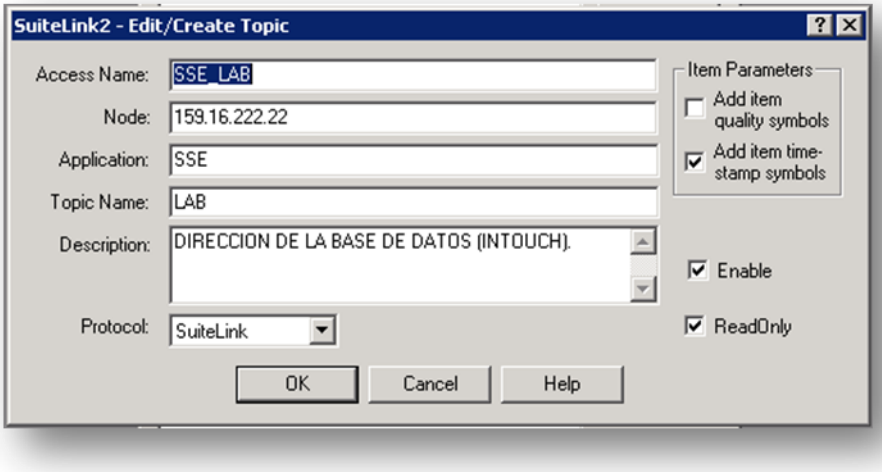


Fig.3.16 Create topic.

Seleccionar ***SSE_LAB(159.16.222.22\ SSE │LAB)*** y Add.. se abrirá la ventana donde crearemos las Tags para interruptor, cuchillas y protecciones, obtenidas de la base de datos del sistema SAD400.

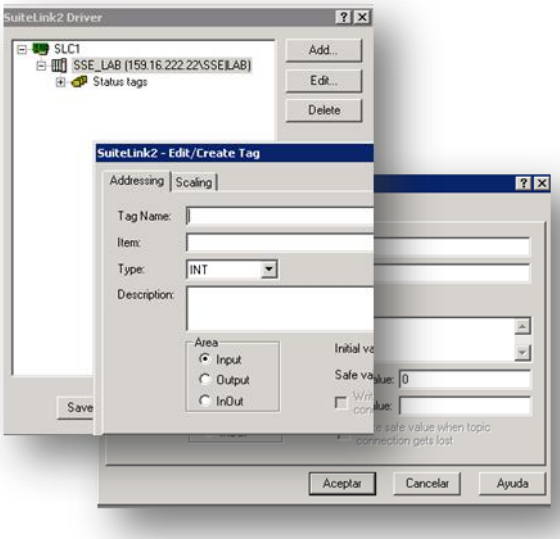


Fig.3.17 Create Tag.

Donde:

Tag Name: La Tag de laboratorio correspondiente de los elementos de interruptor, cuchillas y protecciones.

Item: Es la dirección de la tag.

Type: Bool dado que es una entrada o salida (0 ò 1).

Area: En este caso sería InOut dado que estamos manejando Type Bool

Initial Value: se coloca en FALSE ya que utilizamos Type Bool; es su valor inicial.

Safe Value: se coloca en FALSE ya que utilizamos Type Bool; valor asignado.

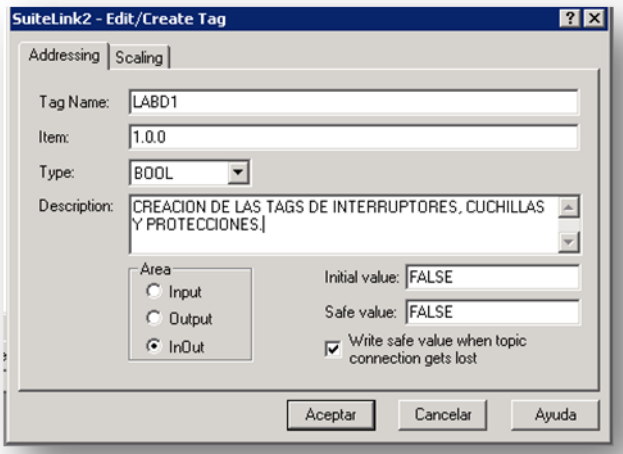


Fig.3.18 Tag creada.

Para este caso se crearán 9 tags más, repitiendo los pasos de Edit/Create Tag los datos que se van tomar en cuenta son los siguientes:

Tabla 3.1 Creación de las Tags.

TAG NAME	ITEM	TYPE	AREA	INICIAL VALUE	SAFE VALUE
LABD2	1.0.1	Bool	InOut	FALSE	FALSE
LABD3	1.0.2	Bool	InOut	FALSE	FALSE
LABD4	1.0.3	Bool	InOut	FALSE	FALSE
LABD5	1.0.4	Bool	InOut	FALSE	FALSE
LABD6	1.0.5	Bool	InOut	FALSE	FALSE
LABD7	1.0.6	Bool	InOut	FALSE	FALSE
LABD8	1.0.7	Bool	InOut	FALSE	FALSE
LABD9	1.0.8	Bool	InOut	FALSE	FALSE

LABD10	1.0.9	Bool	InOut	FALSE	FALSE

NOTA: En todas las tags seleccionar el checkbox con la leyenda: *Write safe value when topic connection gets lost*, la cual nos permitirá no perder la información de las tags en tiempo real en el momento que deje de correr el programa.

Después de haber creado las tags y haber introducido los datos correspondientes en la ventana donde se encuentran almacenadas, dar clic en *Save&Exit*

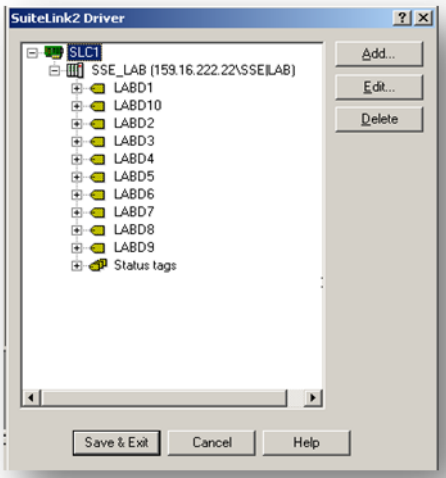


Fig.3.19 Tags creadas

Observamos que la izquierda en la pestaña *Project View* en **I/O** ya están creadas nuestras entradas y salidas (Tags de los elementos que contienen una bahía; Interruptor, Cuchillas y Protecciones) En el panel *Project* abrimos la pestaña *Project View,* dar clic derecho en *Programs* y seleccionar *New Program*. Aparecerá un cuadro de diálogo llamado: *New*, seleccione: *RLL Program* y dar clic en *Aceptar*

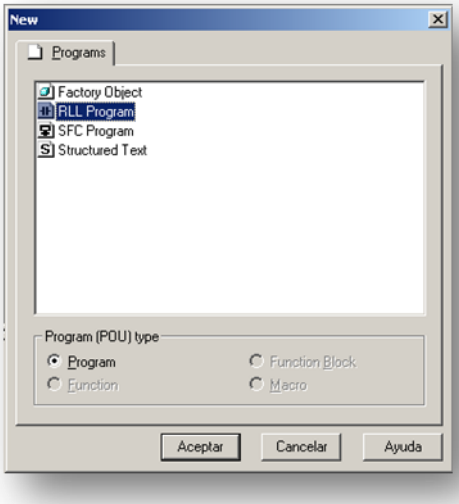


Fig.3.20 Creacion del programa en RLL.

Se abrirá una pantalla de programación RLL que a conveniencia le llamamos PROGRAMA_ESCALERA.

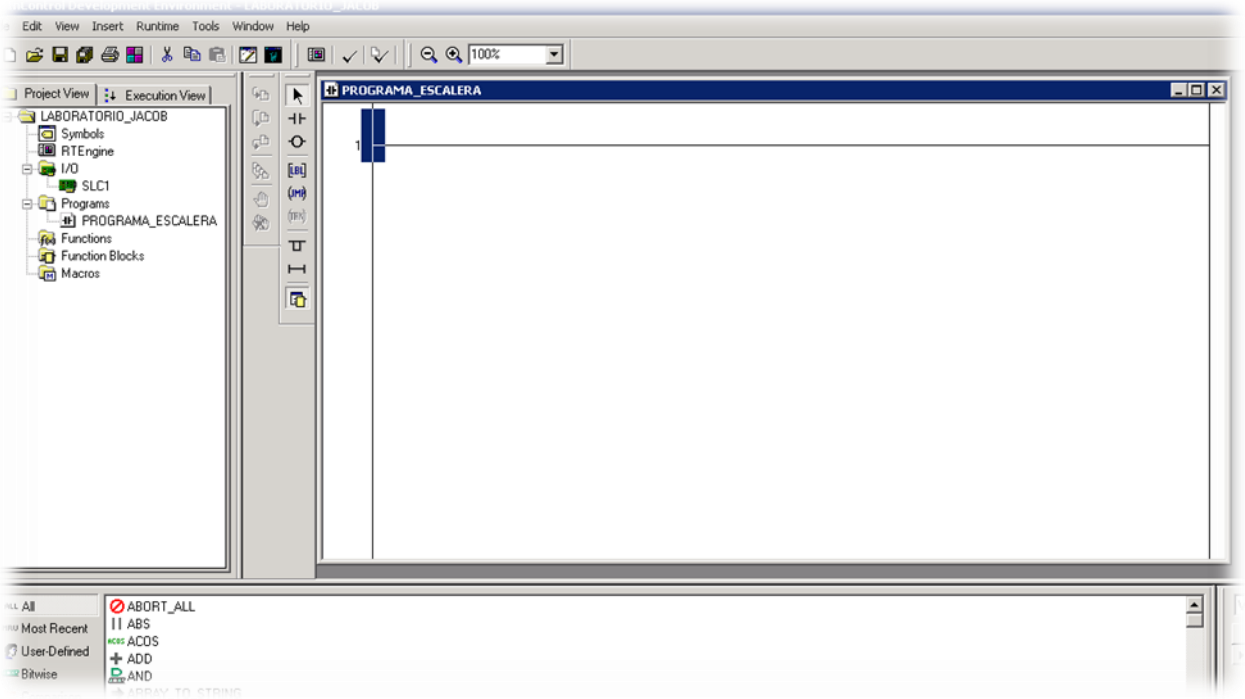


Fig.3.21 Hoja de RLL.

Nota: Debido a que este documento va dirigido a personas que tienen conocimiento del este tipo de programación; solo cabe aclarar unas pequeñas cosas. Es la herramienta de RLL de interruptores, salidas, creación de una línea paralela y creación de nueva línea,etc. Dentro de cada uno se tiene lo siguiente.

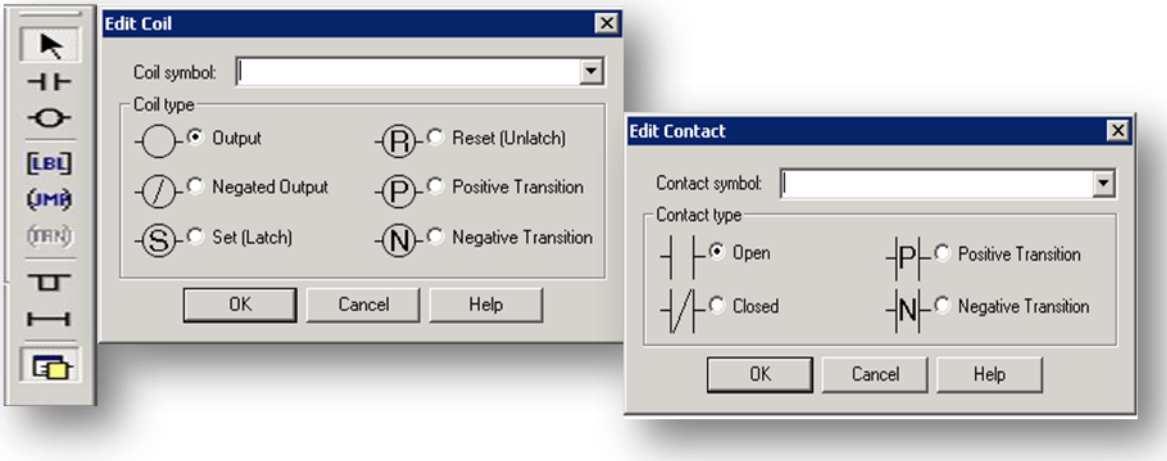


Fig.3.22 Herramientas de RLL.

Activa o desactiva Block Palette  en este block se encuentran muchos elementos lógicos, matemáticos, programables, etc.

Conforme sea la necesidad del proyecto se requerirá de dichos elementos.

Fig.3.23 Block Palette

Nos basaremos de la lógica de compuertas para la programación en escalera, cabe mencionar que será de suma importancia utilizar programación Estructurada de Texto estructurado (ST); lo explicaremos a su debido tiempo. Colocaremos una bandera para las condiciones normales con respecto a las cuchillas como se observa también en el diagrama unifilar.

3.6. Programación de la lógica de escalera con respecto a los dispositivos de la bahía de laboratorio.

Programación de las cuchillas con RLL.

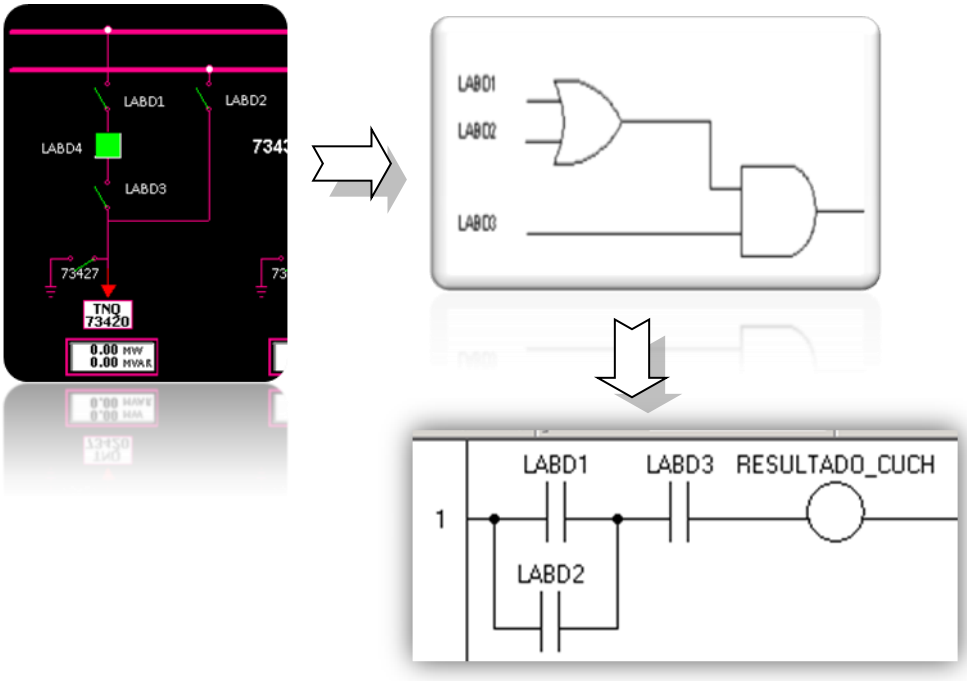


Fig.3.23 Lógica de cuchillas.

Con respecto al interruptor en condiciones normalmente cerrado se utiliza una compuerta inversora, en RLL corresponde a un interruptor cerrado con su respectiva bandera.

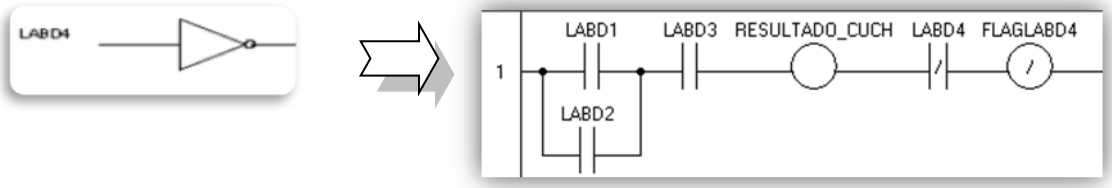
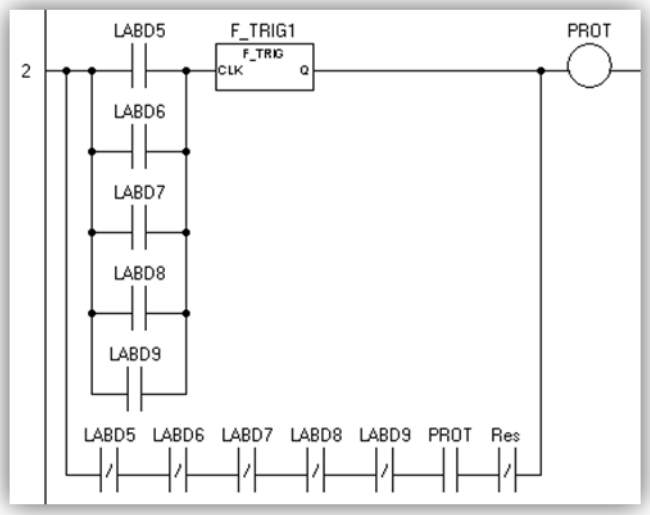


Fig.3.24 Lógica de interruptor.

El siguiente punto nos lleva a la programacion de las protecciones en este caso es necesario utilizar una programacion estructurada como la es Texto Estructurado (ST) que se encuentra en un block programable con entradas y salidas.

Antes de empezar con el block nuestro proyecto tiene necesidades técnicas que debemos de programar con respecto a las protecciones:

a) El sistema se activara con respecto a la última protección activada.
b) El sistema creara un archivo de disparo cuando el interruptor se encuentre abierto y se detecte, un minuto antes y después, una protección.
c) El sistema enviara un creara un archivo de cerrado cuando el interruptor se encuentre cerrado y se detecte, un minuto antes y después, una protección.



LABD5, LABD6, LABD7, LABD8, LABD9 son las Tags de las protecciones asignadas a nuestro laboratorio.

F_TRIG1 es un detector de flanco descendente.

PROT es nuestra salida de la lógica para las protecciones.

Fig.3.25 Lógica de protecciones caso1.

Explicación del arreglo de las protecciones:

En este caso se tomara en cuenta que la protección fue un disparo por eso se tomara en cuenta cuando la protección tiene una “caída”. Se toma en cuenta

este aspecto debido a que la protección puede estar activo “N” tiempo como será el siguiente caso.

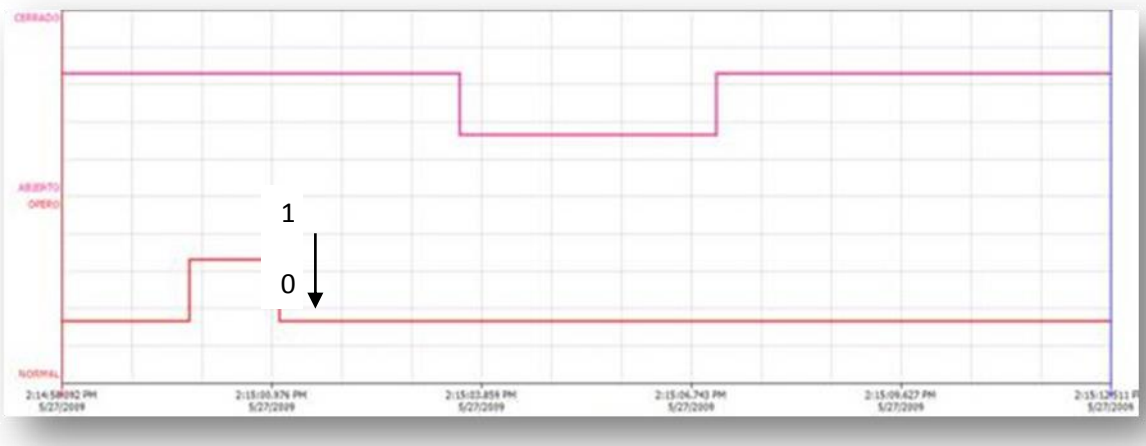


Fig.3.26 Caso 1 del la protección en diagrama de tiempo.

Caso 1.

Cuando exista este evento nuestra salida PROT se activara y quedara activada por medio del arreglo AND. Y cuando una protección se activa justo en el tiempo de flanco ASCENDENTE se desactivara la salida por medio de este mismo arreglo, y se activara cuando el detector de flanco perciba un flanco DESCENDENTE. De esta forma se soluciona el problema de detección de la última protección ya que nuestro block contendrá un cronometro con respecto a PROT mientras que se encuentra encendida y se pondrá en cero en caso contrario, por lo tanto cada vez que se active y desactive PROT el cronometro se “resetea”.

Caso 2.

Esta sería la siguiente condición cuando la protección se encuentra encendida. PRO estará activada siempre y cuando una protección se encuentre activada. Como se enseña en la figura

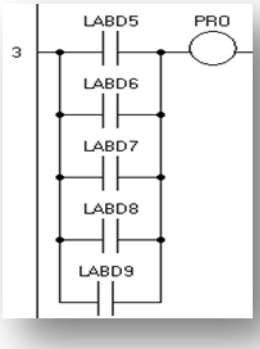


Fig.3.27 Lógica de protecciones caso2.

Este caso cumple con la analogía de protecciones.

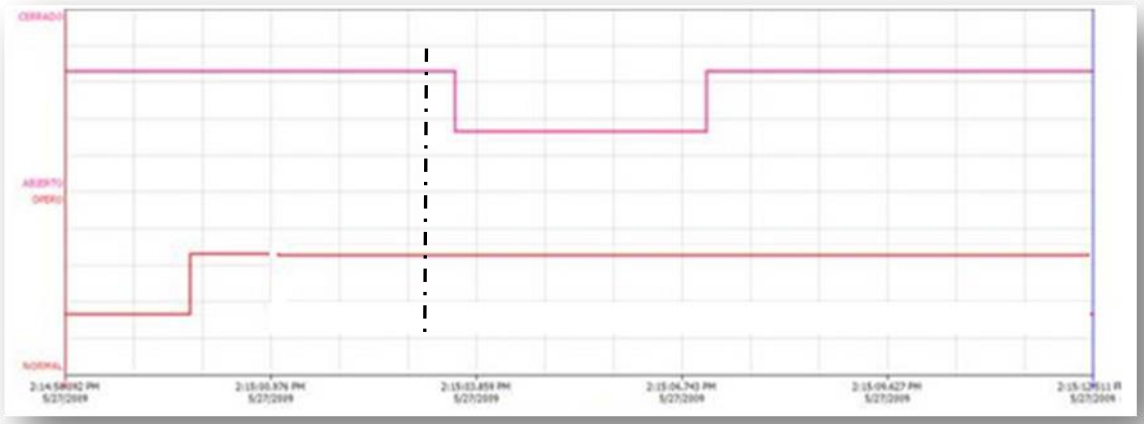


Fig.3.28 Caso 2 del la protección en diagrama de tiempo.

Este arreglo de interruptores de la programación RLL se puede programar en un *Function Block*

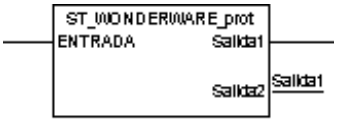


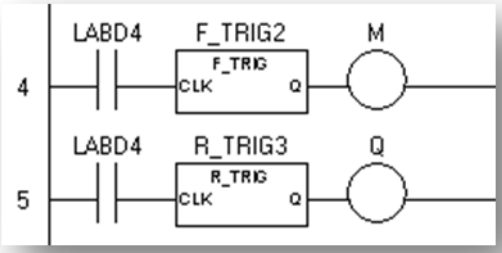
Fig.3.29 Function Block de las protecciones.

Programación del Function Block para las protecciones

```
F_TRIG001 (CLK:=ENTRADA); salida:=F_TRIG001.Q;(*Detector de flanco negativo*)
R_TRIG001 (CLK:=ENTRADA); salida_R:=R_TRIG001.Q; (*Detector de flanco
positivo*)
(*Condición cuando la protección es un disparo<= a 1 minuto *)
IF salida=1  THEN
   salida1:=1;
 END_IF;
IF salida_R  THEN
salida1:=0;
END_IF;
(* Condición cuando la protección es continua *)
IF ENTRADA = 1 THEN

   salida2 := 1;
ELSE
   salida2 := 0;
END_IF;
```

Siguiendo con las analogías del proyecto corresponde la detección de flancos para el interruptor.



F_TRIG2 detector de flanco descendente.

R_TRIG3 detector de flanco ascendente.

M salida de detector de flanco descendente.

Q salida de detector de flanco ascendente

Fig.3.30 Lógica para los dos casos del interruptor.

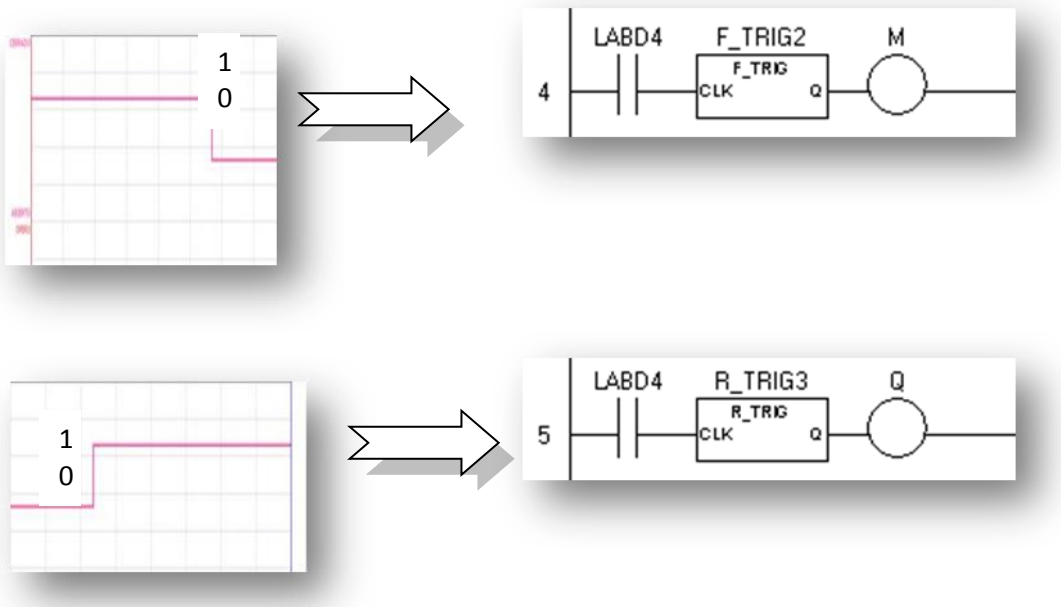


Fig.3.3
1 Comportamiento del interruptor en los 2 casos.

Al igual que las protecciones estos detectores se puedes sustituir por una función Trigger en ST localizado en el Block Palette:

F_TRIG?(CLK:=())
F_TRIG001(CLK:=INTERRUPTOR);salida_disparo:=F_TRIG001.Q;

(*Detecta un flanco descendente como en la protección debido a que el interruptor se abre en la entrada INTERRUPTOR de nuestro Function Block*)

R_TRIG?(CLK:=())

R_TRIG001(CLK:=INTERRUPTOR);salida_cerrado:=R_TRIG001.Q;

(*Detecta un flanco ascendente debido a que el interruptor se cierra en la entrada INTERRUPTOR de nuestro Function Block*)

Una vez programadas los comportamientos de elementos comprendidos en la bahía (interruptor, cuchilla y protección). Utilizaremos el block programable, utilizando la herramienta de programación ST.

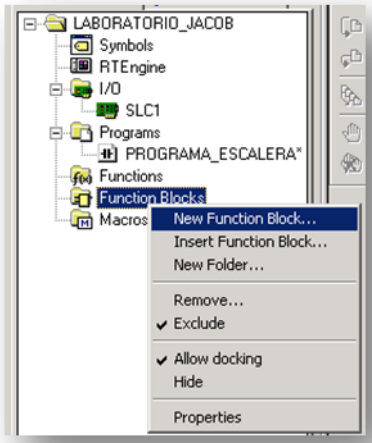


Fig.3.32 Paso1 para crear Function Block.

Aparecerá el cuadro *New* donde crearemos el programa *Structured Text.*

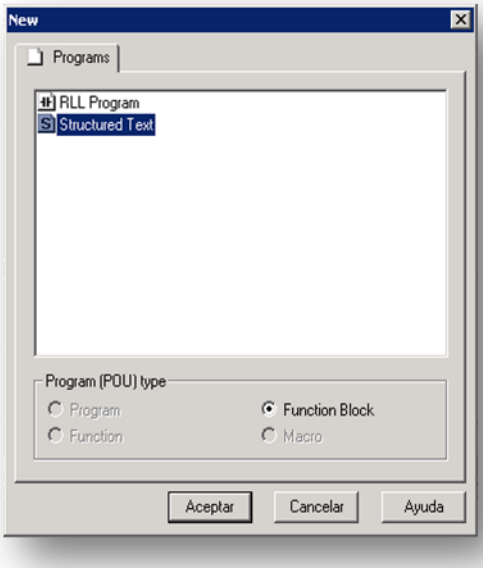


Fig.3.33 Paso2 para crear Function Block.

Guardaremos el archivo en la carpeta LABORATORIO_JACOB con el nombre PROGRAMA_ST.

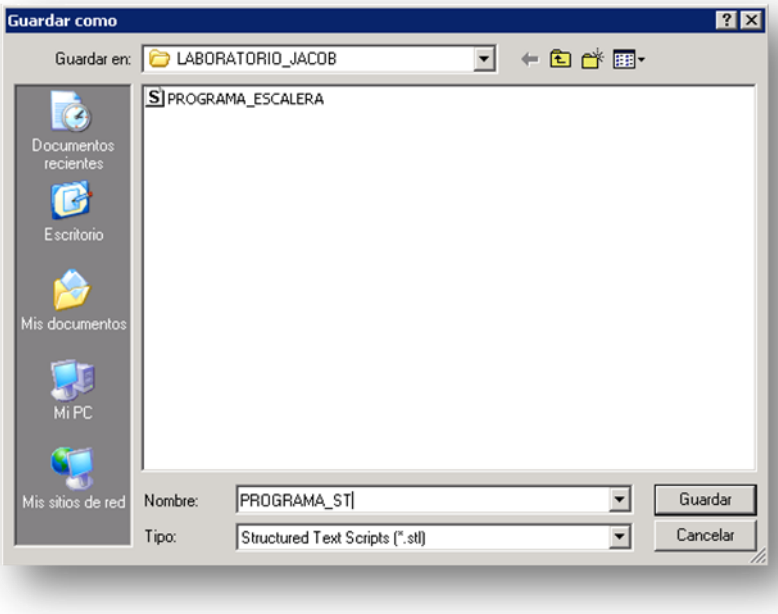


Fig.3.34 Paso3 para crear Function Block.

Una vez creado el archivo *.stl (ST) tenemos que ir a símbolo de programa para crear todos los elementos necesarios para nuestro programa;

- Temporizadores
- Entradas
- Salidas
- Variables
- Constantes
- Etiquetas de tiempo
- Etc.

En la pestaña *Project View*, Abrir el directorio *Function Blocks*, seleccionar PROGRAMA_ST, dar clic derecho y seleccionar Symbols, aparecerá el cuadro de diálogo: *Symbol Manager*.

Dar clic en el botón *New symbol*:  localizado en la parte superior del cuadro de diálogo *Symbol Manager* como se muestra en la figura.

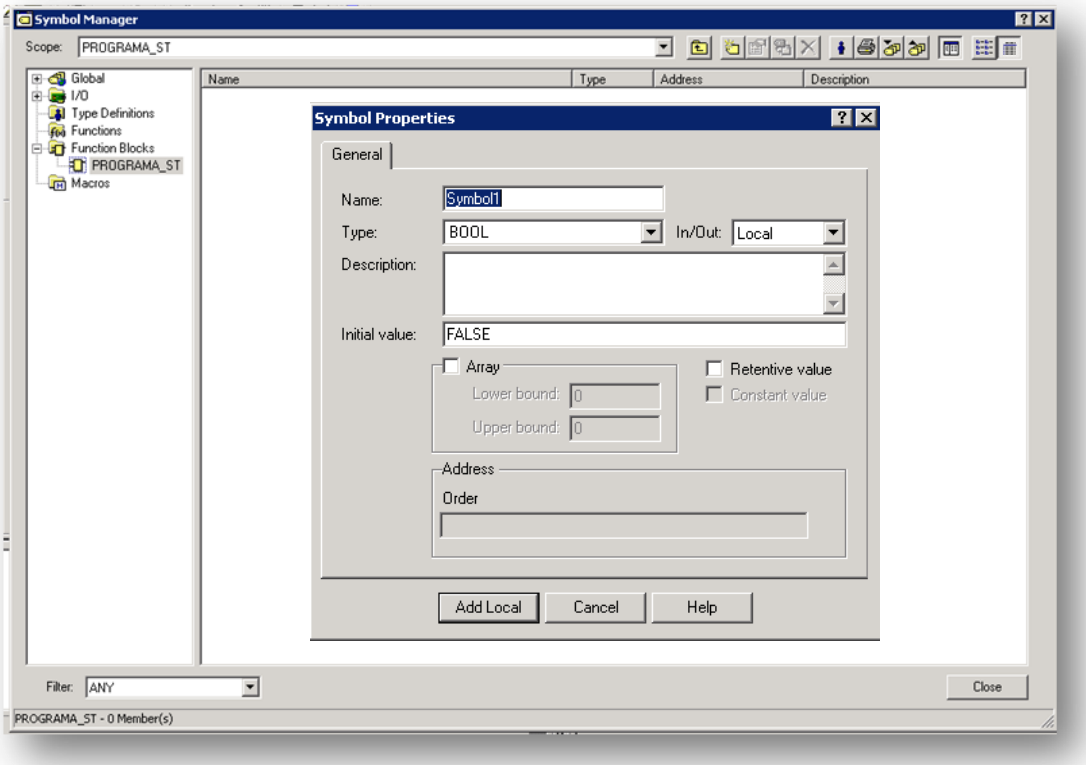


Fig.3.35 Symbol Manager.

Las analogías del programa nos exigen un criterio de programación estable y limitante en tanto a los recursos de entradas de nuestro bloque (8 entradas).

Propondremos nuestras entradas:

- Entrada de bloque
- Disparo (entrada indicadora de flanco descendente del interruptor)
- Cerrado(entrada indicadora de flanco ascendente del interruptor)
- Cuchilla(entrada indicadora del estado de las cuchillas)
- Protección(entrada indicadora de flanco descendente de la protección)
- Tiempo del interruptor (Tiempo transcurrido del interruptor)
- Tiempo de protección (Tiempo transcurrido de la protección)

Salidas:

- Salida de bloque
- Reset del sistema (Reset del sistema cuando exista una entrada al bloque)
- Creación de un archivo (Creara un archivo .tex, .inf dependiendo de la necesidad del proyecto)

Con estas analogías crearemos nuestro sistema de símbolos.

Para la entrada de bloque:

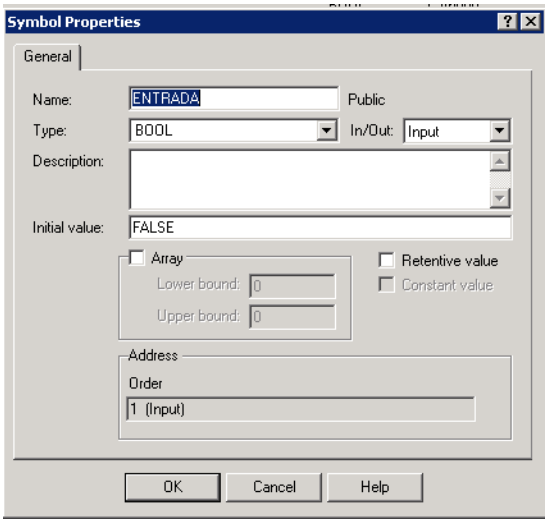


Fig.3.36 Creación de las variables en Symbol Manager.

Colóquese la siguiente información para entradas y salidas de bloque.

Tabla 3.2 Creación de las Entradas y Salidas.

NAME	TYPE	IN/OUT	INITIAL_ VALUE
ENTRADA	BOOL	InPut	FALSE
CUCH	BOOL	InPut	FALSE
PROT	BOOL	InPut	FALSE
INTERRUPTOR_NUM	STRING	InPut	----------
TIEMPO_INTERRUPTOR	DT	InPut	DT#2009-02-19-00:00:00.000
SALIDA	BOOL	OutPut	FALSE

Nota: un Type DT (Date and Time) nos “arroja” fecha y hora

Formato de fecha año-mes-día.

Formato hora hora: minutos: segundos: milisegundos.

Al utilizar recursos de programación debemos “dar de alta” en nuestro *Symbol Manager.*

Tabla 3.3 Creación de las Variables de Tiempo.

NAME	TYPE	IN/OUT	INITIAL_ VALUE
INTERR	BOOL	Local	FALSE
DR	STRING	Local	----------
EW	STRING	Local	----------
RH	STRING	Local	----------
EVENTO	STRING	Local	----------
INTER	STRING	Local	----------
R_TRIG001	R_TRIG	Local	----------
F_TRIG001	F_TRIG	Local	----------
SALIDA_DISPARO	BOOL	Local	FALSE
SALIDA_CERRADO	BOOL	Local	FALSE
SALIDA1_DISPARO	BOOL	Local	FALSE
SALIDA1_CERRADO	BOOL	Local	FALSE
PROTECC	TMR	Local	----------
TIEMPO_RESETEO_DEL_TEMPORIZADOR_PROTECCION	TIME	Local	T#0ms
TIEMPO_TRANSCURRIDO_PROTECCION	TIME	Local	T#0ms

El **TMR** tiene cuatro variables del sistema, que se identifican por el nombre del temporizador:

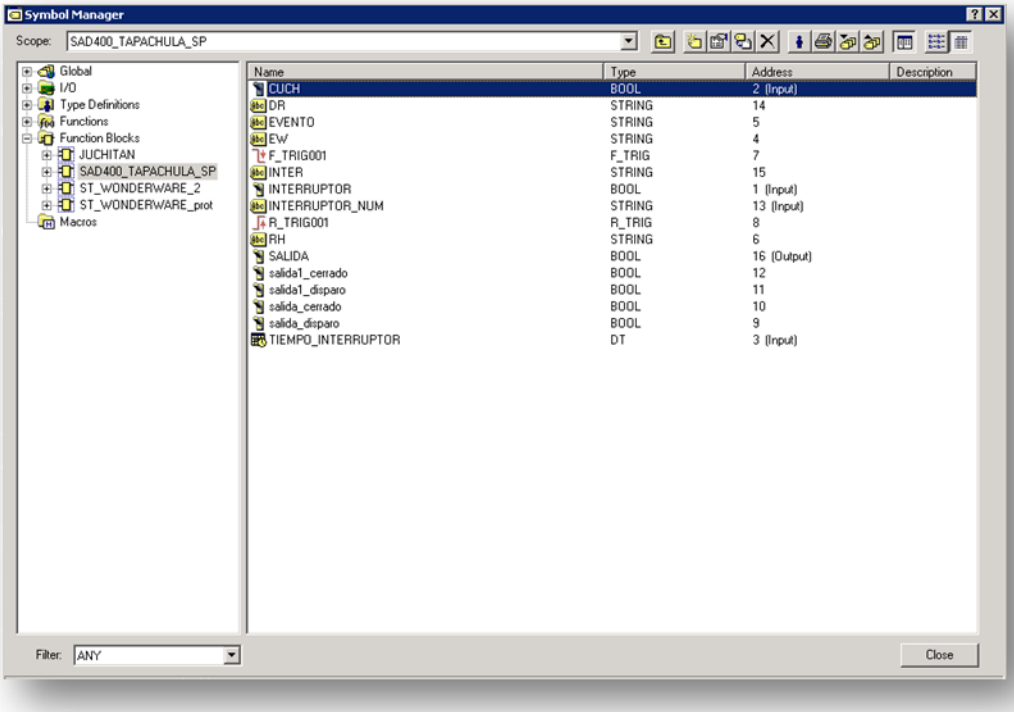


FIG.3.37 Vista de las variables en Symbol Manager.

CONFIGURACION DEL BAT.

Los archivos tipo Batch en el programa InControl permite crear archivos y guardarlos en un directorio específico del sistema operativo con información programada.

En nuestro proyecto el objetivo es crear archivos que contengan información con respecto a las bahías y al sistema utilizado.

En el panel Project abrimos la pestaña Project View, dar clic derecho en *Symbols* y seleccionar *Open*.

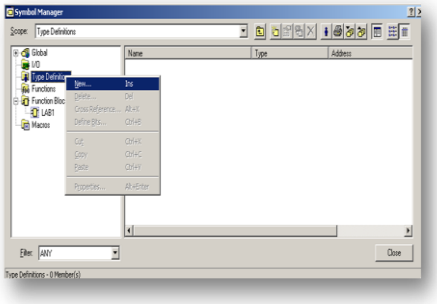


Fig.3.38 Abrir Symbol Manager.

En el cuadro de diálogo del *Symbol Manager* dar clic derecho en *Type Definitions* y seleccionar *New*

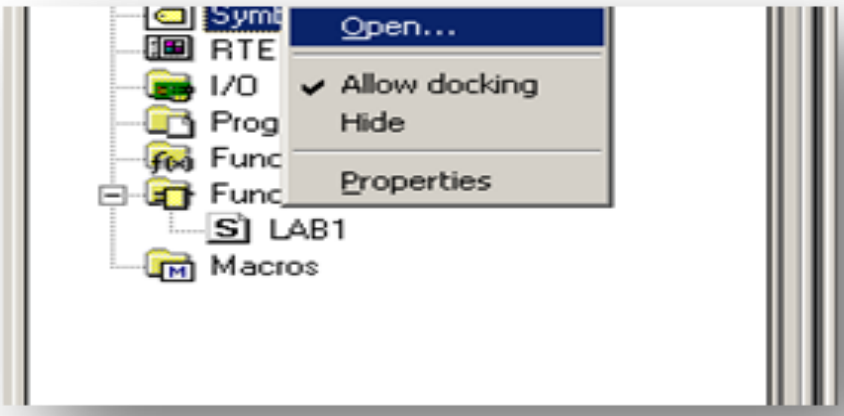


Fig.3.39 Creación de Batch en Symbol Manager.

Colocar el nombre de *Batch*, dejar por default la opción *Structure* y dar clic en *Ok.*

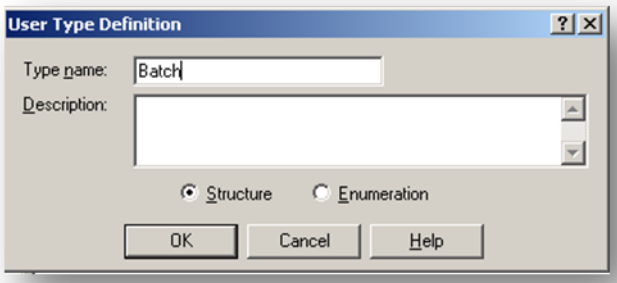


Fig.3.40 Creación de Batch en Symbol Manager.

Aparecerá en la pestaña del *Symbol Manager* el archivo *Batch*

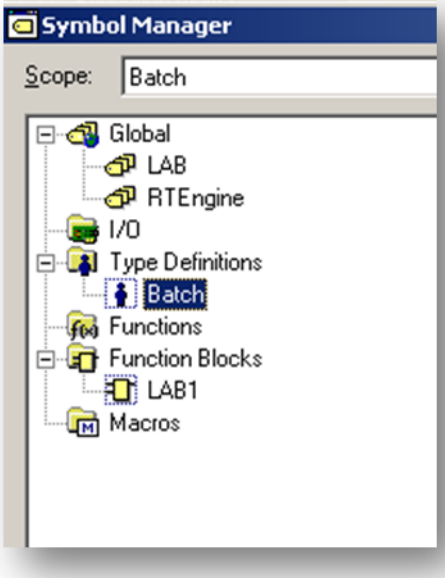


Fig.3.41 Vista de Batch en Symbol Manager.

Para agregar los símbolos que lleva esta función, en el *Symbol Manager* seleccionar *Batch* y darle clic, por si no está seleccionado. Dar clic en *New*  , aparecerá el cuadro de diálogo de *Symbol Properties*, colocar los siguientes datos como aparecen en la imagen *Fig.3.42*.

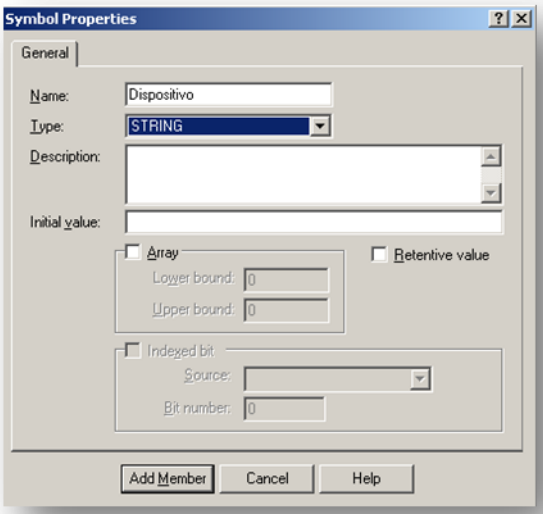


Fig.3.42 Creación de los elementos del Batch en Symbol Manager.

Después de agregarlos dar clic en *Add Member.*

Enseguida agregar los siguientes símbolos con los siguientes parámetros mediante el paso.

Tabla 3.4 Creación de los elementos de Batch.

NAME	TYPE	INITIAL_VALUE
Dispositivo	STRING	**FALSE**
Fecha	DT	**DT#AÑO-MES-DIA-00:00:00.000**
Fenómeno	**STRING**	**FALSE**

En el S*ymbol Manager*, dar clic sobre *Global*. Después dar clic en *New*  .

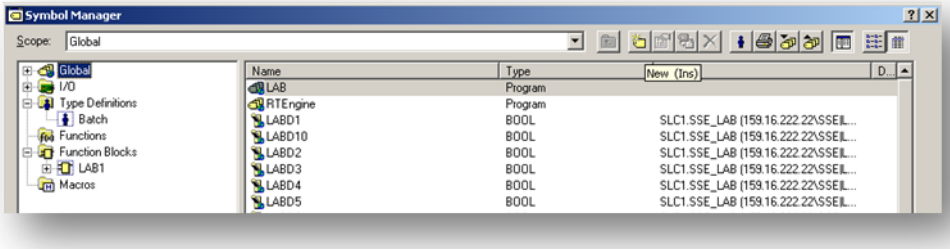


Fig.3.43 Vista de los elementos globales en Symbol Manager.

En la ventana de *Symbol Properties* coloque el nombre de *BatchInfo*: En la opción de *Type*, seleccionar *Type Definitions* y Seleccionar *Batch.*

Dar clic en *Add Global,* el cual nos permite declarar una variable como Global en el programa, y se pueden manejar estas variables en cualquier programa que se diseñe dentro de este proyecto.

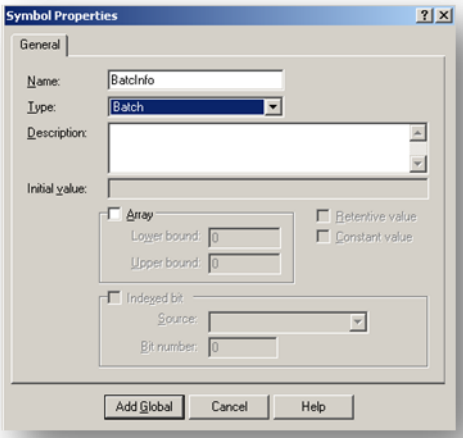


Fig.3.44 Creación del BachInfo.

Crearemos en el S*ymbol Manager*, dar clic sobre *Global* los archivos de disparo y cerrado.

Dar clic en New  En la ventana de *Symbol Properties* coloque el nombre de *ARCHIVO_DISPARO*.En la opción de *Type*, seleccionar *Type Definitions* y Seleccionar *FILE* Dar clic en *Add Global.*

Crear el símbolo *ARCHIVO_CERRADO* mediante los pasos anteriores, Dar clic en *Close* en el *Symbol Manager*.

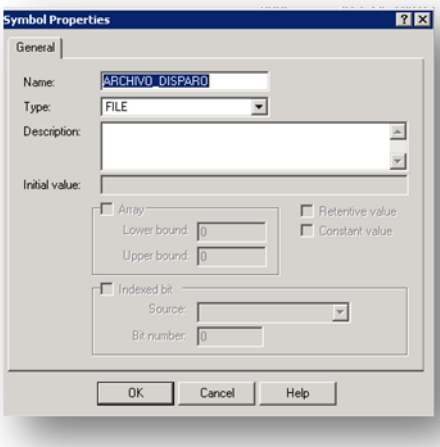


Fig.3.45 Creación del archivo disparo.

PROGRAMACION DEL FUNCTION BLOCK.

Programación de nuestras funciones a utilizar.

```
(*Programación del formato de nuestra estampa de tiempo que se utilizara para etiquetar el nombre de nuestro archivo *)

RES := DATE_TO_STRING(TIEMPO_INT);
(*  Convertimos el dato DT a String para manipularlo*)
RES := DELETE(IN:=RES, L:=1, P:=1);
(*  Eliminamos un elemento y lo recorremos a la posición 1*)
RES:= DELETE(IN:=RES, L:=1, P:=1);
RES:= DELETE(IN:=RES, L:=1, P:=1);
RES:= DELETE(IN:=RES, L:=1, P:=11);
RES:= INSERT(IN1:=RES, IN2:="_", P:=10);
(*Insertamos los elementos del formato que deseamos*)
RES:= DELETE(IN:=RES, L:=1, P:=14);
(*Eliminamos el elemento no deseado*)
RES:= INSERT(IN1:=RES, IN2:="-", P:=13);
RES:= DELETE(IN:=RES, L:=1, P:=17);
RES:= INSERT(IN1:=RES, IN2:="-", P:=16);
formato_DT_1 := RES;
(*Nuestra función nos arrojara una cadena RES*)

(*Programación del formato dentro de nuestro archivo de la estampa de tiempo que se utilizara para colocar el tiempo en el que ocurre un evento *)

RES:= DATE_TO_STRING(TIEMPO_INT);
(*  Convertimos el dato DT a String para manipularlo*)
RES:= DELETE(IN:=RES, L:=1, P:=1);
(*  Eliminamos un elemento y lo recorremos a la posición 1*)
RES:= DELETE(IN:=RES, L:=1, P:=1);
RES:= DELETE(IN:=RES, L:=1, P:=1);
RES:= DELETE(IN:=RES, L:=1, P:=11);
RES4:= MID(IN:=RES, L:=10, P:=1);
(*  Copiamos 10 caracteres desde la posición 1*)
RES1:= MID(IN:=RES4, L:=4, P:=1);
RES2:= MID(IN:=RES4, L:=2, P:=6);
RES3:= MID(IN:=RES4, L:=2, P:=9);
RES4:= REPLACE(IN1:=RES4, IN2:=RES3, L:=2, P:=1);
(*Reemplazamos los elementos creados en nuestro formato prediseñado RES4*)
RES4:= REPLACE(IN1:=RES4, IN2:="/", L:=1, P:=3);
```

```
RES4:= REPLACE(IN1:=RES4, IN2:=RES2, L:=2, P:=4);
RES4:= REPLACE(IN1:=RES4, IN2:="/", L:=1, P:=6);
RES4:= REPLACE(IN1:=RES4, IN2:=RES1, L:=4, P:=7);
RES:= REPLACE(IN1:=RES, IN2:=RES4, L:=10, P:=1);
RES:= INSERT(IN1:=RES, IN2:=" ", P:=10);
formato_DT_2:= RES;
(*Nuestra función nos arrojara una cadena RES*)

(*Programación del formato de nuestra ruta de dirección en donde se crearan los archivos*)

RES:= INSERT(IN1:= "C:\INCONTROL\MENSAJES\DEPOSITO\ () .txt",
IN2:=EVENTO(* ENTRADA *), P:=31);
(*Esta será nuestra ruta de dirección de los archivos, que además estará constituido por el EVENTO, el interruptor y la estampa de tiempo*)
RES:= INSERT(IN1:=RES, IN2:=INTERRUPTOR_NUM(* ENTRADA *), P:=40);
RES:= INSERT(IN1:=RES, IN2:=EW(* ENTRADA *), P:=51);
direccion:=RES;
(*Nuestra función nos arrojara una cadena RES*)
```

Temporizador acumulativo o de retención

Este tipo de temporizador requiere de dos entradas. Una de las entradas inicia la temporización y la otra la restaura a cero. La temporización de los mencionados anteriormente es restaurada a cero una vez que la entrada del censor que los activa cambia de estado sin que haya concluido la temporización, mientras que este tipo de temporizador mantiene el tiempo de temporización que haya transcurrido cuando el mismo sea desactivado a mitad del ciclo de temporización. Por ejemplo, si se desea conocer cuanto tiempo estuvo un censor activado durante el intervalo de una hora, hay que usar temporizador acumulativo ya que si se usan los ordinarios (On / off delay) el temporizador que lleva la cuenta del tiempo se mantendría reseteado cada vez que el censor se desactive / active. Un símbolo para este tipo de temporizador es RTO (retentive timer) o TMRA (accumulating timer).

TMR (EN :=(BOOL), PT :=(TIME), ET :=(TIME));

EN= Es la entrada del bloque, al activarse en 1 lógico comienza a funcionar el bloque cuya variable ya están definidas.
PT= Es el tiempo que se programa que funcione el bloque.
ET= Es el tiempo transcurrido en tiempo real del bloque, en la cual están ya asignadas las variables.

```
(*Inicio del programa*)
PROTECC(EN:=(ENTRADA),PT:=(TIEMPO_RESETEO_DEL_TEMPORIZADOR
_PROTECCION),ET:=(TIEMPO_TRANSCURRIDO_PROTECCION));
```

```
TIEMPO_RESETEO_DEL_TEMPORIZADOR_PROTECCION:=Time#365d;
(* 365 días es el tiempo de reset de la protección*)

INTER:=INTERRUPTOR_NUM;
(*La variable INTER tomara el valor String de la entrada INTERRUPTOR *)
salida1_cerrado:=0;
(*Condición inicial para evitar el envío de mensajes CERRADO cada vez que el programa se ejecute por primera vez*)
F_TRIG001(CLK:=INTERRUPTOR);salida_disparo:=F_TRIG001.Q;
(*Detecta el flanco descendente de la protección: condición para el disparo del interruptor*)
R_TRIG001(CLK:=INTERRUPTOR);salida_cerrado:=R_TRIG001.Q;
(*Detecta el flanco ascendente de la protección: condición para el cerrado del interruptor*)

IF salida_disparo=1 THEN
salida1_disparo:=1;
(* Dado que el detector de flanco es un pulso asignamos una variable para que quede constante hasta el reset *)
END_IF;

IF salida_cerrado=1 THEN
(* Dado que el detector de flanco es un pulso asignamos una variable para que quede constante hasta el reset *)
salida1_cerrado:=1;
END_IF;

IF (( (TIEMPO_TRANSCURRIDO_PROTECCION >T#0s AND TIEMPO_TRANSCURRIDO_PROTECCION<=T#1m) OR ( PROT=1))) and ( salida_disparo=1 OR salida_cerrado=1) AND CUCH=1 THEN
(*  Se cumplirá nuestra lógica si existe una protección un minuto antes de que ocurra un cambio en el interruptor o cuando la protección este continua,  un cambio en el interruptor y las cuchillas estén activadas*)

(*Condición para la creación del archivo de disparo*)
IF  ( salida1_disparo=1) THEN
```

```
(* Si se activa la entrada disparo se llevara a cabo la creación del archivo para
este *)
EW := formato_DT_1(TIEMPO_INTERRUPTOR);
(*Función para el formato de la etiqueta de tiempo del interruptor*)
RH := formato_DT_2(TIEMPO_INTERRUPTOR);
(*Función para el formato de la etiqueta de tiempo del interruptor*)
DR:= direccion(EVENTO:=EVENTO,
INTERRUPTOR_NUM:=INTERRUPTOR_NUM, EW:=EW);
(*Función para el formato de dirección*)

NEWFILE(FCB:=ARCHIVO_DISPARO, FILE:=DR);
(*Creación del archivo esta función se encuentra en la barra de herramientas del
Function Block*)
(* Creamos el archivo con la ruta DR que en este caso construimos con algunas
combinaciones strings *)

(*UTILIZACION DEL BatchInfo*)
BatchInfo.FECHA := RH ;
(* Asignamos la cadena RH al batchinfo que construimos *)
BatchInfo.FENOMENO:=INTER;
(* Asignamos la cadena de la entrada INTER al batchinfo que construimos *)
BatchInfo.DISPOSITIVO:=DISPARO;

WRITEFILE(FCB:=ARCHIVO_DISPARO, IN:=BatchInfo,F:=' ', S:='', T:='');(*
ESCRIBIMOS LOS VALORES DE BATCHINFO *)
(*Escribimos los elementos que constituyen al BachInfo*)
WRITEFILE(FCB:=ARCHIVO_DISPARO, IN:="***INFORMA SAD400
INCONTROL***",F:='', S:='', T:='$N');
(*Escribimos la leyenda de nuestro informador*)
WRITEFILE(FCB:=ARCHIVO_DISPARO, IN:="9611288041",F:='', S:='', T:='');
(*Escribimos el teléfono al que va dirigido*)
CLOSEFILE(ARCHIVO_DISPARO);
(*Cerramos nuestro archivo creado por DR*)
salida1_disparo:=0;
(*Ponemos en estado inicial nuestras condiciones*)
END_IF;
(*Cerramos nuestra condición*)
```

```
(*Condición para la creación del archivo de cerrado*)
(*Debido a que el procedimiento es el mismo solo se cambiara DISPARO por CERRADO en BatchInfo.DISPOSITIVO *)
IF ( salida1_cerrado=1) THEN
EW := formato_DT_1(TIEMPO_INTERRUPTOR);
RH := formato_DT_2(TIEMPO_INTERRUPTOR);
DR:= direccion(EVENTO:=EVENTO, INTERRUPTOR_NUM:=INTERRUPTOR_NUM, EW:=EW);
NEWFILE(FCB:=ARCHIVO_CERRADO, FILE:=DR);
BatchInfo.FECHA := RH ;
BatchInfo.FENOMENO:=INTER;
BatchInfo.DISPOSITIVO:=CERRADO;
WRITEFILE(FCB:=ARCHIVO_CERRADO, IN:=BatchInfo,F:=' ', S:='', T:='');
WRITEFILE(FCB:=ARCHIVO_CERRADO, IN:="***INFORMA SAD400 INCONTROL***",F:='', S:='', T:='$N');
WRITEFILE(FCB:=ARCHIVO_CERRADO, IN:="9611288041",F:='', S:='', T:='');
CLOSEFILE(ARCHIVO_CERRADO);
 END_IF;
END_IF;
```

Creados nuestros Function Block y nuestras funciones en nuestra barra de herramientas de RLL tenemos de vemos de ir a P*rogram/PROGRAMA_ESCALERA*

Fig.3.46 Localización de nuestro programa.

Aparecerá la ventana de programación RLL en la parte inferior del programa en *User-Defined* damos click y después en *ST_WONDERWARE_prot* damos click.

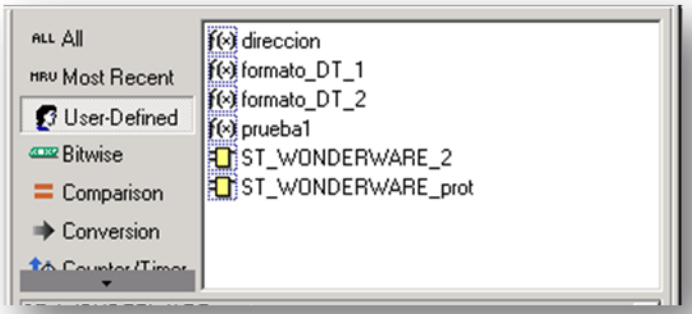


Fig.3.47 Funciones creadas.

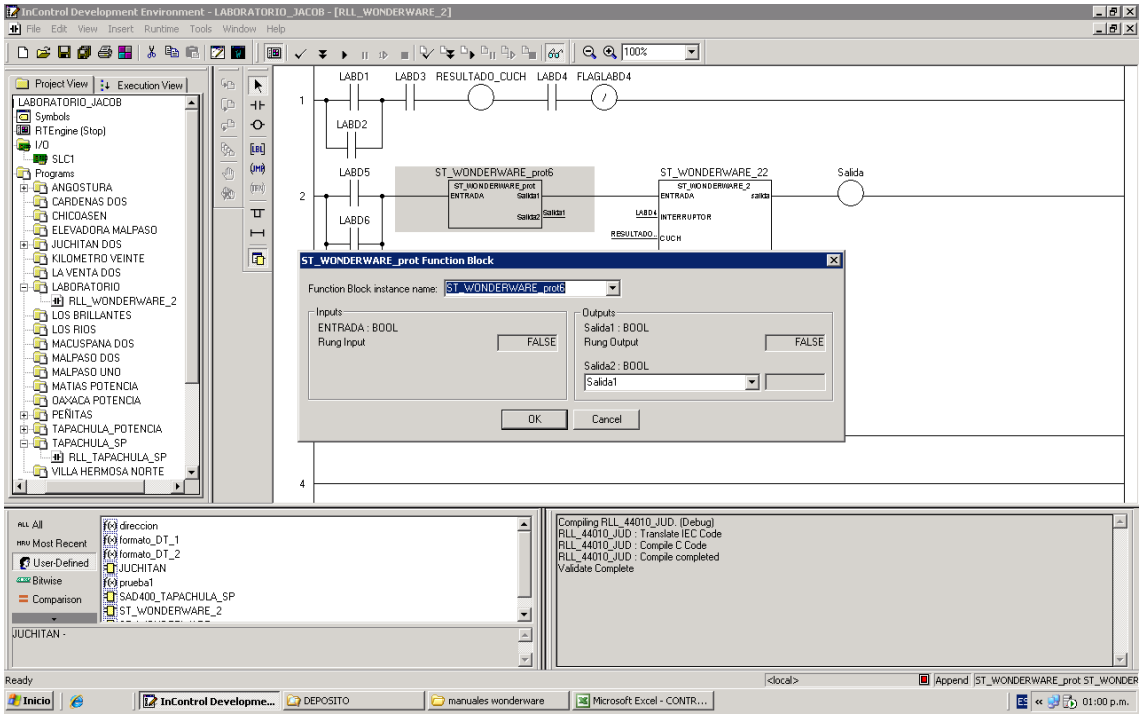


Fig.3.48 Ingresar el Function Block de la protección.

Hacemos lo mismo con nuestro programa principal.

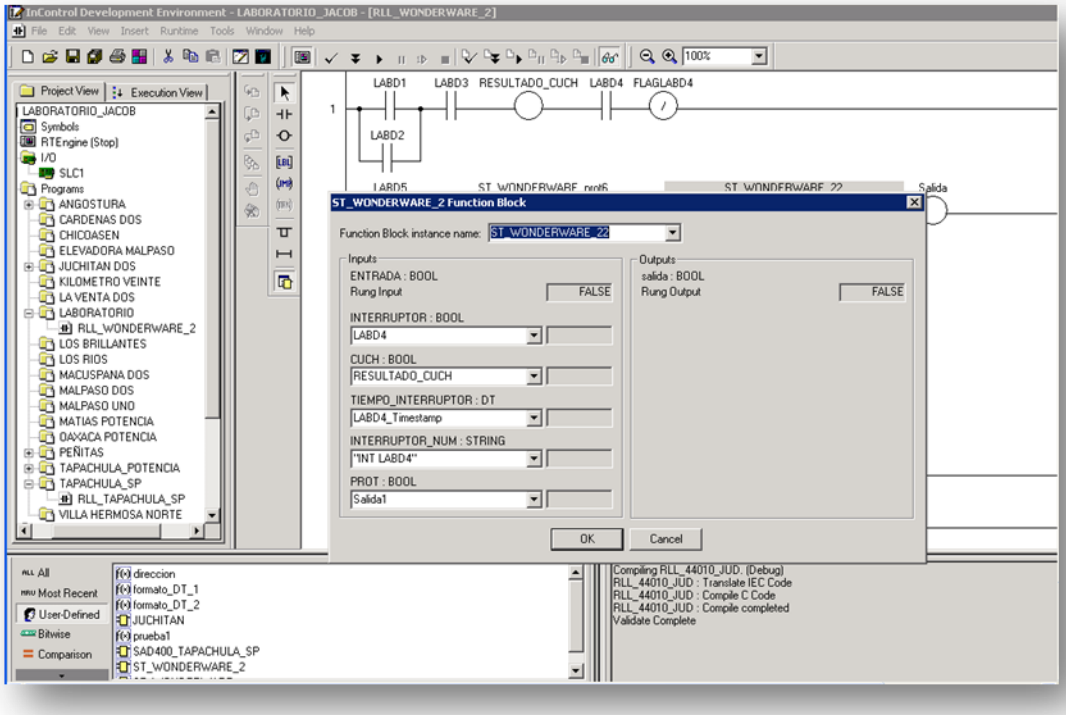


Fig.3.49 Ingresar el Function Block de la creación del archivo.

Compilación del programa en *RTEngine*.

-En la pantalla principal de *InControl* dar clic en el botón de *Connect Runtime Engine*  el cual se localiza en la parte superior de la pantalla. Aparecerá una barra de botones. Dar clic en *Run Project*  . El programa comenzará a correr en tiempo real:

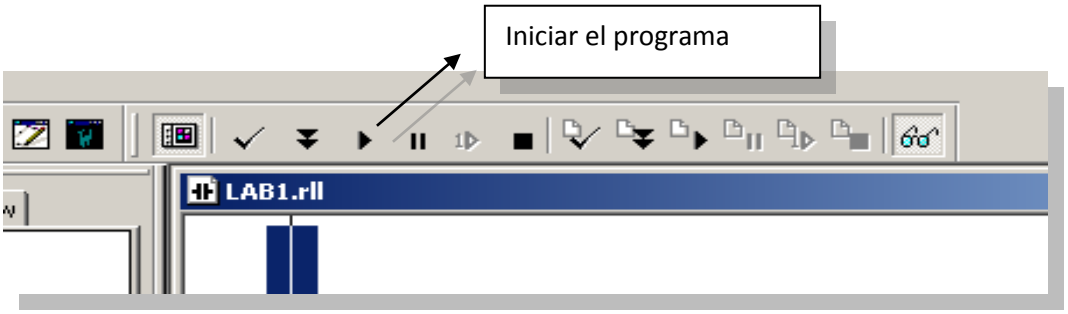


Fig.3.50 Ejecución del programa.

Aparecerá un cuadro de diálogo llamado *Run Project*, seleccionar *Full Restart* y después *Ok (Figura 3.57)*.

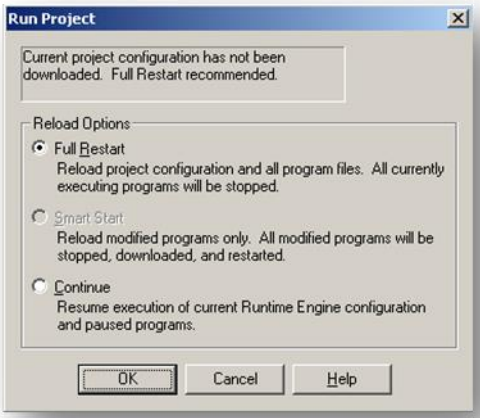


Fig.3.51 Opciones de ejecución del programa.

-El programa comenzará a correr. Observar que en la parte inferior de *InControl* aparecerá una leyenda donde nos está indicando que la ejecución del diagrama que se dibujó o programó en las cuerdas está cargándose y que también nos dice que ya está corriendo el programa:

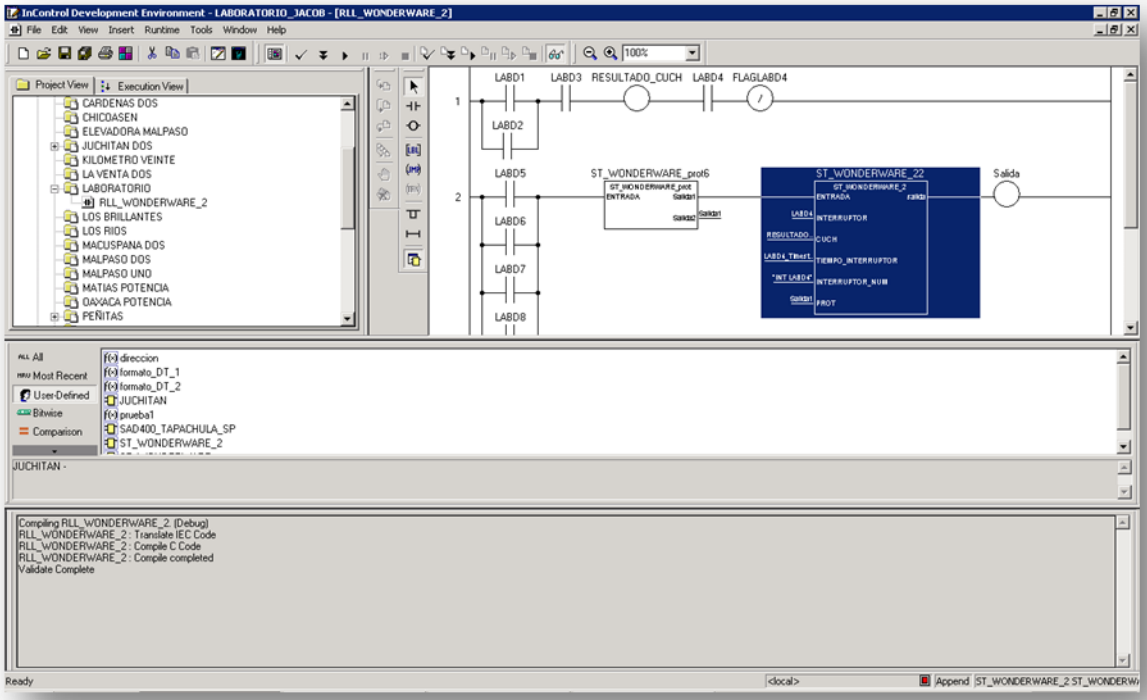


Fig.3.52 Barra de estado: validación y ejecución.

3.7 Configuración de los archivos .BAT en System32 al Modem Telcel para envió de mensajes GSM.

-Abrir el siguiente directorio: *C:\PROYECTOS\INCONTROL\LABORATORIO_JACOB.*

Crear 2 archivos .BAT: *sensegood.bat* y *mastergood.bat*.

sensegood.bat:

En el directorio actúa como censor, el cual esta censando los archivos en .bat que van apareciendo cuando exista un disparo y los manda al archivo mastergood.bat

mastergood.bat:

Permite mandar la información del archivo DISPARO.bat o CERRADO.bat al MODEM de telcel como mensaje de texto.

-Colocar el siguiente código en el archivo *sensegood.bat*:

```
Echo off
if exist DISPARO.bat (
```

```
        echo Enviando Mensaje .....
        call env0.bat
        rem sms_sad 9616524086;el archivo existe;sad400
        echo.
        del env.bat
        echo El mensaje fue enviado con exito.
        ) ELSE  (
        ECHO .
        ECHO NO EXISTE EL ARCHIVO en espera de su
 llegada al HD
        ECHO .
        ECHO .
 )
Echo off
if exist CERRADO.bat (
        echo Enviando Mensaje .....
        call env0.bat
        rem sms_sad 9616524086;el archivo existe;sad400
        echo.
        del env.bat
        echo El mensaje fue enviado con exito.
        ) ELSE  (
        ECHO .
        ECHO NO EXISTE EL ARCHIVO en espera de su
 llegada al HD
        ECHO .
        ECHO . )
```

Si existe en la carpeta el archivo DISPARO.bat o CERRADO.bat mandarlo al archivo mastergood.bat con la leyenda que fue enviado con éxito.

Si no se encuentra el archivo DISPARO.bat

O CERRADO.bat Mandarla al archivo mastergood.bat con la leyenda: que no existe el archivo en espera de su llegada al HD

-Colocar en el archivo *mastergood.bat* el siguiente código:

```
echo off
set cont=1
```

MANDAR TODA LA INFORMACION DEL ARCHIVO sensegood.bat a un archivo ejecutable llamado: sms_sad.exe, el cual se encargara de mandarla al MODEM de telcel y así generar el mensaje de texto con la información de Disparo.

```
set var=0

set inicio=0

:primera

 if %cont% == 1 (

      echo sensando....

      call sensegood.bat

      goto primera ) else (

      SET cont=0)

:segunda

:tercera
```

Insertar el archivo *sms_sad.exe* el cual permitirá mandar los archivos *DISPARO.bat o CERRADO.bat* al Modem de Telcel y así generar los mensajes *GSM* de alarma generados en el programa de *InControl.*

se hicieron pruebas en el laboratorio creándose los archivos a enviarse como se muestra en la siguiente figura.

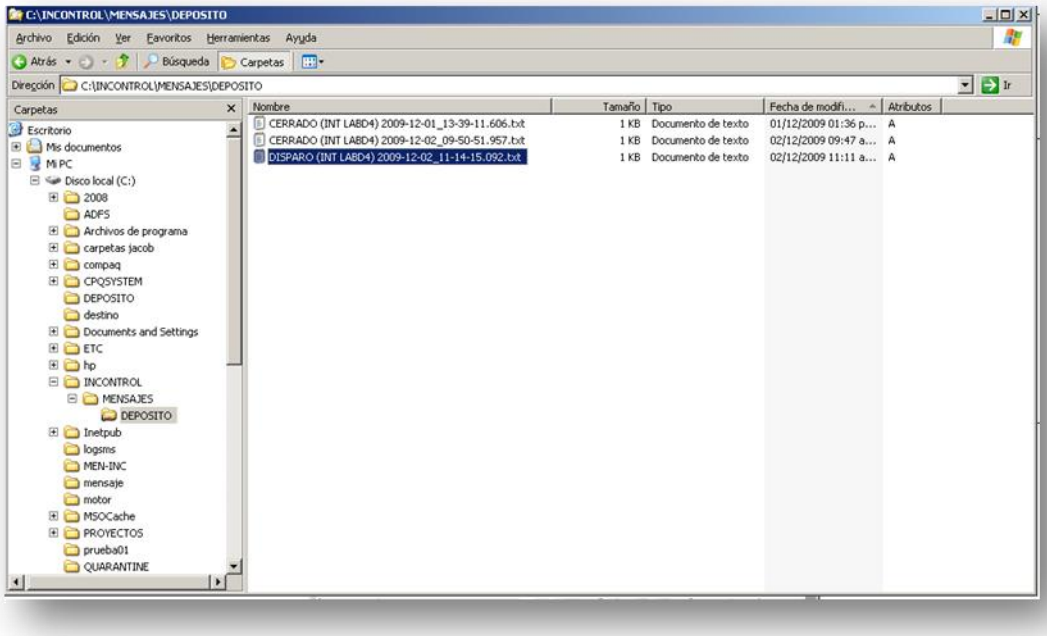


Fig.3.53 creación del archivo.

3.8 Red Troncal de las subestaciones zona sureste.

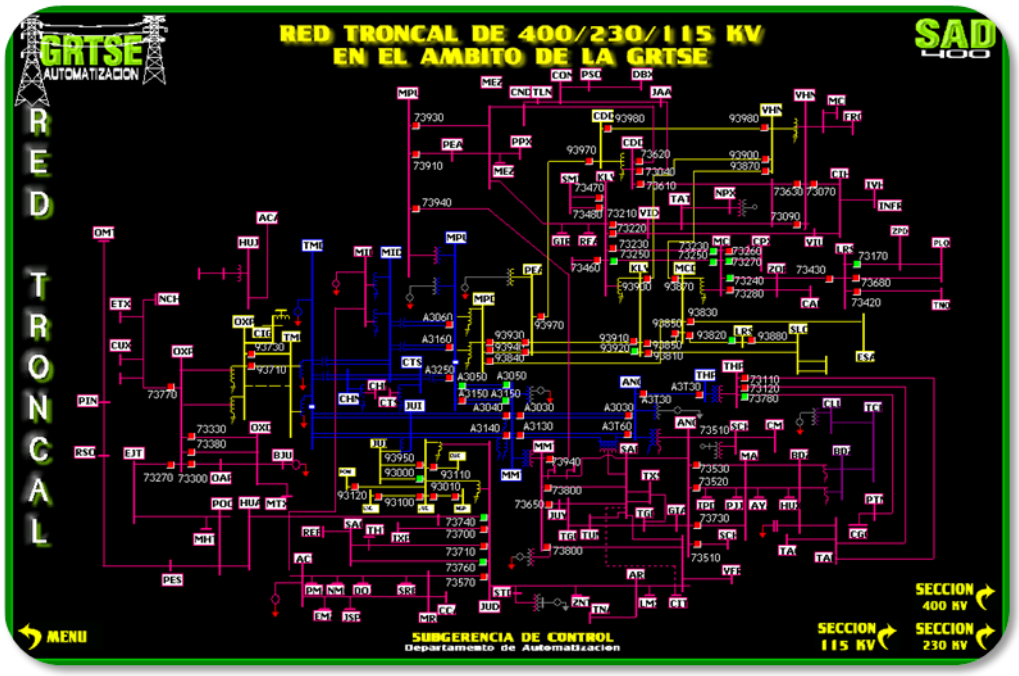


Fig.3.54 Red Troncal de las subestaciones zona sureste

3.9 Unifilar de la subestación de Angostura.

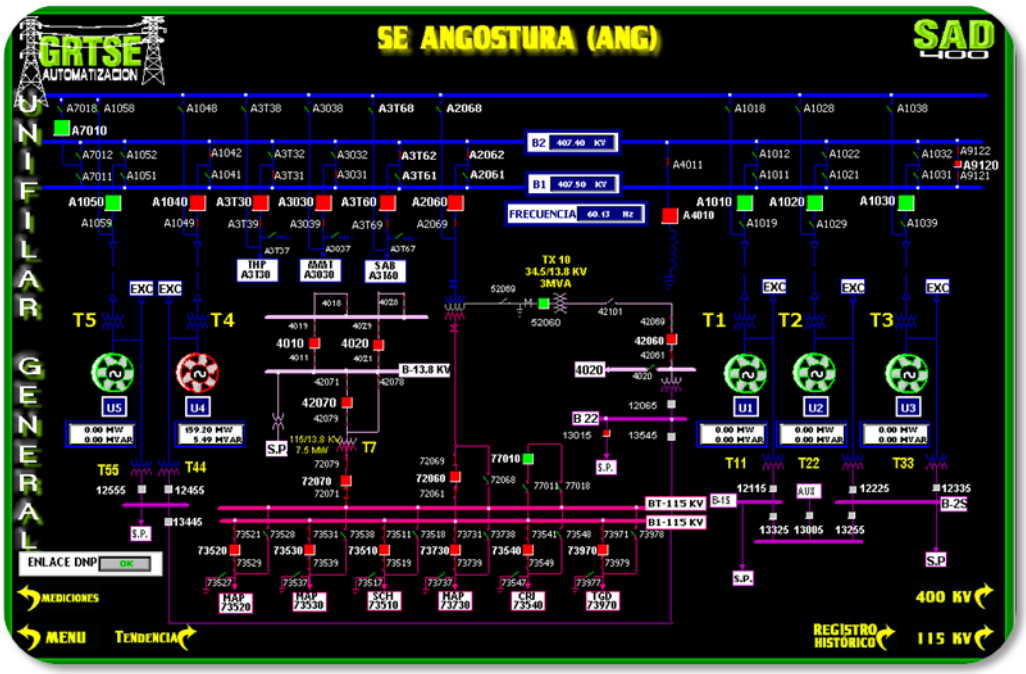


Fig.3.55 Diagrama unifilar de Angostura.

3.10 Interruptores y cuchillas de Angostura.

Fig.3.56 Estados de interruptores y cuchillas.

3.11 Comportamiento del interruptor gráficamente.

Fig.3.56 Comportamiento del interruptor gráficamente.

3.12 Programación de la lógica de escalera con respecto a los dispositivos de la bahía de la subestación de Angostura.

Debido a la similitud que existe entre las lógicas de las bahías (en este caso solo cambia las cuchillas e interruptores: que no es de gran importancia debido a nuestra programación ya que se toma como entrada el conjunto de las cuchillas como *REULTADO_CUCHILLA* y en las protecciones la salida de nuestro *Function Block llamado ST_Wonderware_prot*) tomaremos como ejemplo el interruptor A3T30 de la subestación de Angostura uno de los más importantes.

Create Topic en SLC1.

Creamos nuestras Tags de entradas asignándole la misma dirección IP que es nuestro nodo estos datos son proporcionados por el jefe del departamento de Automatización

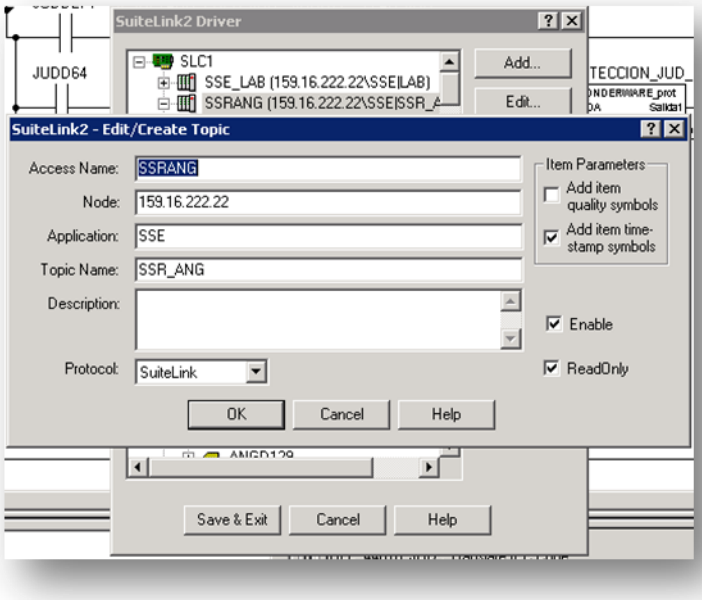


Fig.3.56 Create Topic en SLC1.

Ingresar las I/O en el SuitLink.

Las Tags se encuentran en una base de datos del departamento de automatización.

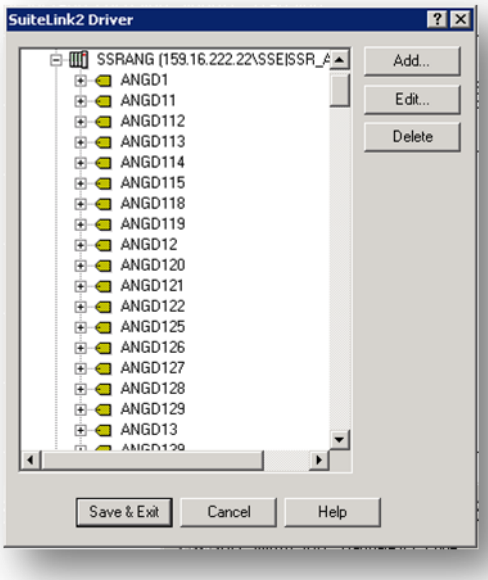


Fig.3.57 Ingresar las I/O en el SuitLink.

Al igual que en nuestro laboratorio creamos nuestro programa RLL y arrastramos los Function Blocks.

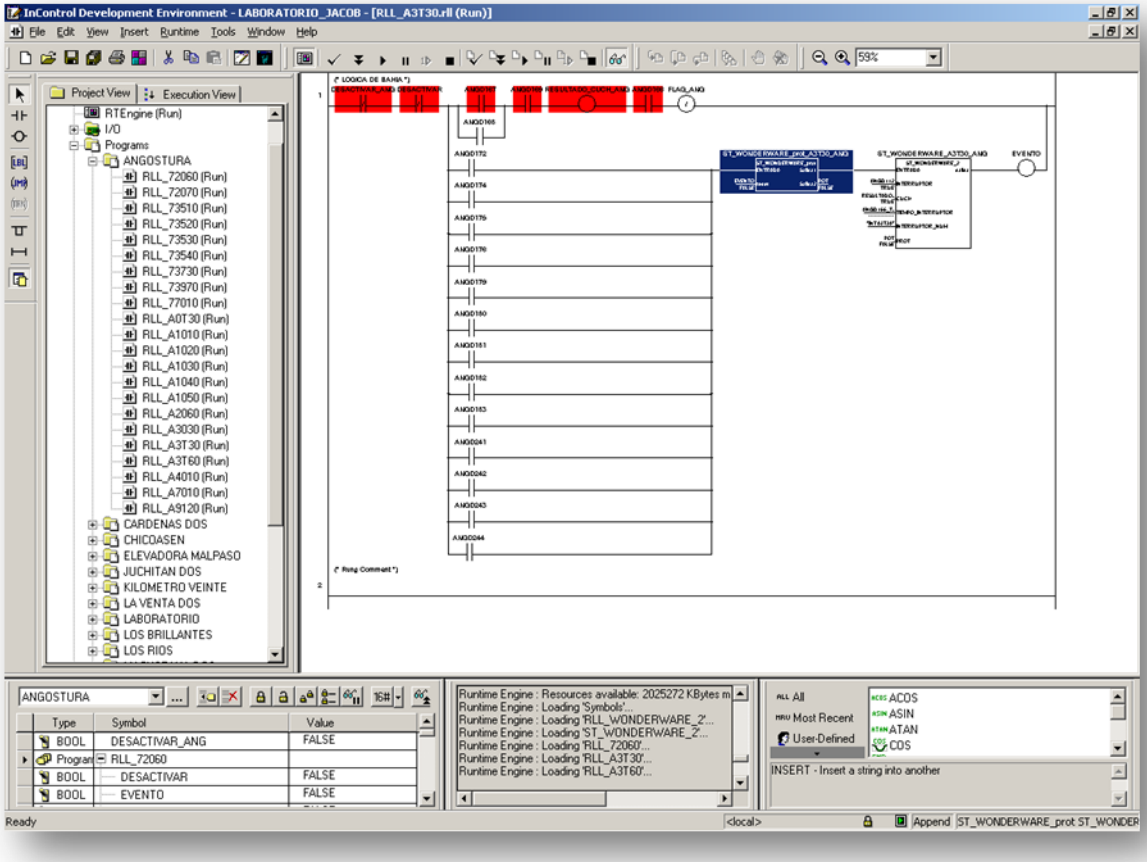


Fig.3.58 Lógica en RLL del interruptor A3T30.

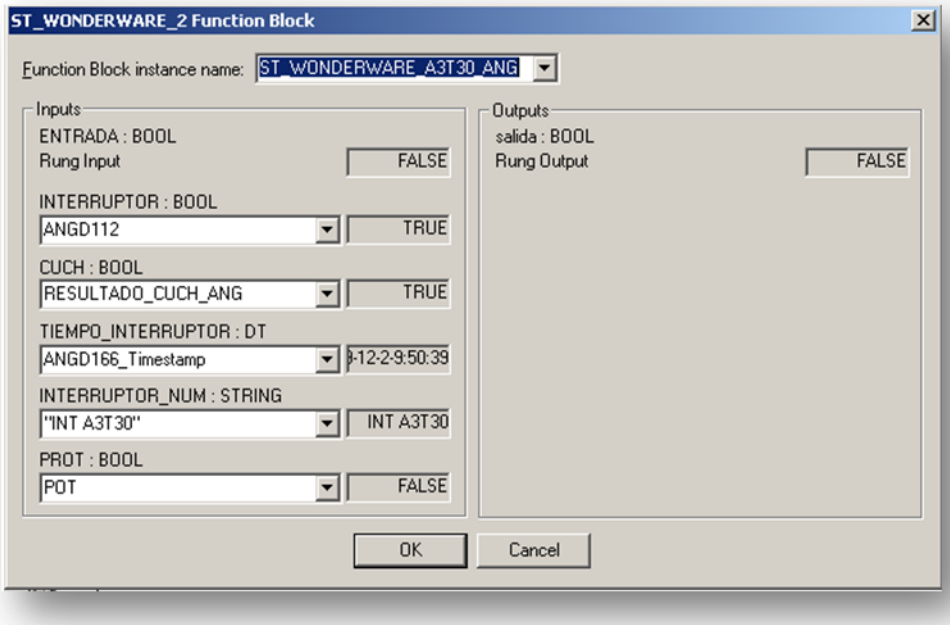


Fig.3.59 Function Block del interruptor A3T30.

En la figura 3.59 se observa los datos asignados asen referencia a los obtenidos en campo como es la tag, el tiempo del evento y el interruptor.

Creamos una ventana de control para cada uno de las subestaciones en donde se puede observar el comportamiento de cada elementos.

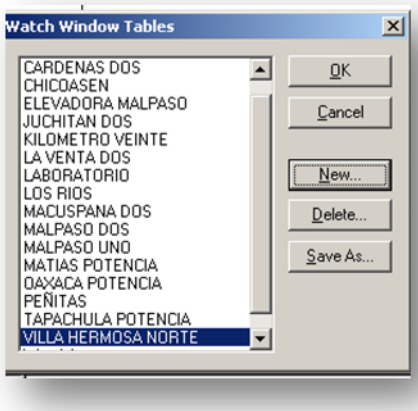


Fig.3.60 Crear Watch Windows Table.

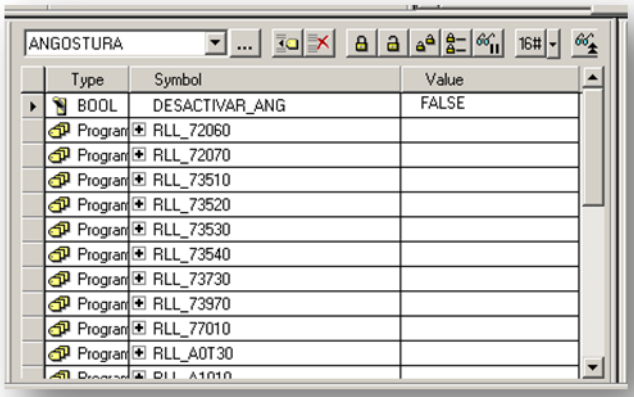


Fig.3.60 Watch Windows Table.

En esta ventana podemos activar y desactivar nuestro sistema por bahía o por sub estación como se muestra en las siguientes figuras.

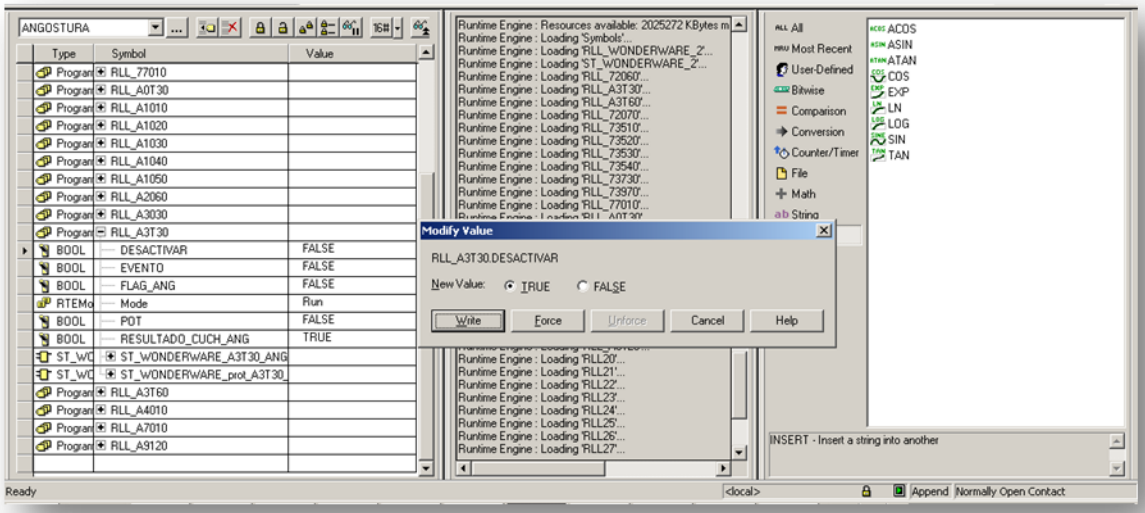


Fig.3.61 Desactivación por bahía.

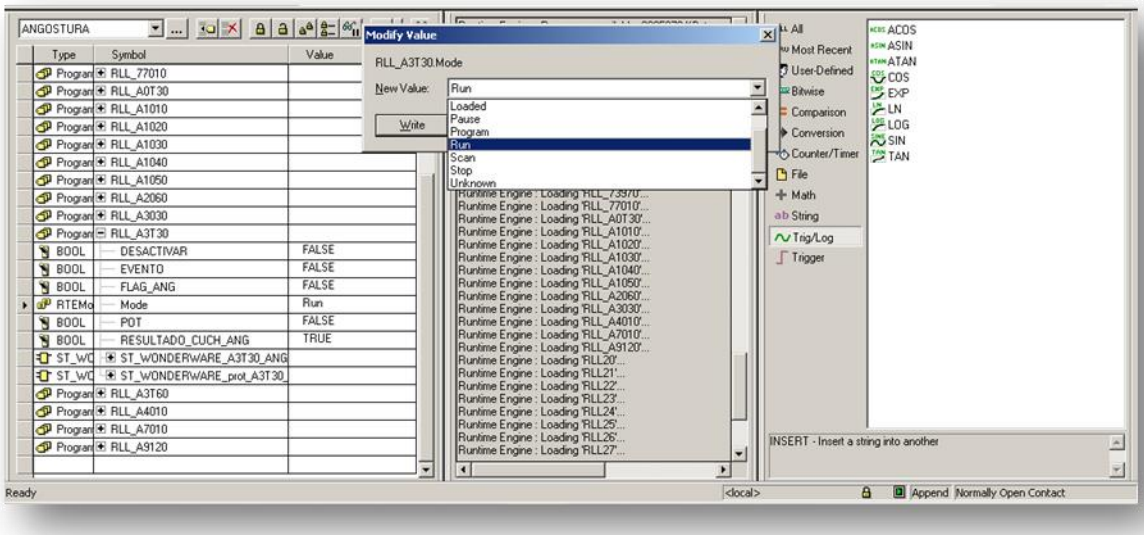


Fig.3.62 Desactivación por modo(inhabilita): por bahía o programa.

Nuestro Funtion Block trabaja en tiempo real al igual que los elementos con un *Cache* cada 100milisegundos, también es modificable.

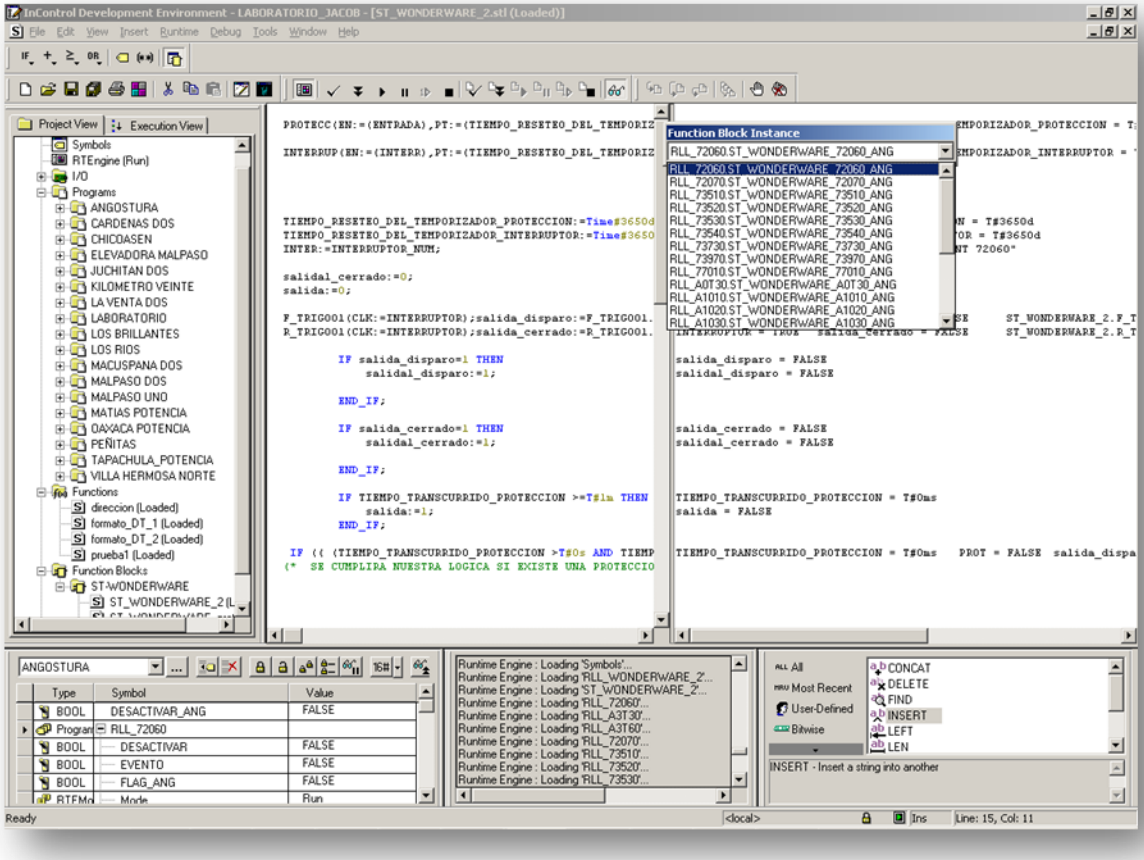


Fig.3.63 Function Block Instance.

En la figura 3.arriba se observa la ventaja de utilizar los *Function Block* ya que al utilizar funciones solo se mandan a llamar a todos los interruptores como se muestra en el cuadro *Function Block Instance.*

Observaciones y sugerencias.

Observaciones.

La empresa internacional Comisión Federal de Electricidad otorga a los estudiantes de aplicaciones tecnológicas la facilidad de implementar sus conocimientos en las tecnologías de punta con las que cuenta.

En la actualidad el instituto Tecnológico de Tuxtla Gutiérrez no cuenta con la tecnología actual.

Sugerencias.

Como instituto tecnológico debe de promover y orientarse a la tecnología actual.

Conclusiones.

Gracias al proyecto realizado se preverá las fallas de las subestaciones teniendo como prioridad informar a las personas adecuadas para las tomas de decisiones. Esto nos lleva a concluir que el proyecto es 100% viable debido al capital que ya no se invertirá en la prevención de dichos eventos asumiendo un gran impacto benéfico para la empresa ya que también se podrá monitorear y almacenar en una base de datos benéficos para el control histórico de la empresa.

Referencias.

http://global.wonderware.com/EN/Pages/default.aspx

http://global.wonderware.com/EN/Pages/WonderwareInControlSoftware.aspx

http://www.encoweb.com/?q=forum/16

http://fzayas.com/con-textos/index.php/La_estructura_del_texto

http://www.grupo-maser.com/PAG_Cursos/Step/step7/Proyecto%20step7/paginas/contenido/step7/2/2.3.6.htm

http://es.wikipedia.org/wiki/Protocolo_de_comunicaciones

http://es.wikipedia.org/wiki/Etiqueta_(lenguaje_de_marcado)

http://global.wonderware.com/EN/Pages/default.aspx

Anexos.

Sistema de comunicación de control en las subestaciones.

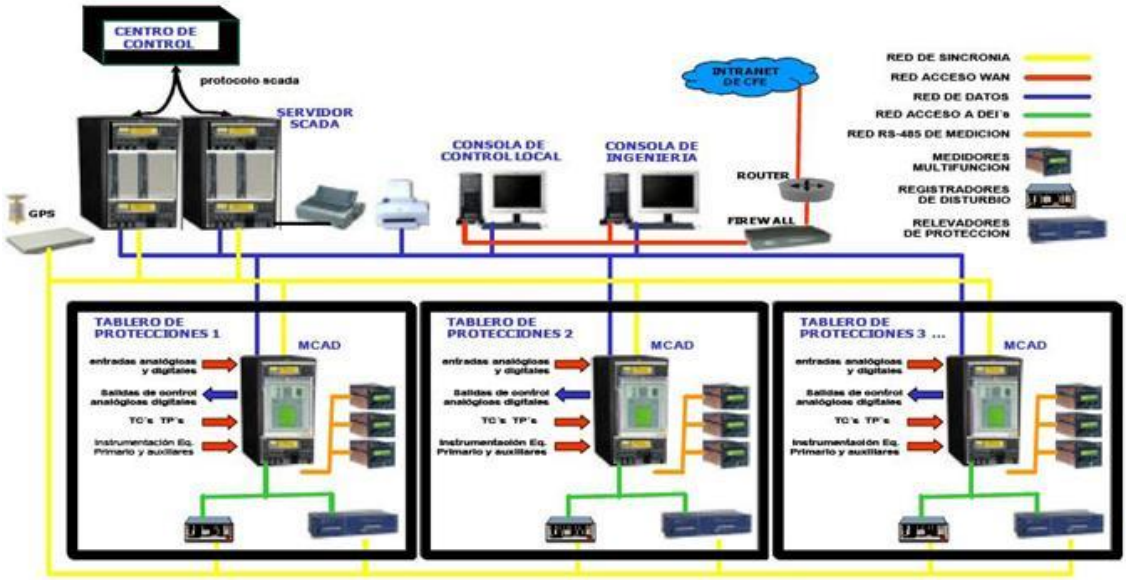


Fig. Anexo Sistema de comunicación de control en las subestaciones.

Comunicación de la subárea al SAD400.

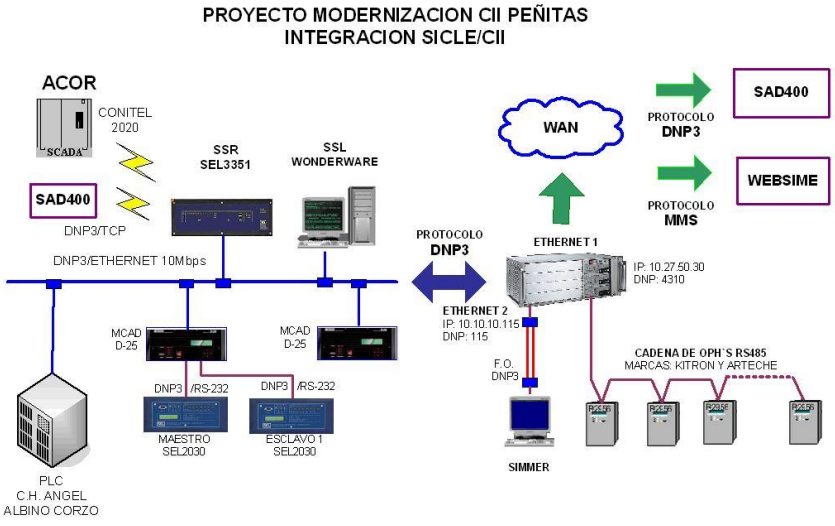


Fig. Anexo Comunicación de la subárea al SAD400.

CFE Una empresa de clase mundial **COMISIÓN FEDERAL DE ELECTRICIDAD**

GERENCIA REGIONAL DE TRANSMISIÓN SURESTE

Oficio No. SPR-040/09
Tuxtla Gutiérrez, Chiapas
18 de Diciembre del 2009

DR. DANIEL SAMAYOA PENAGOS
JEFE DEL DEPTO. DE GESTION TECNOLOGICA Y VINCULACION
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIERREZ
P R E S E N T E

93
CONCLUSIÓN
RESIDENCIA PROFESIONAL

Por medio del presente comunico a usted que el alumno(a):
JACOB GOMEZ JOSE
de la carrera **INGENIERIA ELECTRONICA**
No. de control **05270271** concluyó satisfactoriamente su **RESIDENCIA PROFESIONAL**
en esta empresa, Comisión Federal de Electricidad, Cubriendo un total de **640** hrs.
asignado en la **GRTSE.- DEPTO. AUTOMATIZACION**
fecha de inicio **24 Agosto 2009** fecha de termino **18 Diciembre 2009**
observando durante el desempeño de sus actividades, un alto sentido de responsabilidad, disciplina y buena conducta.

RECIBIDO
18 DIC 2009
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ
Oficina de Prácticas y Promoción Profesional

Sin otro particular de momento, quedo de usted.

ATENTAMENTE

ING. MANUEL CASTILLO VAZQUEZ
Jefe de Ofna. de Capacitación y Seguridad Industrial

C.c.p.- C.P. Agustín G. Ballinas Moreno.- Jefe Depto. Recursos Humanos GRTSE.
C.c.p.- C.P. Luís Alfredo Acuña Araujo.- Secretario Gral. de la Secc. 47 SUTERM.

C.c.p.- Interesado.
C.c.p.- EXPEDIENTE
*JAMG-MCV-FCV

Carretera Panamericana No. 5675 Int. 500 MST Tuxtla Gutiérrez, Chiapas Tel. y Fax. (961) 617-36-24 M.O. 3555 al 58 Fax 3589