

UNIVERSIDAD AUTÓNOMA DE QUERETARO

FACULTAD DE INGENIERIA

TESINA

Ingeniería en Automatización

CONFIGURACIÓN DE UNA RED DEVICENET CON HMI EN LABVIEW MEDIANTE
LABSOCKET

Que presenta alumno:

Omar Pablo Pérez Guevara

Asesor:

Dr. Juvenal Rodríguez Reséndiz

Agosto del 2014

La presente obra está bajo la licencia:
<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:



Atribución — Usted debe dar [crédito de manera adecuada](#), brindar un enlace a la licencia, e [indicar si se han realizado cambios](#). Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.



NoComercial — Usted no puede hacer uso del material con [propósitos comerciales](#).



SinDerivadas — Si [remezcla, transforma o crea a partir](#) del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni [medidas tecnológicas](#) que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una [excepción o limitación](#) aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como [publicidad, privacidad, o derechos morales](#) pueden limitar la forma en que utilice el material.

I. DATOS GENERALES

Opción de Titulación: Tesina

Título del Trabajo: Configuración de una red DeviceNet con HMI en LabVIEW mediante Labsocket

Nombre del alumno: Omar Pablo Pérez Guevara

Expediente: 188939

Nombre de la licenciatura: Ingeniería en Automatización

Nombre del Asesor: Dr. Juvenal Rodríguez Reséndiz

Teléfono Celular: 4423796540

INDICE

▪ Resumen	I
▪ Justificación	II
▪ Objetivo	III
▪ Antecedentes	IV
▪ Introducción	V
Capítulo 1 Sistemas SCADA	13
1.1 Introducción a los sistemas de supervisión	13
1.2 SCADA	13
1.3 Elementos de un sistema SCADA	16
Capítulo 2 Introducción a DeviceNet	17
2.1 Conceptos básicos	17
2.1.1 Características de la red	18
2.2 Buses de Campo	19
2.2.1 Ventajas de los buses de campo	20
2.3 Protocolo DeviceNet	21
2.3.1 El modelo de Objeto	23
2.3.2 La capa de Enlace de Datos (Data link layer)	23
2.3.3 Modos de comunicación	24
2.3.4 Normas Internacionales	25
2.3.5 El protocolo CIP (Common Industrial Protocol)	25
2.3.6 Protocolo de comunicación CAN	26
2.3.7 Características de CAN	27
2.4 Modelo OSI	28
2.4.1 Modelo OSI para las comunicaciones Industriales	29
2.5 Adaptación del Protocolo DeviceNet al modelo OSI	30

2.6 Capa física del protocolo DeviceNet	31
2.6.1 Topología de red del Protocolo DeviceNet	32
2.6.2 Terminaciones de red	33
2.6.3 Conectores Derivadores para DeviceNet	33
2.6.4 Suministro de alimentación	35
2.7 Capa de enlace del protocolo DeviceNet	35
2.7.1 CSMA/CD	35
2.7.2 Comunicación Broadcast	35
2.7.3 Trama de datos	36
2.7.4 Transmisión de Mensajes	36
2.8 Capa de aplicación del protocolo DeviceNet	37
Capítulo 3 Arquitectura de la red	37
3.1 Configuración de la red y ajustes principales de RSNetWorx para DeviceNet	37
3.2 Armado de la Red	38
3.3 Configuración de Entradas y salidas en el RSlogix 500	44
3.4 RSLogix 500	45
3.5 Configuración del módulo escáner 1769-SDN	46
3.6 Comunicación entre el PLC's	48
3.7 Mapeo de la Red	49
3.8 Configuración de módulos IO	50
Capítulo 4 Introducción a LabVIEW	51
4.1 conceptos Básicos	53
4.2 Poder de la Programación Gráfica	54
4.3 HMI	56
4.4 Estructura del software labVIEW	57
4.5 Protocolos de comunicación	59

4.5.1 Intercambio Dinámico de Datos (DDE)	59
4.5.2 Objetivo Vinculado e incrustado (OLE)	59
4.5.3 OLE Automation	60
4.5.4 OLE para control de proceso (OPC)	61
Capítulo 5 LabSocket	62
5.1 Introducción	62
5.2 Teoría de Operación	63
5.2.1 Cliente Mapping Target VI	65
5.2.2 Servidor LabSocket	65
5.2.3 MultiClient LabSocket	66
5.2.4 Funcionalidad del MultiClient	66
5.3 Requisitos del Sistema	67
5.4 Instalación del Software labSocket-Basic	68
5.5 Demostración de Aplicación de Labsocket	68
5.6 Comunicación entre instancias de servidor VI y Target VI	75
5.7 Aplicación del Labsocket	75
• Conclusión	79
• Bibliografía	80

ÌNDICE DE IMÁGENES

Capítulo 1 Sistemas SCADA	13
Fig. 1.1 Partes de un sistema SCADA conectado a un proceso automatizado.	15
Capítulo 2 Introducción a DeviceNet	17
Fig. 2.1 Ejemplo de topología de la red DeviceNet	18
Fig. 2.2 El modelo OSI de los objetos del CIP	22
Fig.2.3 Formato de la estructura de datos.	23

Fig.2.4 Sistema de control distribuido.	28
Fig.2.5 Modelo OSI y adaptación del protocolo DeviceNet al modelo OSI.	30
Fig. 2.6 Topología de DeviceNet	32
Fig. 2.7 Esquema de colores de los cables DeviceNet	32
Fig. 2.8 Conexión de los resistores de terminación.	33
Fig. 2.9 Formato de la trama de datos	36
Capítulo 3 Arquitectura de la red	37
Fig. 3.1 Pantalla general de Navegación	38
Fig. 3.2 Construcción de la Red	38
Fig. 3.3 Adición de los dispositivos a la red	39
Fig. 3.4 Tipo de visualización de los dispositivos	39
Fig. 3.5 Configuración para Dispositivos basados en archivos EDS	40
Fig. 3.6 Configuración de las propiedades de los dispositivos	40
Fig. 3.7 Guardando la configuración	41
Fig. 3.8 Configuración del Scanner (1769-SDN)	41
Fig. 3.9 Agregar y/o Quitar dispositivos de la red	42
Fig. 3.10 Mapeo de los dispositivos	42
Fig. 3.11 Uso de la recuperación de configuración y “ADR”	43
Fig. 3.12 Configuración Auto-recuperación	43
Fig. 3.13 Características de mantenimiento y Diagnóstico	44
Fig. 3.14 Ejemplo de una red DeviceNet con controlador MicroLogix 1500	44
Fig. 3.15 Como iniciar un proyecto en RSLogix 500	45
Fig. 3.16 Pantalla de configuración de entradas y salidas	45
Fig. 3.17 Leer configuración de entradas y salidas	46
Fig. 3.18 Entradas y Salidas instaladas	46
Fig. 3.19 Cambio de la cantidad de datos	47

Fig. 3.20 Lectura de la red	48
Fig. 3.21 En este módulo 1769-SDN solo son compatibles las formas de comunicación Polled y COS	49
Fig. 3.22 Mapeo de entradas y salidas del scanner nodo 0	49
Fig. 3.23 Mapeo del modo esclavo del scanner nodo 0	50
Fig. 3.24 Configuración módulos de I/O	51
Capítulo 4 Introducción a LabVIEW	51
Fig. 4.1 Panel Frontal y Diagrama de bloques	52
Fig. 4.2 Instrumento virtual en LabVIEW.	53
Fig. 4.3 Panel frontal de simple aplicación de preguntas	54
Fig. 4.4 Fragmento de Math Quiz Diagrama de bloques Mostrando Procesamiento de Datos Lógica.	55
Fig. 4.5 Plataforma LabVIEW	56
Fig. 4.6 Panel Frontal.	57
Fig. 4.7 Diagramas de bloques.	58
Fig. 4.8 Panel del conector	58
Fig. 4.9 Tecnología OLE	60
Fig. 4.10 OPC-integración de sistemas	61
Fig. 4.11 Especificaciones más importantes de la tecnología OPC	62
Capítulo 5 LabSocket	62
Fig. 5.1 LabSocket-Basic-Extender una Meta VI a uno o varios navegadores.	63
Fig. 5.2 Panorámica LabSocket-Basic, sistema Nube Cliente	64
Fig. 5.3 VI's deben ser incluida en el diagrama de bloques Target VI para iniciar y terminar LabSocket Support VI's.	65

Fig. 5.4 LabSocket-MultiClient - Extender instancias únicas de un Meta VI	
Separar Navegadores.	67
Fig. 5.5 WebSockets compatibles con Navegadores.	67
Fig. 5.6 Selecciona LabSocket Start.vi en la paleta de funciones	69
Fig. 5.7 Simple-Demo.vi	71
Fig. 5.8 LabSocket Ventana de Estado (Página de Registro)	72
Fig. 5.9 Panel Frontal Target VI “SimpleDemo.vi”	72
Fig. 5.10 Representación del Navegador de Target VI “SimpleDemo.vi”	73
Fig. 5.11 Ventana de Estado de LabSocket - Página para Clientes	73
Fig. 5.12 Ventana de Estado de LabSocket – Página de Diagnósticos	74
Fig. 5.13 Usando LabVIEW para supervisar las instancias Target VI en Servidor VI	75
Fig. 5.14 Sistema LabSocket general	76
Fig. 5.15 Sistema LabSocket-MultiClient general	77
Fig. 5.16 Tres Navegadores conectados simultáneamente a instancias únicas de concurso de aplicaciones	78

Resumen

La presente tesina proporciona conceptos fundamentales en redes de comunicación, protocolos industriales, configuración de interfaces, y herramientas Software.

Con todo integrado en una misma aplicación, se logra la implementación de sistemas de monitoreo y control las cuales pretenden como objetivo principal, la optimización de procesos concernientes a los campos de la educación e ingeniería y obtener un amplio panorama y conocimiento en la configuración de redes industriales.

Una vez analizados los protocolos es posible trabajar con labVIEW para crear nuestra configuración. En el capítulo I se estudiarán tareas de supervisión y control las cuales generalmente están más relacionadas con el software SCADA, en donde el operador puede visualizar en la pantalla del computador de cada una de las estaciones remotas que conforman un sistema, los estados de esta, las situaciones de alarma y toma de acciones físicas sobre algún equipo lejano.

El capítulo II proporciona las herramientas y los conocimientos necesarios para implementar una red industrial.

Se analizan las especificaciones en base a las normas de referencia tales como el modelo OSI las cuales definen las capas físicas, de enlace y de aplicación misma que se ocupan del modo en el que se transmiten, reciben y procesan los datos, la información y los mensajes así como los medios físicos que se utilizan en el armado de la red DeviceNet.

En el capítulo III se presentan las instrucciones para comunicar vía maestro-esclavo, una vez hecha la comunicación se analizan acciones de mapeo y configuración de los módulos de entrada/salida, mismas que serán necesarias en estudios posteriores para la configuración de interfaces Humanas hechas en programas como labVIEW.

En el capítulo IV se presentan conceptos básicos para empezar a utilizar LabVIEW y crear interfaces Hombre- Máquina a través de la configuración de protocolos de comunicación propios del software utilizado para poder hacer posible la interacción de aplicaciones dentro de una red DeviceNet para el caso del presente estudio.

Para poder llevar a cabo la interacción de dispositivos con DeviceNet se requiere de software específico cuya arquitectura muestra la configuración de instrucciones en programas como RsNetworx, y RSLogix de Allen Bradley.

Justificación

La irrupción de los microprocesadores en la industria ha posibilitado su integración a redes de comunicación con importantes ventajas, entre las cuales figuran:

- Mayor precisión derivada de la integración de tecnología digital en las mediciones.
- Mayor y mejor disponibilidad de información de los dispositivos de campo.
- Diagnóstico remoto de componentes
- DeviceNet al ser un estándar abierto proporciona a los usuarios finales multitud de soluciones técnicas.
- Menor tiempo de paro gracias a sus sistemas de diagnóstico y la posibilidad de aplicar mantenimiento predictivo.
- Reducción de cableado en las señales de Entrada/Salida.
- Posibilidad de pasar a otros sistemas de bus (gateways o pasarelas)
- Posibilidad de eliminar o añadir nodos en marcha sin afectar al resto de elementos y sin necesidad de herramientas de programación (plug and play).
- Programación y configuración durante el funcionamiento.
- DeviceNet ofrece la opción de seguridad intrínseca, permitiendo así su instalación en zonas con riesgo de explosión

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

Posibilidad de intercambio de información entre equipos que controlan fases sucesivas de un mismo proceso.

Facilidad de comunicación hombre máquina.

Uso de una base de datos común.

Un sistema DeviceNet ofrece un único punto de conexión para la configuración y el control, soportando tanto Entradas/Salidas, como mensajería. También ofrece la posibilidad de alimentar directamente del cable de red los nodos con bajos consumos, simplificando el cableado.

Objetivo

Muchas veces escuchamos en la industria la palabra protocolos de comunicación sin tener claro de que estamos hablando. El objetivo es familiarizarnos con este término, en este caso con el protocolo DeviceNet, expondré las principales características y fundamentos de uno de los más utilizados.

El principio de un protocolo de comunicación es un conjunto de reglas que permiten la transferencia e intercambio de datos entre los distintos que conforman una red.

Los buses de datos que permiten la integración de equipos para la medición y control de variables de proceso, reciben la denominación genérica de buses de campo.

El objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional lazo de corriente de 4-20 mA o 0 a 10 VDC, según corresponda.

Generalmente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLC's, transductores, actuadores, sensores y equipos de supervisión.

Antecedentes (SCADA)

Los primeros SCADA eran simplemente sistemas de telemetría, que proporcionaban reportes periódicos de las condiciones de campo vigilando las señales que representaban medidas y/o condiciones de estado en ubicaciones de campo remotas. Estos sistemas ofrecían capacidades muy simples de monitoreo y control, sin proveer funciones de aplicación alguna.

Un ejemplo de telemetría es el control de llenado de un depósito de agua desde una mina. Unos sensores se ocupan de vigilar el nivel de agua del depósito y el de la mina, informando a la bomba de cuándo ponerse en marcha para bombear agua al depósito, si hay nivel insuficiente en éste o cuando debe detenerse, si el depósito está lleno o el nivel de la mina no es suficiente.

Desde el centro de control podríamos ver la evolución del nivel de la estación y decidir las acciones pertinentes en caso de problemas (detener el bombeo manualmente, ver si hay elementos defectuosos, etc.).

Se puede decir que las primeras comunicaciones serias empezaron con el ferrocarril y el telégrafo. Para organizar el tráfico ferroviario se avisaban entre estaciones de las salidas y llegadas de los trenes, pues sólo se tendía una línea.

La clasificación de los diferentes sistemas de intercambio de información, telemetría y monitorización podría hacerse basándose en el sistema de transmisión:

Sistema de marcación automática

Los RTU (Remote Terminal Unit) capaces de comunicarse mediante caracteres ASCII (por ejemplo, mediante el protocolo ModBus), puede conectarse en cualquier combinación ordenador-módem.

Los primeros sistemas automatizados SCADA fueron altamente modificados con programas de aplicación específicos para atender a requisitos de algún proyecto particular.

Las redes de comunicaciones industriales deben su origen a la fundación FieldBus (Redes de Campo). La fundación FieldBus desarrolló un nuevo protocolo de comunicación para la medición y el control de procesos donde todos los instrumentos puedan comunicarse en una misma plataforma.

Un SCADA es un sistema basado en computadores que permite supervisar y controlar a distancia una instalación de cualquier tipo. A diferencia de los sistemas de control distribuido, el lazo de control es generalmente cerrado por el operador. Los sistemas de control distribuido se caracterizan por realizar las acciones de control en forma automática. Hoy en día es fácil hallar un sistema SCADA realizando labores de control automático en cualquiera de sus niveles, aunque su labor principal sea de supervisión y control por parte del operador.

Introducción

Los sistemas SCADA originalmente se diseñaron para cubrir las necesidades de un sistema de control centralizado, sobre procesos o complejos industriales distribuidos sobre áreas geográficas muy extensas. Tal es así que en definición clásica de un sistema SCADA se hace referencia a esta característica. Hoy en día, con el desarrollo de las redes digitales, la definición se tiene que modificar para incluir esta nueva forma de conectividad.

Las comunicaciones entre los instrumentos de proceso y el sistema de control se basan principalmente en señales analógicas (neumáticas de 3 a 15 psi en las válvulas de control y electrónicas de 4 a 20mA de corriente continua). Pero ya existen instrumentos digitales capaces de manejar gran cantidad de datos y guardarlos históricamente; su precisión es diez veces mayor que la de la señal típica de 4-20mA de corriente continua. En vez de transmitir cada variable por un par de hilos, transmiten secuencialmente las variables por medio de un cable de comunicaciones llamado bus.

DeviceNet se basa en CAN. Desarrollado por Allen Bradley en los años 90, e integrado posteriormente en ODVA (Open Device Vendor Association). Más de 700 vendedores o fabricantes han sido certificados por ODVA. Este hecho permite a los usuarios la implementación de productos de DeviceNet con la seguridad de disponer de productos para sus aplicaciones en cualquier lugar.

DeviceNet es una red digital de tipo abierto, muy flexible en su implementación, y de bajo coste, que sirve de nexo de unión entre reguladores industriales y dispositivos de Entrada/Salida (sensores fotoeléctricos, sensores magnéticos, pulsadores, etc.)

Un sistema DeviceNet ofrece un único punto de conexión para la configuración y el control, soportando tanto Entradas/Salidas, como mensajería. También ofrece la posibilidad de alimentar directamente del cable de red a los nodos con bajos consumos, simplificando el cableado.

Capítulo 1 Sistemas SCADA

1.1 Introducción a los sistemas de supervisión

Los sistemas SCADA contemporáneos exhiben predominantemente características de control a lazo abierto y utilizan comunicaciones generalmente interurbanas.

Sistemas similares a un sistema SCADA son vistos rutinariamente en fábricas, plantas de tratamiento, estos son llamados a menudo como Sistemas de Control Distribuido (DCS – “Distributed Control Systems”). Tienen funciones similares a los sistemas SCADA, pero las unidades de colección o de control de datos de campo se establecen generalmente dentro de un área confinada. Las comunicaciones pueden ser vía una red LAN, y serían normalmente confiables y de alta velocidad. Un sistema DCS emplea generalmente cantidades significativas de control a lazo cerrado.

Un sistema SCADA por otra parte, generalmente cubre áreas geográficas más grandes, y normalmente depende de una variedad de sistemas de comunicación menos confiables que una LAN.

La adquisición de datos es lograda en primer lugar por las RTU.

1.2 SCADA

SCADA viene de las siglas: “**Supervisory Control And Data Acquisition**”, es decir: hacer referencia a un sistema de adquisición de datos y control supervisor.

Tradicionalmente se define a un SCADA como un sistema que permite supervisar una planta o proceso por medio de una estación central que hace de Master (llamada también estación maestra o unidad terminal maestra, MTU) y una o varias unidades remotas (generalmente RTUs) por medio de las cuales se hace el control / adquisición de datos hacia / desde el campo.

Si bien las topologías que sobre las que se sustentan los sistemas SCADA se han adecuando a los servicios de los sistemas operativos y protocolos actuales, las funciones de adquisición de datos y supervisión no han variado mucho respecto a las que proponían en sus inicios.

Esquemáticamente, un sistema SCADA conectado a un proceso automatizado consta de las siguientes partes:

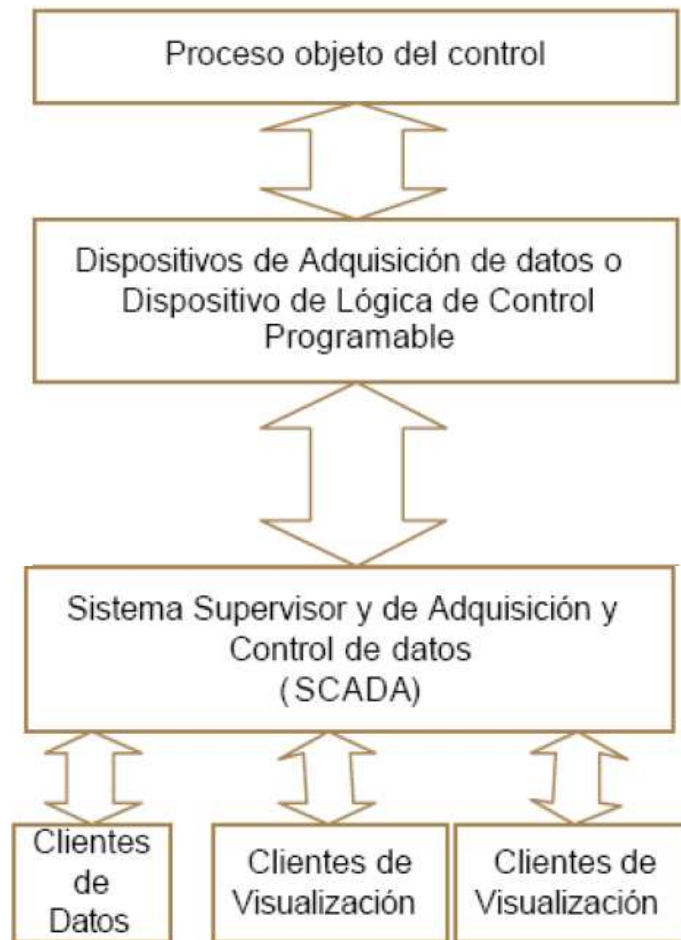


Fig.1.1 Partes de un sistema SCADA conectado a un proceso automatizado.

Proceso Objeto del Control: Es el proceso que se desea supervisar. En consecuencia, es el origen de los datos que se requiere coleccionar y distribuir.

Adquisición de Datos: Son un conjunto de instrumentos de medición dotados de alguna interfase de comunicación que permita su interconexión.

SCADA: combinación de hardware y software que permita la colección y visualización de los datos proporcionados por los instrumentos.

Cientes: Conjunto de aplicaciones que utilizan los datos obtenidos por el sistema SCADA.

Un término clave en la definición, al que muchas veces no se le da adecuada atención, es el de supervisión, que significa que un operador humano es el que al final tiene la última decisión sobre operaciones, generalmente críticas, de una planta industrial. La importancia de esta definición está en que se contrapone a la idea generalizada, que a veces si se hace, de que en la unidad master se hace control automático del proceso supervisado.

Es cierto que puede hacerse control automático, pero debe evaluarse suficientemente su implementación, tomando sobre todo en consideración la confiabilidad de los enlaces (en particular si son de larga distancia) que transportan los datos y comandos desde y hacia el campo. Una falla de comunicación, significaría dejar fuera de control el proceso. Esto explica por qué ahora la industria favorece a los sistemas de control distribuido.

1.3 Elementos de un sistema SCADA

- **Transductores:** reciben una señal de entrada función de una o más cantidades físicas y la convierten modificada o no a una señal de salida. Son transductores, un relé, un elemento primario, un transmisor, un convertidor PP/I (presión de proceso intensidad), PP/P (Presión de proceso a señal neumática), etc.
- **Sistema de comunicaciones:** Es la infraestructura tanto física como lógica encargada de llevar la información desde la fuente de generación hasta el procesador, incluye aspectos como: Medio físico (cables coaxial, fibra óptica, cobre, etc.), protocolo de comunicación (RS232, modbus, etc.) y topología de campo.
- **Unidad remota (Remote Terminal Unit: RTU):** Conjunto de elementos dedicados a labores de control y/o supervisión de un sistema, alejados del centro de control e interactuados con éste mediante algún canal de comunicación. Dentro de esta clasificación se puede mencionar. PLC (Programmable Logic Controller), tareas generales de control, IED (Intelligent Electronic Device) y tareas específicas de control.
- **Unidad central (Master TerminalUnit: MTU):** Centraliza el mando del sistema. Debido a la arquitectura abierta, es posible intercambiar información en tiempo real. En la MTU se

realiza, principalmente, la tarea de recopilación y archivado de datos. Toda esta información que se genera en el proceso se pone a disposición de los diversos usuarios que puedan requerirla. Se encarga de: Gestionar las comunicaciones, recopilar datos de todas las estaciones remotas, envío de información, comunicación con los operadores, análisis, impresión, visualización de datos y seguridad.

- **Interface Humano-Máquina (Human-Machine Interface, HMI):** Es un enlace entre el operador y el proceso, representa de forma gráfica ya sea una parte del proceso, o todo el proceso en sí, incluye aspectos como monitoreo, alarmas, registros de datos, históricos, seguridad, configuración de parámetros, etc.

Capítulo 2 Introducción a DeviceNet

2.1 Conceptos básicos

DeviceNet es una red digital, multipunto para conexión entre sensores, actuadores y sistemas de automatización industrial en general. Esta tecnología fue desarrollada para tener máxima flexibilidad entre los equipos de campo e interoperabilidad entre diferentes fabricantes.

Introducido originalmente en 1994 por Allen-Bradley, DeviceNet transfirió su tecnología ODVA en 1995. La ODVA (Open DeviceNet Vendor Association) es una organización sin fines de lucro compuesta por cientos de empresas alrededor del mundo que mantiene, difunde y promueve la tecnología DeviceNet y otras redes basadas en el protocolo CIP (Common Industrial Protocol).

La tecnología DeviceNet es un estándar abierto de automatización con el objetivo de transportar 2 tipos principales de información:

- Datos cíclicos de sensores y actuadores, directamente relacionados al control y,
- Datos no cíclicos indirectamente relacionados al control, como configuración y diagnóstico.

Los datos cíclicos representan la información intercambiada periódicamente entre el equipo de campo y el controlador. Por otro lado, los no cíclicos son informaciones intercambiadas eventualmente durante la configuración o diagnóstico del equipo de campo.

La capa física y de acceso a la red DeviceNet está basada en la tecnología CAN (Controller Area Network) y las capas superiores en el protocolo CIP, que define una arquitectura basada en objetos y conexiones entre ellos.

El CAN fue originalmente desarrollado por la BOSCH para el mercado de automóviles Europeos para sustituir el cableado costoso por un cable en red de bajo costo en automóviles. Como resultado, el CAN tiene respuesta rápida y confiabilidad alta para aplicaciones principalmente en las áreas automovilística.

Un ejemplo de red DeviceNet se muestra en la figura siguiente.

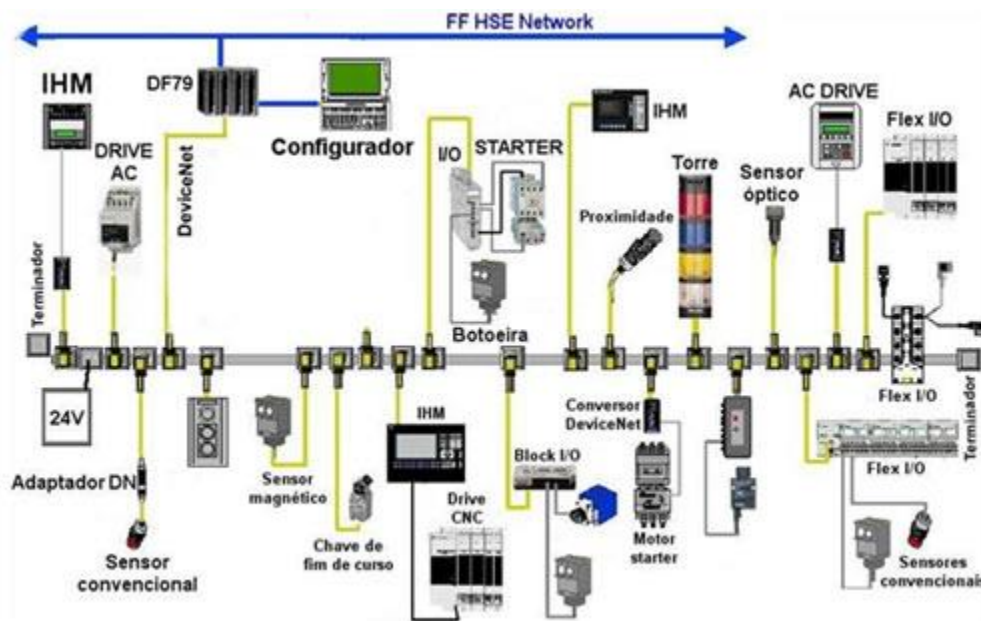


Fig. 2.1 Ejemplo de topología de la red DeviceNet

2.1.1 Características de la red

- Topología basada en bus principal con ramificaciones. El bus principal debe ser hecho con el cable DeviceNet grueso, y las ramificaciones con el cable DeviceNet delgado o plano. Cables similares podrán usarse siempre y cuando sus características eléctricas y mecánicas sean compatibles con las especificaciones de los cables estándar DeviceNet.
- Permite o uso de repetidores, bridges, ruteadores y gateways.
- Soporta hasta 64 nodos, incluyendo el maestro, direccionados de 0 a 63 (MAC ID).
- Cable de dos pares; uno para alimentación de 24V y otro para comunicación.

- Capacidad de insertar y cambiar en caliente, sin interrumpir a la red.
- Protección contra conexión inversa y corto-circuito.
- Alta capacidad de corriente en la red (hasta 16A).
- Varias fuentes pueden ser usadas en la misma red para satisfacer las necesidades de la aplicación en términos de la carga y la longitud de los cables.
- Velocidad de comunicación seleccionable: de 125,250 y 500Kbps.
- Comunicación basada en conexiones de E/S y modelos de pregunta y respuesta.
- Diagnóstico de cada equipo y de la red.
- Transporte eficiente de datos de control discretos y analógicos.
- Detección de direccionamiento duplicado en la red.

2.2 Buses de Campo

Son sistemas que facilitan la comunicación entre dispositivos de diferentes fabricantes sin necesidad de adaptaciones adicionales. En pocas palabras, los usuarios podrá, desarrollar productos basados en estos buses de campo a un costo razonable y sin mucho esfuerzo. Existe una completa disponibilidad de herramientas y componentes hardware y software.

Un bus de campo debe permitir por lo menos tres cosas:

- Interconectividad: los equipos de diversos fabricantes pueden ser conectados en forma segura al bus.
- Interoperabilidad: es la habilidad para la conexión de diversos elementos de diversos fabricantes,
- Intercambio: los equipos de un fabricante pueden ser reemplazados con funcionalidad equivalente por equipos de otros fabricantes.

La interconectividad es el común denominador. Si la interoperabilidad no puede ser lograda, la misma operación del bus de campo es limitada y tal bus de campo se convierte en una opción poco útil. La última meta es la capacidad de intercambio. Esto solo puedes ser posible si las especificaciones son completas y se cuenta con un apropiado sistema de pruebas y validación de los equipos.

2.2.1 Ventajas de los buses de campo

La principal ventaja que ofrecen los buses de campo, y la que los hace más atractivos a los usuarios finales, es la reducción de costes. El ahorro proviene fundamentalmente de tres fuentes: ahorro en coste de instalación, ahorro en el coste de mantenimiento y ahorros derivados de la mejora del funcionamiento del sistema.

Una de las principales características de los buses de campo es una significativa reducción en el cableado necesario para el control de una instalación. Cada célula de proceso sólo requiere un cable para la conexión de los diversos nodos. Se estima que puede ofrecer una reducción de 5 a 1 en los costes de cableado. En comparación con otros tipos de redes, dispone de herramientas de administración del bus que permiten la reducción del número de horas necesarias para la instalación y puesta en marcha.

El hecho de que los buses de campo sean más sencillos que otras redes de uso industrial como por ejemplo MAP, hace que las necesidades de **mantenimiento de la red** sean menores, de modo que la fiabilidad del sistema a largo plazo aumenta. Además, los buses de campo permiten a los operadores monitorizar todos los dispositivos que integran el sistema e interpretar fácilmente las interacciones entre ellos. De esta forma, la detección de las fuentes de problemas en la planta y su corrección resulta mucho más sencilla, reduciendo los costes de mantenimiento y el tiempo de parada de la planta.

Los buses de campo ofrecen mayor **flexibilidad** al usuario en el diseño del sistema. Algunos algoritmos y procedimientos de control que con sistemas de comunicación tradicionales decían incluirse en los propios algoritmos de control, radican ahora en los propios dispositivos de campo, simplificando el sistema de control y sus posibles ampliaciones.

También hay que tener en cuenta que las prestaciones del sistema mejoran con el uso de la tecnología de los buses de campo debido a la **simplificación** en la forma de obtener información de la planta desde los distintos sensores. Las mediciones de los distintos elementos de la red están disponibles para todos los demás dispositivos. La simplificación en la obtención de datos permitirá el diseño de sistemas de control más eficientes.

Con la tecnología de los buses, se permite la **comunicación bidireccional** entre los dispositivos de campo y los sistemas de control, pero también entre los propios dispositivos de campo.

2.3 Protocolo DeviceNet

DeviceNet se basa en CAN. Desarrollado por Allen Bradley en los años 90, e integrado posteriormente en ODVA (Open Device Vendor Association). Más de 700 vendedores o fabricantes han sido certificados por ODVA. Este hecho permite a los usuarios la implementación de productos de DeviceNet con la seguridad de disponer de productos para sus aplicaciones en cualquier lugar.

DeviceNet es una red digital de tipo abierto, muy flexible en su implementación, y de bajo coste, que sirve de nexo de unión entre reguladores industriales y dispositivos de Entrada/Salida (sensores fotoeléctricos, sensores magnéticos, pulsadores, etc.)

Cada dispositivo es un nodo de red.

Es una tecnología diseñada para satisfacer las exigencias de fiabilidad requeridas por los ambientes industriales. Hace servir CAN y CIP para las capas superiores de red.

DeviceNet también permite que un fabricante añada facultades exclusivas de sus productos además de las mínimas requeridas (evidentemente, los “extras” deben seguir las especificaciones de DeviceNet).

Se basa en el diálogo Productor/Consumidor, y soporta múltiples jerarquías de comunicación y priorización de mensajes. Los sistemas basados en DeviceNet pueden configurarse para funcionar como Maestro/Esclavo, o con enlaces punto a punto (peer to peer)

Un sistema DeviceNet ofrece un único punto de conexión para la configuración y el control, soportando tanto Entradas/Salidas, como mensajería. También ofrece la posibilidad de alimentar directamente del cable de red los nodos con bajos consumos, simplificando el cableado.

DeviceNet es una de las tres tecnologías de redes abiertas y estandarizadas, cuya capa de aplicación usa el CIP (Common Application Layer – Capa de Aplicación Común). Al lado de ControlNet y EtherNet/IP, posee una estructura común de objetos, Es decir, es independiente del medio físico y de la capa de enlace de datos.

Esta capa es una capa de aplicación estándar, integrada a interfaces de hardware y software abiertas, constituye una plataforma de conexión universal entre componentes en un sistema de automatización, desde la fábrica hasta el nivel de internet. La siguiente Figura muestra una arquitectura del CIP dentro del modelo OSI.

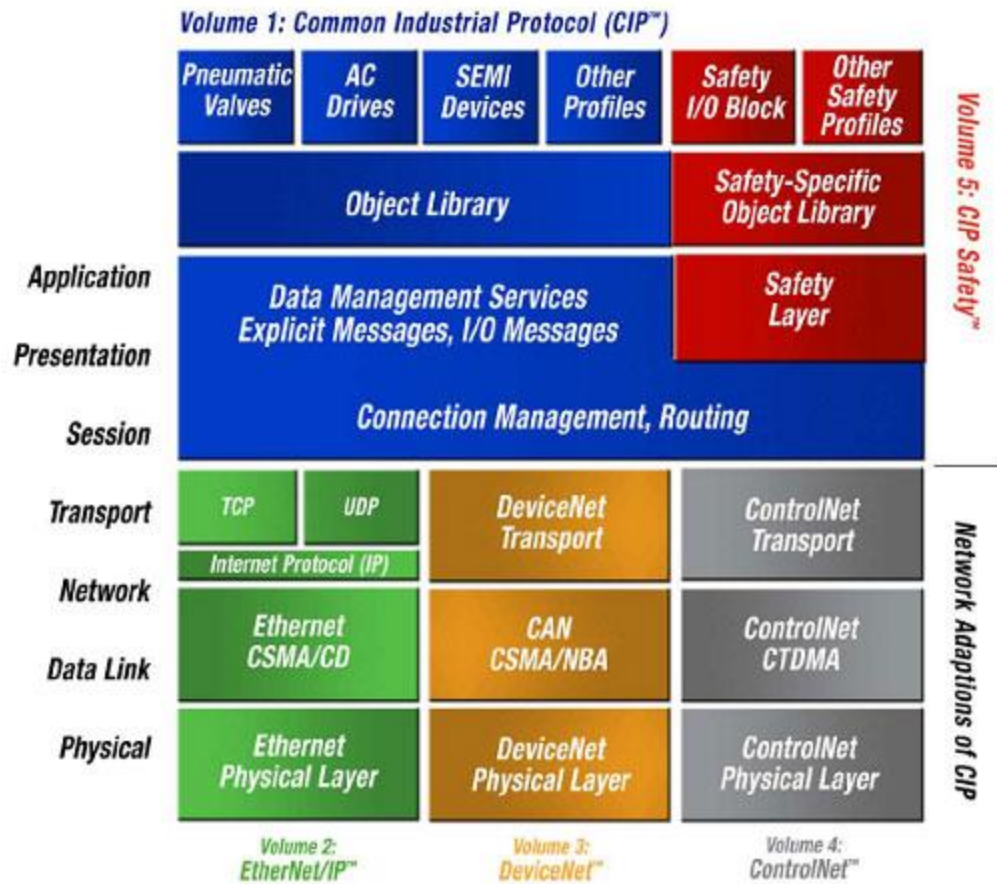


Fig. 2.2 El modelo OSI de los objetos del CIP

CIP tiene dos objetivos principales:

- Transporte de datos de control de los dispositivos de I/O.
- Transporte de informaciones de configuración y diagnóstico del sistema que se controla.

Un nodo de DeviceNet es entonces modelado por un conjunto de objetos CIP, los cuales encapsulan datos y servicios y determinan así mismo su comportamiento.

2.3.1 El modelo de Objeto

Un nodo DeviceNet es modelado como una colección de objetos. Un objeto proporciona una representación abstracta de un componente dentro de un producto. Un ejemplo de objetos y una clase de objeto tiene atributos (datos), que proporcionan los servicios (métodos o procedimientos), e implementan comportamientos. Atributos, ejemplos, clases y direccionamiento del nodo desde (0-63) son direccionados por número.

Existen objetos obligatorios (que tienen todos dispositivos) y objetos opcionales. Los objetos opcionales son aquellos que moldean el dispositivo de acuerdo con su categoría (llamado perfil) a la que pertenecen, tales como: AC/DC Drive, lector de código de barras o válvula neumática. Por ser diferentes, cada uno de ellos tendrá también un conjunto diferente de objetos.

2.3.2 La capa de Enlace de Datos (Data link layer)

DeviceNet utiliza el estándar CAN sobre la capa de enlace de datos. El gasto mínimo requerido por el protocolo CAN en la capa Enlace de Datos mejora el trabajo de DeviceNet cuando se trata de mensajes. El marco de datos DeviceNet utiliza solamente el tipo de estructura de datos del protocolo CAN (entre otros existentes en el protocolo CAN). El protocolo utiliza un mínimo de banda para la transmisión de los mensajes de CIP. El formato de la estructura de datos DeviceNet se muestra en la siguiente Figura.

1bit	11 bits	1 bit	6 bits	0-8 bytes	15 bits	1 bit	1 bit	1 bit	7 bits	3 bits
Início da Frame	Identificador	RTR bit	Campo de Controle	Campo de DADOS (Variável de 0 a 8 bytes)	Sequenciador CRC	Delimitador de CRC	Bit de Ack	Delimitador de Ack	Final da Frame	Espaço entre Frames
Campo de Arbitração										

Fig.2.3 Formato de la estructura de datos.

2.3.3 Modos de comunicación

El protocolo DeviceNet tiene dos tipos básicos de mensajes, mensaje cíclico I/O y explícito. Cada uno de ellos es adecuado a un determinado tipo de dato, conforme se describe abajo:

- **Cíclic I/O:** tipo de telegrama síncrono dedicado al procesamiento de datos prioritarios entre un productor y uno o más consumidores. Se dividen de acuerdo con el método de intercambio de datos. Los principales son:
 - **Polled:** método de comunicación donde el maestro envía un telegrama a casa uno de su lista de esclavos (scan list). Así mismo, en cuanto reciba la solicitud, el esclavo que, se direcciona y responde de acuerdo con lo programado.
 - **Bit-strobe:** método de comunicación donde el maestro envía un telegrama por la red con 8 bytes de datos. Cada bit de estos 8 bytes representa un esclavo que, se direcciona y responde de acuerdo con lo programado.
 - **Cambio de estado:** método de comunicación donde el intercambio de datos entre el maestro esclavo que ocurre cuando hubo cambios en los valores monitoreados/controlados, hasta un cierto límite de tiempo. Cuando este límite es alcanzado, la transmisión y recepción ocurren, incluso sin alteraciones. La configuración de esta variable de tiempo es hecha en el programa de configuración de la red.
 - **Cíclico:** otro método de comunicación muy semejante al anterior. La única diferencia está en la producción y consumo de mensajes. En este tipo, todo el intercambio de datos ocurre en intervalos regulares de tiempo, independiente de ser alterados o no. Este periodo también es ajustado en el software de configuración de la red.
- **Mensaje Explícito:** tipo de telegrama de uso general y no prioritario. Utilizado principalmente en tareas asíncronas tales como parametrización y configuración del equipo.

2.3.4 Normas Internacionales

Son normas que afectan a nivel mundial:

- ISO (International Standards Organization), que genera normas para todas las áreas y coordina las creadas por organizaciones regionales.
- IEC (International Electrotechnical Commission). Elabora normas para el área eléctrica.
- ITU o UIT (International Telecommunication Union). Es un organismo constituido por administraciones de más de 150 países, adopta normas que regulan el uso del espectro radioeléctrico en los ámbitos espacial y terrestre. Está estructurada en tres sectores, que son :
 - ITU-T para las telecomunicaciones.
 - ITU-R para la radiocomunicación.
 - ITU-D para el desarrollo de las telecomunicaciones.

2.3.5 El protocolo CIP (Common Industrial Protocol)

OSI representa una arquitectura genérica de red. Cualquier red de tipo abierto se ciñe, en mayor o menor medida, a este esquema, haciendo uso de las tecnologías más adecuadas a cada situación.

Cada medio físico de transmisión tiene sus propios requerimientos. Si, por ejemplo, un usuario necesita una red orientada a seguridad intrínseca, los objetivos principales del diseño serán del tipo de soporte. Por tanto, un fabricante utilizará una capa de aplicación que se adapte a las necesidades de las capas más bajas de su producto.

La consecuencia es que, debido a los múltiples entornos que aparecen en una planta de fabricación, se utilizarán las redes que mejor cubran las necesidades de cada entorno.

Esto arrastrará numerosos protocolos de aplicación trabajando en el mismo espacio, o lo que es lo mismo:

- ❖ Diferencias de configuración.
- ❖ Formación del personal en múltiples disciplinas.
- ❖ Costes de desarrollo para los fabricantes.

Las soluciones implementadas según las directrices del protocolo industrial común, CIP, permiten integrar la gestión de entradas y salidas, la posibilidad de configuración de dispositivos, y la recogida de datos de los elementos de la red, siendo posible que todo ello tenga lugar en varias redes de comunicación. Esto último permite reducir los costes de ingeniería, puesta en marcha y mantenimiento.

CIP es una aproximación basada en objetos, orientada a permitir el diseño de dispositivos de control que combina el método de direccionamiento de red y las reglas de intercambio de mensajes.

2.3.6 Protocolo de comunicación CAN

El enlace de comunicación de DeviceNet está basado en la transmisión orientada, del protocolo de comunicación CAN (Controlador de Red por Área).

El protocolo CAN fue originalmente desarrollado por BOSCH para industria automotriz Europa hecha para el reemplazo costoso de los arneses, con un cable de red de bajo costo en los automóviles. Como resultado, el protocolo CAN tiene una rápida respuesta y alta confiabilidad para aplicaciones que se requieren como el control anti-bloqueo de frenos y bolsas de aire. Los productos de DeviceNet usan los mismos integrados CAN que son usados en la industria automotriz y para otras aplicaciones de consumo comercial.

El protocolo de comunicaciones CAN proporciona los siguientes beneficios:

- Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o bus.
- El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas.
- Al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto, excepto en los enganches.

2.3.7 Características de CAN

CAN se basa en el modelo productor/consumidor, el cual es un concepto, o paradigma de comunicaciones de datos, que describe una relación entre un productor y uno o más consumidores. CAN es un protocolo orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los nodos deciden aceptar o no dicho mensaje. Dentro de sus principales características se encuentran:

- Prioridad de mensajes.
- Garantía de tiempos de latencia.
- Flexibilidad en la configuración.
- Recepción por multidifusión (multicast) con sincronización de tiempos.
- Sistema robusto en cuanto a consistencia de datos.
- Sistema multimaestro.
- Detección y señalización de errores.
- Retransmisión automática de tramas erróneas.
- Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

DeviceNet es una red de bajo nivel adecuada para conectar dispositivos simples como sensores fotoeléctricos, sensores magnéticos, pulsadores, etc. Y dispositivos de alto nivel (PLC, controladores, computadores, HMI, entre otros). Provee información adicional sobre el estado de la red, cuyos datos serán desplegados en la interfaz del usuario.

La conexión directa proporciona mejoras de comunicación entre los dispositivos, también como el nivel de diagnóstico de los dispositivos que en la mayoría de las instalaciones no es disponible o accesible a través del cableado de las interfaces de entradas y salidas.

DeviceNet es una sencilla red de trabajo que reduce el costo, el tiempo de cableado y la instalación de los dispositivos industriales de automatización, la cual proporciona la habilidad de intercambio de componentes de diferentes fabricantes de dispositivos industriales.

2.4 Modelo OSI

El modelo OSI, quiere decir Open System Interconnection o interconexión de Sistemas abiertos, fue definido por la OSI en el año 1983.

El modelo OSI está formado por siete capas o niveles.

Cada capa o nivel tiene unas funciones claramente definidas y que son las siguientes:

- Nivel 1. Física: Especifica cuál será el medio físico de transporte a utilizar. Señales eléctricas.
- Nivel 2. Enlace: Estructuración de los datos dentro de la trama y control de errores.
- Nivel 3. Red: Interviene en el caso en el que intervenga más de una red.
- Nivel 4. Transporte: División de los datos en paquetes de envío.
- Nivel 5. Sesión: Para el control del inicio y finalización de las conexiones.
- Nivel 6. Aplicación: Utilización de los datos.

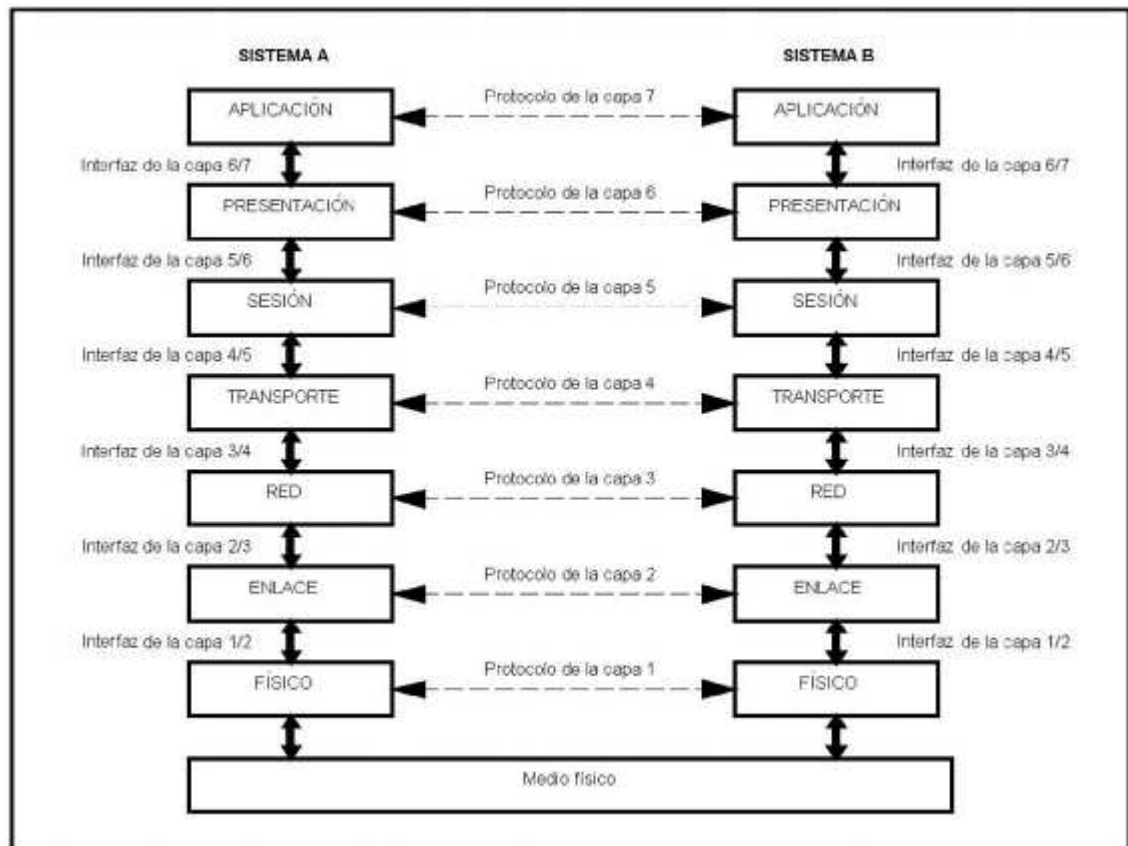


Fig.2.4 Sistema de control distribuido.

2.4.1 Modelo OSI para las comunicaciones Industriales

A nivel de las comunicaciones industriales las capas utilizadas son:

Física: Se encarga de la transmisión de los bits al canal de comunicación.

Define los niveles de la señal eléctrica con la que se trabajará.

Controla la velocidad de transmisión (duración de un bit).

Esta capa física contiene tres subniveles, que son los siguientes:

MEDIO: Canal de transmisión, si es el cable, FO, radio, etc.

MAU (Media Attachment Unit): Contiene la electrónica donde se generan o donde se reciben los niveles eléctricos.

PLS (Physical Logical Signal): Codificación en la emisión de la información binaria a señales eléctricas y decodificación en la recepción de la señal eléctrica a señal digital binaria.

Enlace: Se encarga de establecer una comunicación libre de errores entre dos equipos.

Forma la trama organizando la información binaria y la pasa a la capa física.

Esta capa física contiene dos subniveles, que son los siguientes:

MAC (Media Access Control): Control del canal de transmisión para que en el momento que esté libre, pueda enviar la información.

LLC (Logical Link Control): Controla y recupera los errores, también codifica la información (hexadecimal o ASCII) a enviar a formato binario o decodifica la información binaria recibida a hexadecimal o ASCII.

Aplicación: Es la capa más próxima al usuario y puede ofrecer servicios tales como correo electrónico, acceso a base de datos, transferencia de ficheros, videoconferencia.

2.5 Adaptación del Protocolo DeviceNet al modelo OSI

El protocolo DeviceNet incorpora en su estructura los protocolos CAN y CIP.

La capa de enlace de datos está definida íntegramente por las especificaciones del protocolo CAN para la capa de enlace de datos.

DeviceNet utiliza el protocolo CIP (Protocolo Industrial Común) para las capas superiores (Aplicación, Presentación y Sesión). Como el protocolo CIP define una capa de aplicación común, los datos de las redes basadas en este protocolo no variarán aunque cambie el tipo de red. Al definir los perfiles de los dispositivos, si éstos se ciñen al estándar, dos dispositivos que sigan el mismo perfil reaccionarán de manera similar.

Los mensajes que se generan en una red que utilice este protocolo, podrán pasar por varias redes sin necesidad de retocar el mensaje en la capa de aplicación, solo es necesario indicar la ruta que deberán seguir los datos.

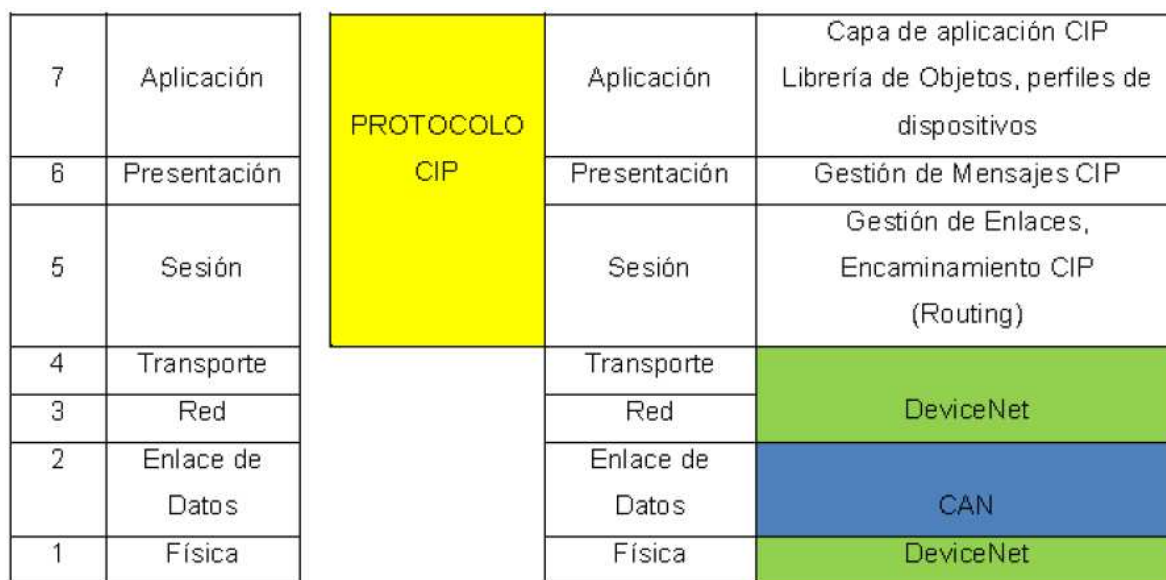


Fig.2.5 Modelo OSI y adaptación del protocolo DeviceNet al modelo OSI.

2.6 Capa física del protocolo DeviceNet

Esta capa especifica las características eléctricas, mecánicas, funcionales y de procedimiento que se requieren para mover los bits entre cada extremo del enlace de la comunicación.

Define:

- Las topologías aceptadas.
- El modo de emisión, o toma de la señal.
- El soporte de transmisión.
- Características físicas de los conectores.
- Características eléctricas de las señales.
- Características eléctricas del hardware.
- Implementación de las señales.
- Codificación.

DeviceNet Soporta hasta 64 Nodos (Gama de direcciones del nodo a partir de la 0-63), los nodos se pueden remover con el bus energizado y sin dividir el cable troncal, las velocidades de transmisión son de 125Kbps, 250Kbps, 500Kbps dependiendo del tipo de cable elegido, utiliza la topología de bus troncal, con caída de líneas, con par trenzado se tienen muy bajas pérdidas, y bajos retardos de comunicación.

Utiliza conectores abiertos y conectores sellados además, soporta tanto dispositivos Opto-Aislados y no Aislados, el bus de alimentación está separado está contenido en el mismo cable soportando tanto dispositivos con alimentación propia como aquellos sin alimentación, el consumo total de corriente por los dispositivos es de hasta 8 Amperes.

La función de "Power Tap" permite alimentar la red con varias Fuentes, dependiendo de los requerimientos.

2.6.1 Topología de red del Protocolo DeviceNet

La topología define la disposición de los diferentes equipos alrededor del medio de transmisión de datos.

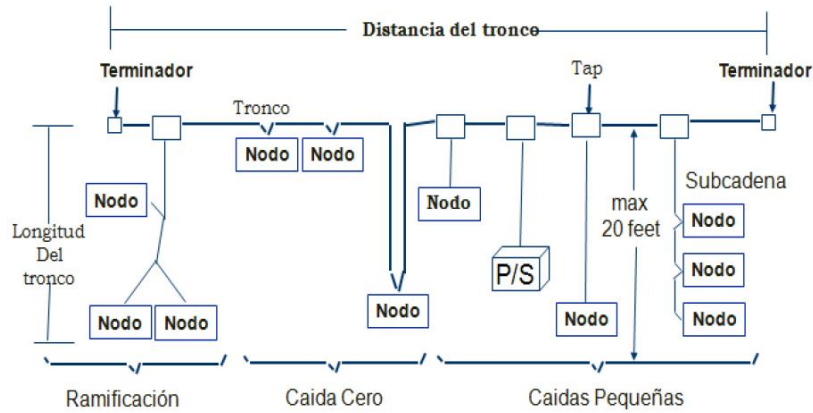


Fig. 2.6 Topología de DeviceNet

Los cables DeviceNet más usados tienen 5 conductores identificados y utilizados de acuerdo con la siguiente tabla:

Color del cable	Señal	Cable redondo	Cable chato
Bianco	CAN_H	Sinal DN	Sinal DN
Azul	CAN_L	Sinal DN	Sinal DN
Alambre desnudo	Dreno	Protección	No usado
Negro	V-	Alimentación	Alimentación
Rojo	V+	Alimentación	Alimentación

Fig. 2.7 Esquema de colores de los cables DeviceNet

2.6.2 Terminaciones de red

Los terminadores en la red DeviceNet ayudan a minimizar las reflexiones en la comunicación y son esenciales para el funcionamiento de la red. Los resistores de terminación (121Ω, 1%, 1/4W) deben ser colocados en los extremos del bus, entre los hilos CAN_H y CAN_L (blanco azul).

- No coloque el terminador dentro de un equipo o en el conector porque al ser movido también mueve el terminador causando una falla general en la red.
Deje los terminadores siempre independientes y aislé los extremos del bus, de preferencia dentro de las cajas protectoras p cajas de paso.
- Par verificar si los terminadores están presentes en la red, mida la resistencia entre los hilos CAN_H y CAN_L (blanco y azul) con la red desenergizada: la resistencia medida debe estar entre 50 y 60 Ohms.

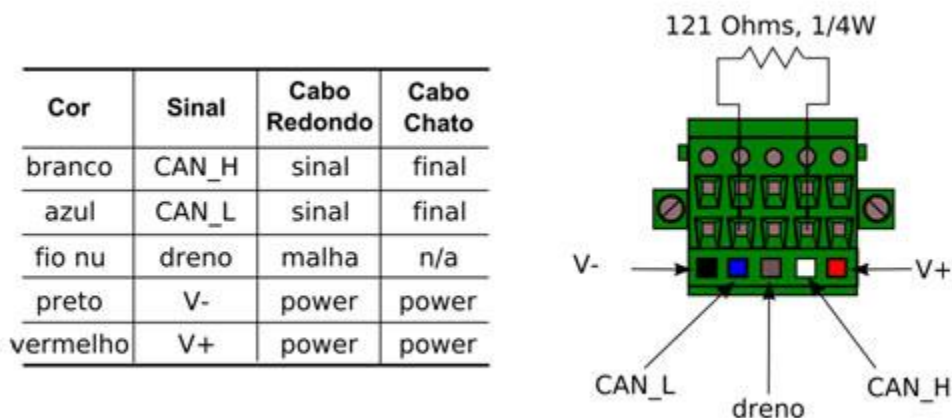


Fig. 2.8 Conexión de los resistores de terminación.

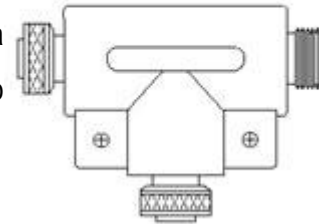
2.6.3 Conectores Derivadores para DeviceNet

Existen varios tipos de derivadores “TAPS” que pueden ser conectados en una red del tipo DeviceNet. Estos derivadores permiten combinar varios elementos de la red. Se clasifican como:

Derivación en T “T-Port TAP”

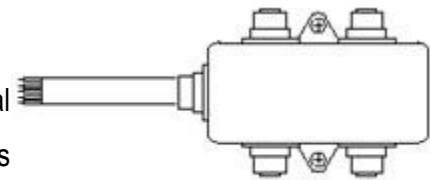
T-Port Tap

El derivador “T-Port” conecta un dispositivo simple o una línea de derivación “línea caída” a través de un conector tipo conexión rápida.



Derivación de dispositivo “Device-Port”

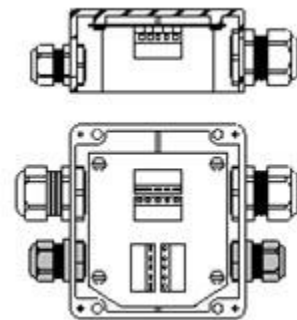
“DevicePort” son componentes sellados que conectan al “Bus de la línea” vía “línea caída” a través de conectores de desconexión rápida solamente dispositivos compatibles a la red DeviceNet.



DeviceBox Tap

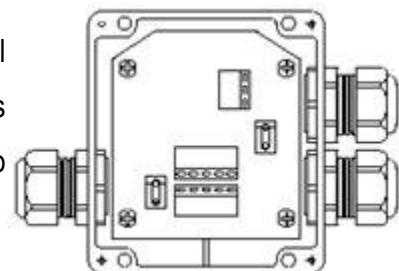
Derivación tipo box “DeviceBox”

“DeviceBox” son elementos pasivos que se conectan directamente a los dispositivos DeviceNet en “Trunk Line” a través de conexiones de terminales para hasta 8 nodos. Ellos poseen tapa removible sellada que permite montaje sobre la máquina o fábrica.



Derivación de Alimentación “PowerTap”

El “PowerTap” posee protección de sobre corriente para el cable tipo “thick” (grosso). Con protección de diodo y es posible utilizar varios “PowerTap” permitiendo así mismo el uso de varias fuentes de alimentación.



2.6.4 Suministro de alimentación

Una primera norma para la alimentación determina que el voltaje de la fuente en el arranque tiene que elevarse al 5% de su voltaje nominal en un tiempo de elevación menor a 250 ms.

Al mismo tiempo se deberá verificar lo siguiente:

- Que la fuente dispone de protección limitadora de corriente.
- Que todo segmento de la red dispone de protección por fusible.
- El correcto dimensionado de la fuente.

2.7 Capa de enlace del protocolo DeviceNet

Establece la forma de agrupar los datos en paquetes de longitud adecuada y añade los mecanismos necesarios para poder controlar la transmisión de información y poder detectar y corregir los errores que puedan aparecer, se basa en el protocolo de comunicación serial Controller Area Network (CAN) desarrollado por Rober Bosch.

2.7.1 CSMA/CD

El método para enviar y recibir la información evitando colisiones está regido por un algoritmo CSMA/CD (Carrier Sense Multiple Access / Collision Detection), Además de contar con identificadores de prioridad.

En el método de acceso CSMA/CD, los dispositivos de red que tienen datos para transmitir funcionan en el modo “escuchar antes de transmitir”. Esto significa que cuando un nodo desea enviar datos, primero debe determinar si los medios de red están ocupados o no.

2.7.2 Comunicación Broadcast

La comunicación Broadcast, es cuando todos los dispositivos en la red escuchan la información que se está transmitiendo, y es tarea de cada dispositivo decidir si la información es relevante para él.

Remote Transmission Requests, son preguntas hechas por un nodo a uno o varios nodos de la red y el nodo o aquellos nodos que tengan la respuesta contestan la información solicitada.

Cuando un nodo transmite, todos los demás nodos escuchan y solo comienza a transmitir cualquier otro nodo cuando el bus está libre.

2.7.3 Trama de datos

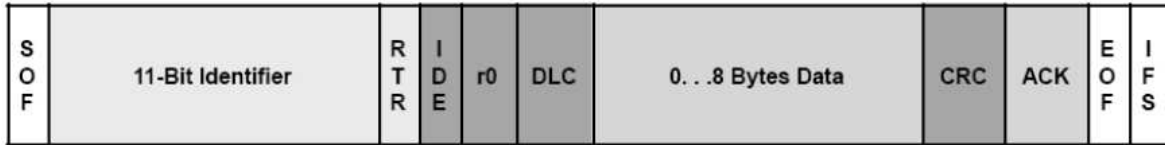


Fig. 2.9 Formato de la trama de datos

- **SOF:** Bit de inicio de trama sirve para sincronizar el bus.
- **11-Bits identifier:** Identificador del Mensaje que está transmitiendo.
- **RTR (Remote Transmission Request):** Arbitrea Prioridad baja en tramas de respuesta.
- **IDE:** Identifica la versión de CAN Estándar o Extendida: Versión Extendida posee un identificador de 29 bit, separados 11bits + 18 bits que siguen al bit IDE. DeviceNet usa la versión estándar.
- **r0:** Reservado para posible uso futuro.
- **DLC:** 4 Bits que indican el número de Bytes de datos que se están transmitiendo Min = 0; Max=8.
- **0...8 Bytes Data:** Datos transmitidos.
- **CRC:** Chequeo de Errores con el polinomio: $x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+1$
- **ACK 2 Bit:** Reconocimiento de campo, el nodo que corresponde a la trama indica mediante estos bits que la información es correcta.
- **EOF 7 Bits:** Fin de trama.
- **IFS:** Intervalo de espera para dar acceso a otro nodo.

2.7.4 Transmisión de Mensajes

La transmisión de mensajes puede ser inicializada por:

- **Encuesta:** Mensaje hecho por algún nodo (cliente) a algún específico (Servidor), quien proporciona sus datos.
- **Cambio de estado:** Es una transferencia que se envía cada que ocurre algún cambio de estado en los datos del dispositivo.
- **Mensaje Estroboscopio:** tiene una longitud de 64 Bits, uno para cada nodo, en el cual cada nodo responde con datos, (8 Bytes máximo).

- **Cíclico:** Se envía el mensaje cada determinado tiempo, Ejemplo: cada 10 ms.

2.8 Capa de aplicación del protocolo DeviceNet

Localiza las funciones de usuario y los servicios de comunicación. Presta servicios al usuario, que comprenden la interacción directa con los procesos de aplicación, manejando las transferencias de ficheros, base de datos, correo electrónico, etc. “En esta capa se establecen una serie de normas estándar para realizar los servicios de comunicación. En automatización se utiliza mayoritariamente el MMS (Manufacturing Message Specification), en el que incluyen los servicios y protocolos MAP (Manufacturing Automation Protocol).”

Capítulo 3 Arquitectura de la red

3.1 Configuración de la red y ajustes principales de RSNetWorx para DeviceNet

La configuración de la red no es más que establecer los parámetros de funcionamiento de todos los dispositivos que la conforman, para que estos tengan una respuesta apropiada en el momento que se ponga en marcha la red, y en su funcionamiento continuo o posterior. Para establecer estos parámetros se debe conocer cómo actuará la red para la aplicación.

Dependiendo de la funcionalidad de los dispositivos, se podrá realizar la configuración a través de software o sobre el mismo dispositivo. En la red del proyecto todos los dispositivos se los puede configurar mediante software; sin embargo, algunos presentaran ciertas características o parámetros que tendrán que ser establecidos manualmente (sobre el dispositivo).

RSNetWorx es el software que permite el manejo y al configuración de la red DeviceNet; permite el monitoreo y programación de los dispositivos o mantenimiento, si así se requiere. Además, verificar su instalación en el computador.

Si se dispone de una red flexible el computador que contiene el software de manejo, configuración y mantenimiento se conecta directamente a la red DeviceNet a través de la interfaz 1747-SDN o indirectamente a la red DeviceNet a través del PLC por el puerto RS-232.

Para todas estas tareas el software a utilizar es:RSlogix 500; RSLinx Classic Professional; el RSNetWork para DeviceNet; y LabVIEW.

3.2 Armado de la Red

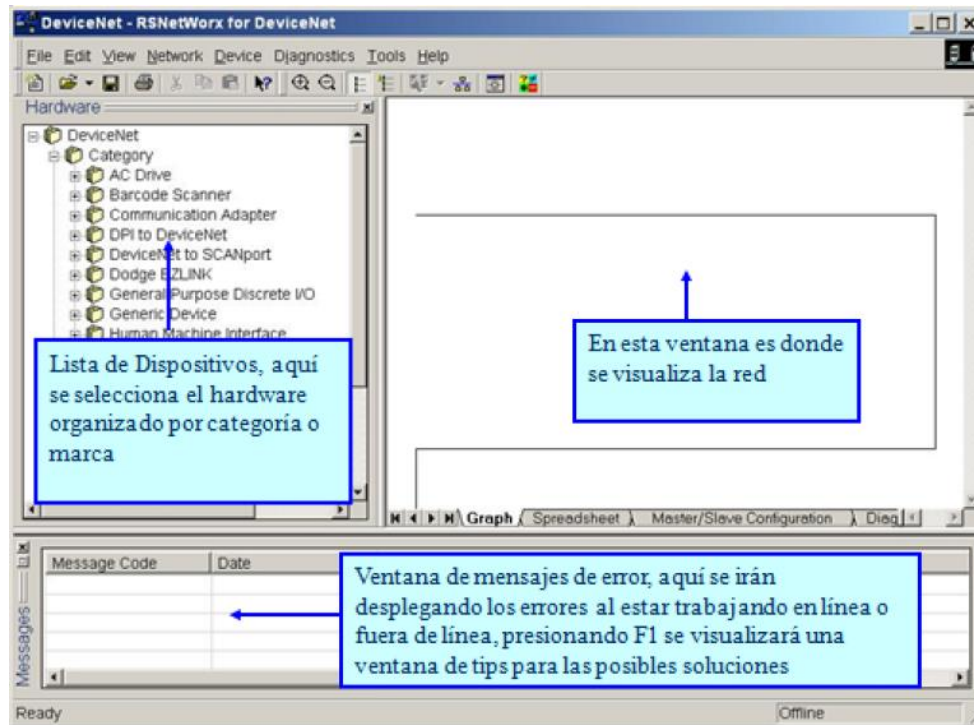


Fig. 3.1 Pantalla general de Navegación

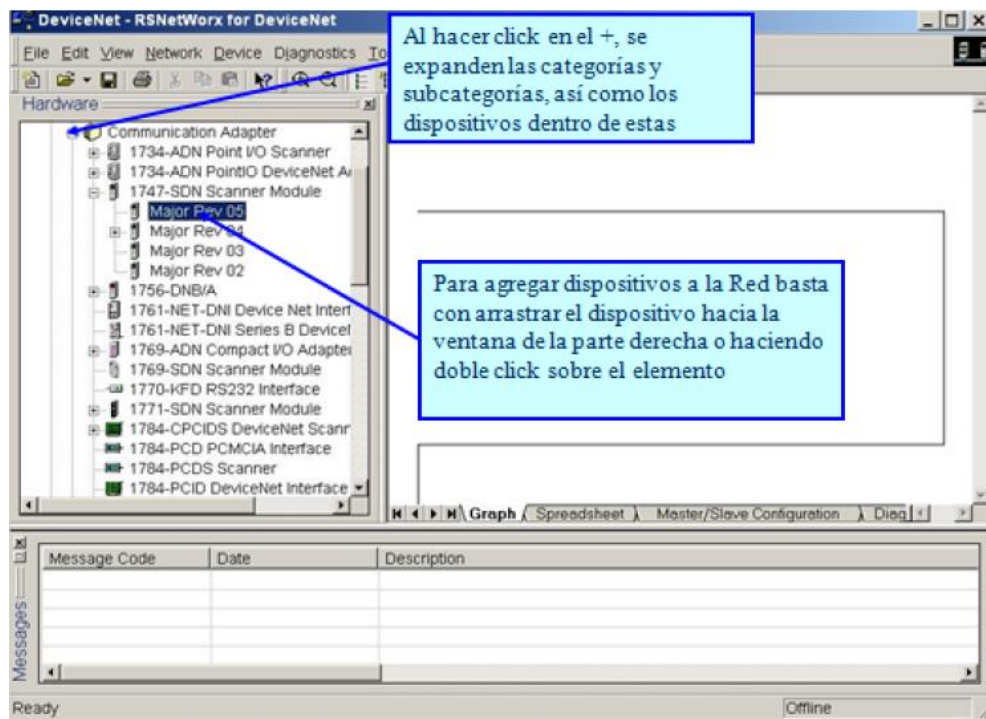


Fig. 3.2 Construcción de la red.

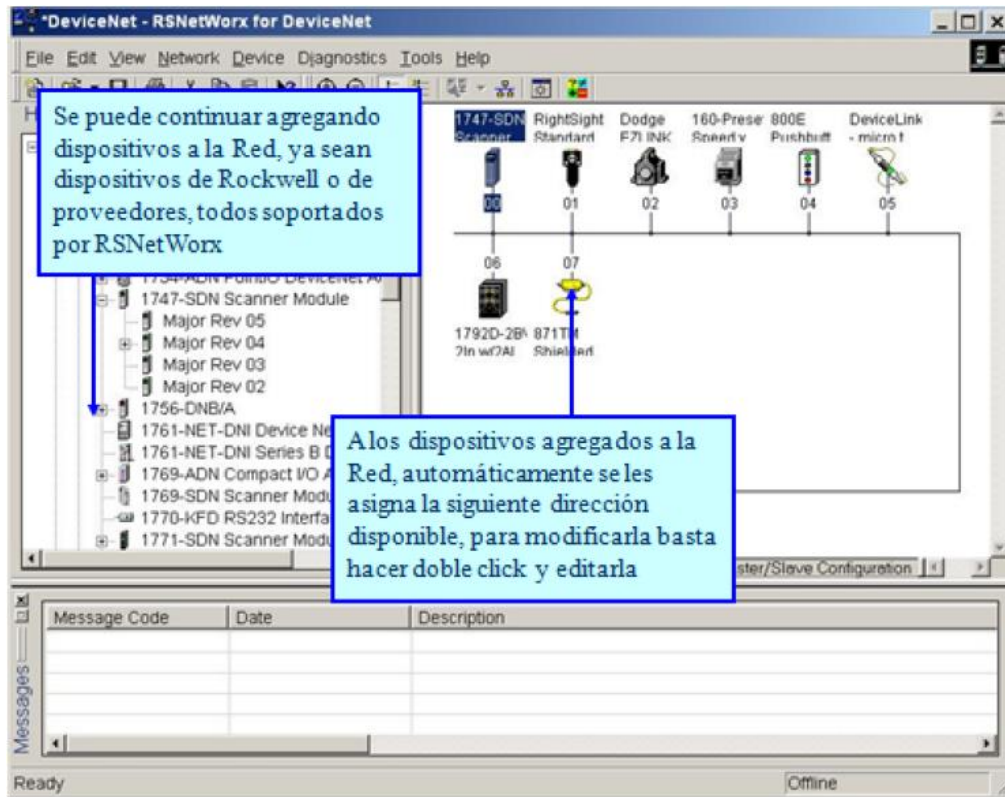


Fig. 3.3 Adición de los dispositivos a la Red.

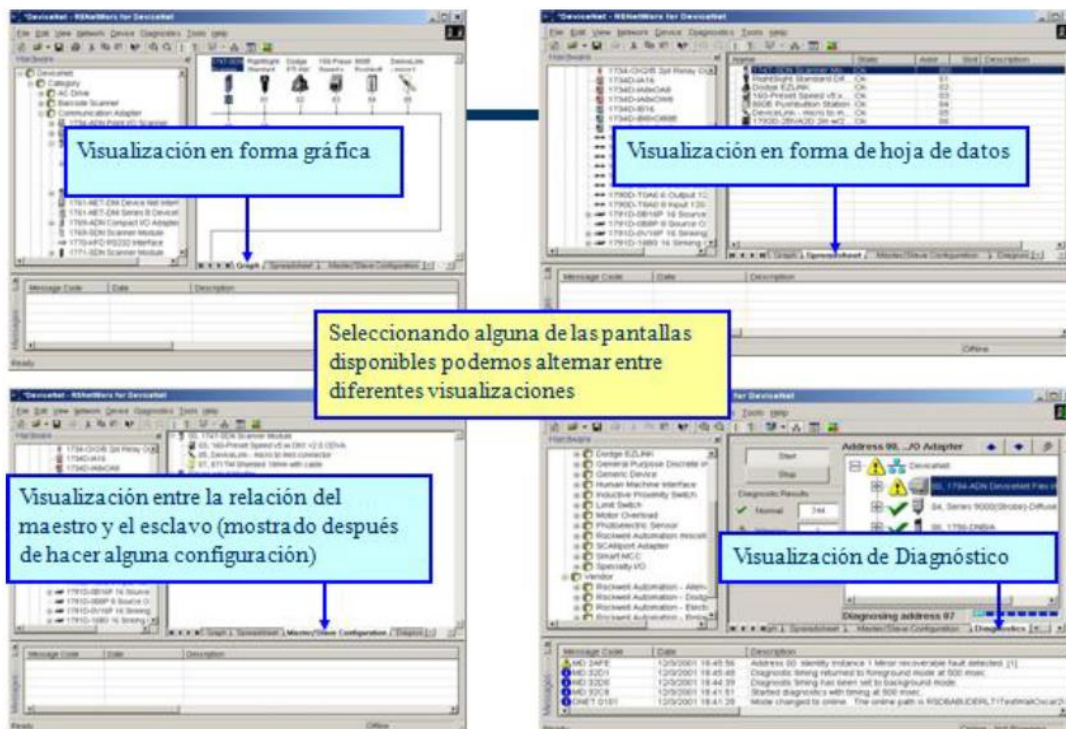


Fig. 3.4 Tipo de visualización de los dispositivos.

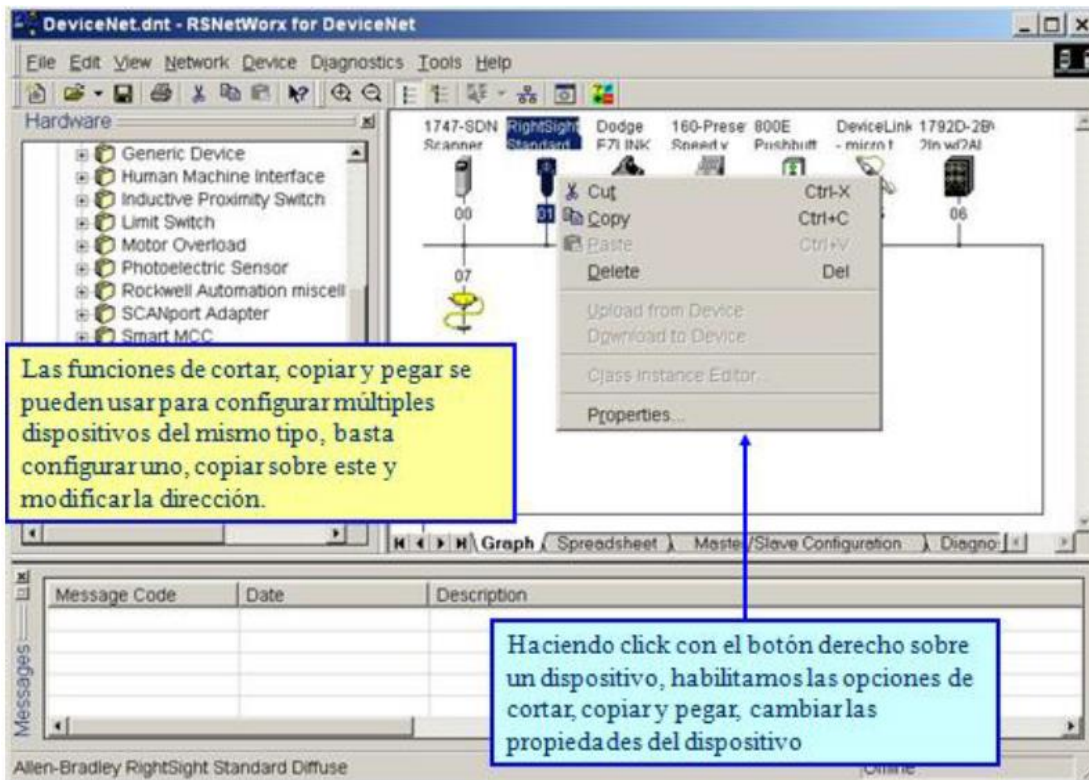


Fig. 3.5 Configuración para Dispositivos basados en archivos EDS

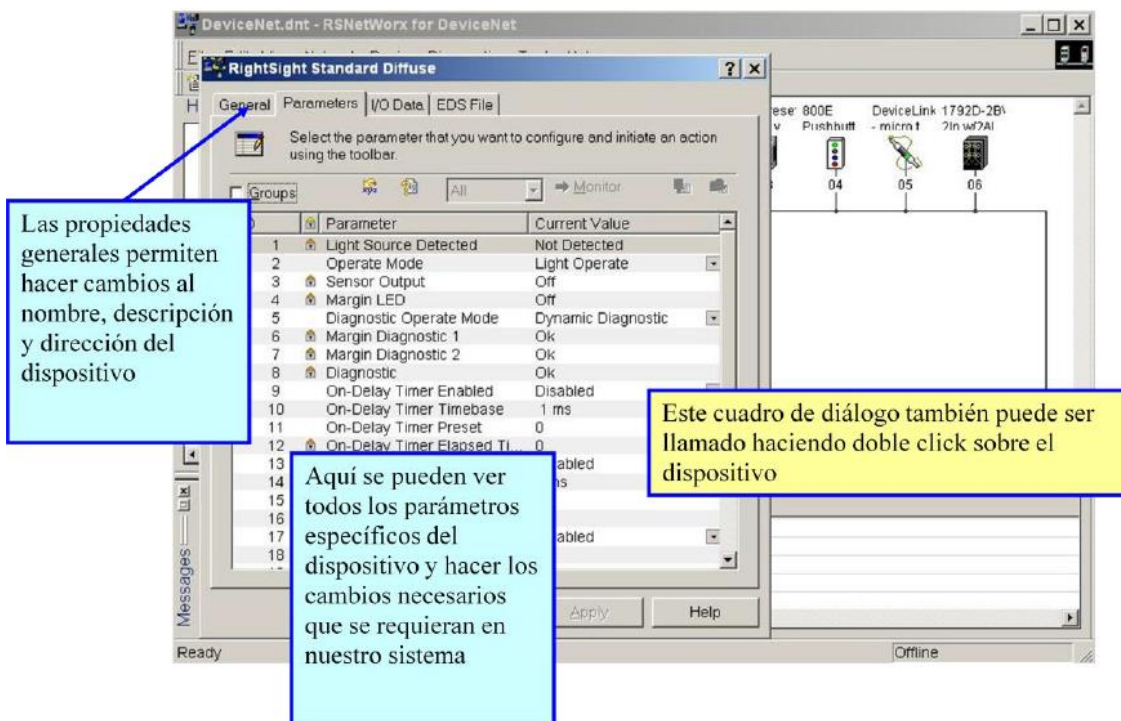


Fig.3.6 Configuración de las propiedades de los dispositivos.

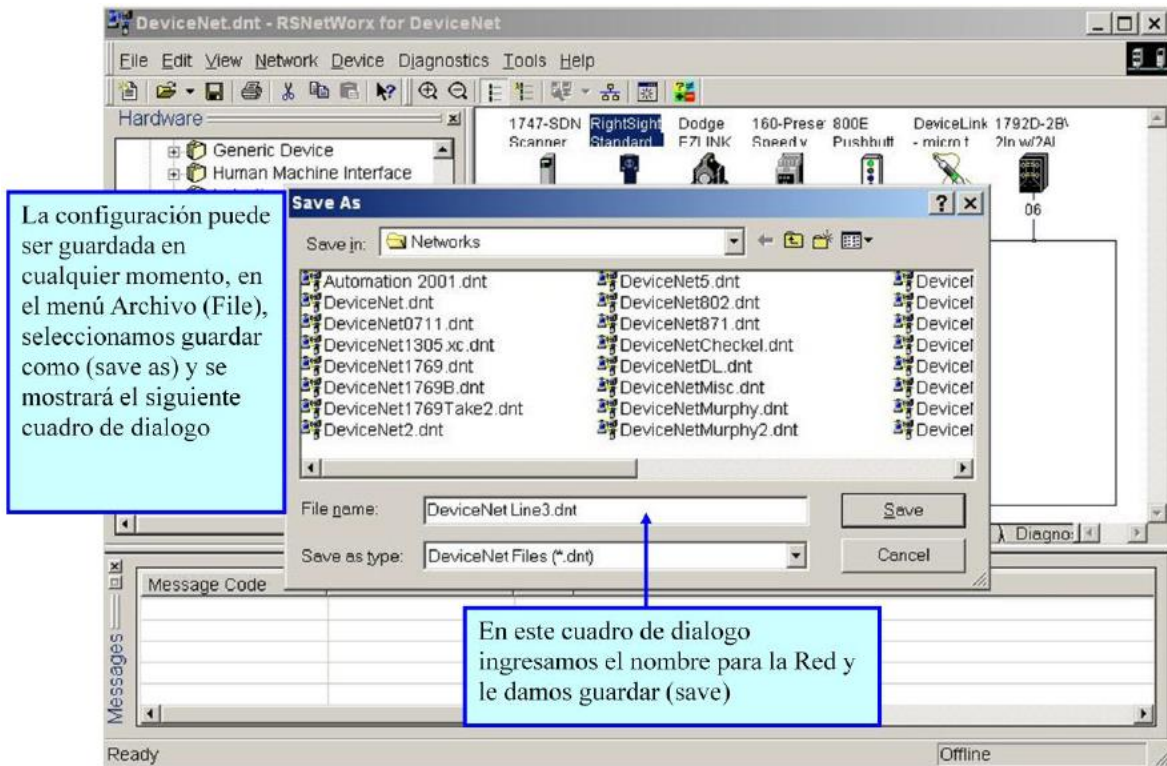


Fig. 3.7 Guardando la configuración.

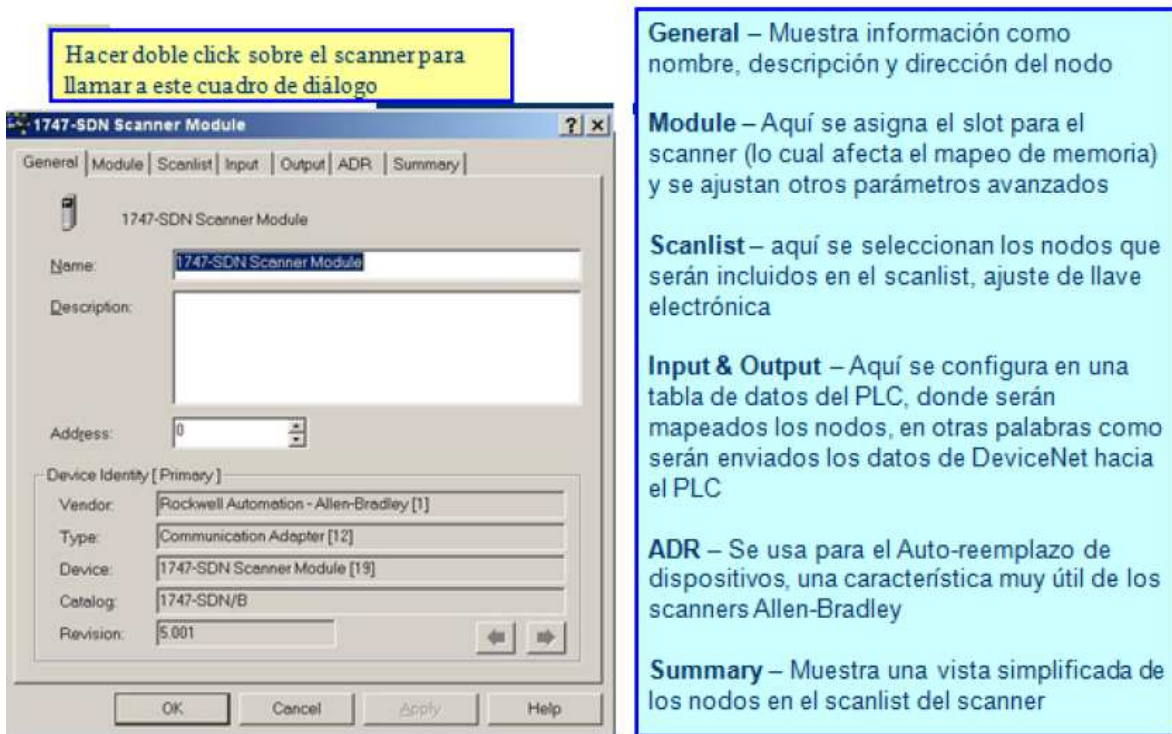


Fig. 3.8 Configuración del Scanner (1747-SDN).

El Scanner 1747-SDN (modulo del PLC Compact Logix) y el PLC (Compact Logix), estos dos dispositivos forman una sola unidad; es decir, el elemento maestro.

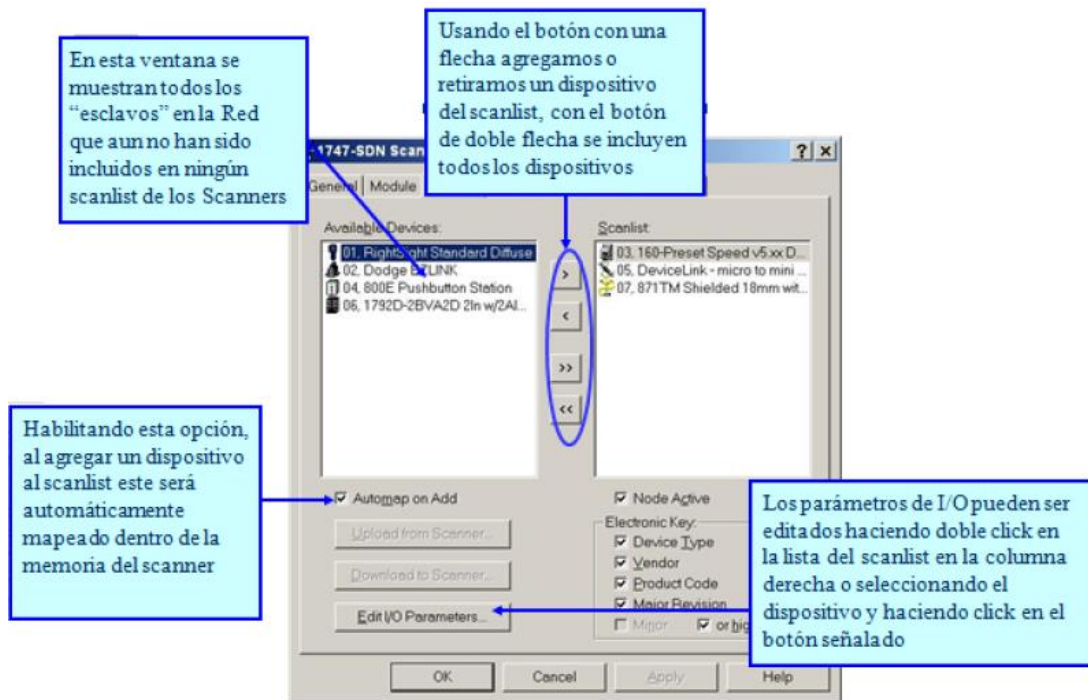


Fig. 3.9 Agregar y/o Quitar dispositivos de la red.

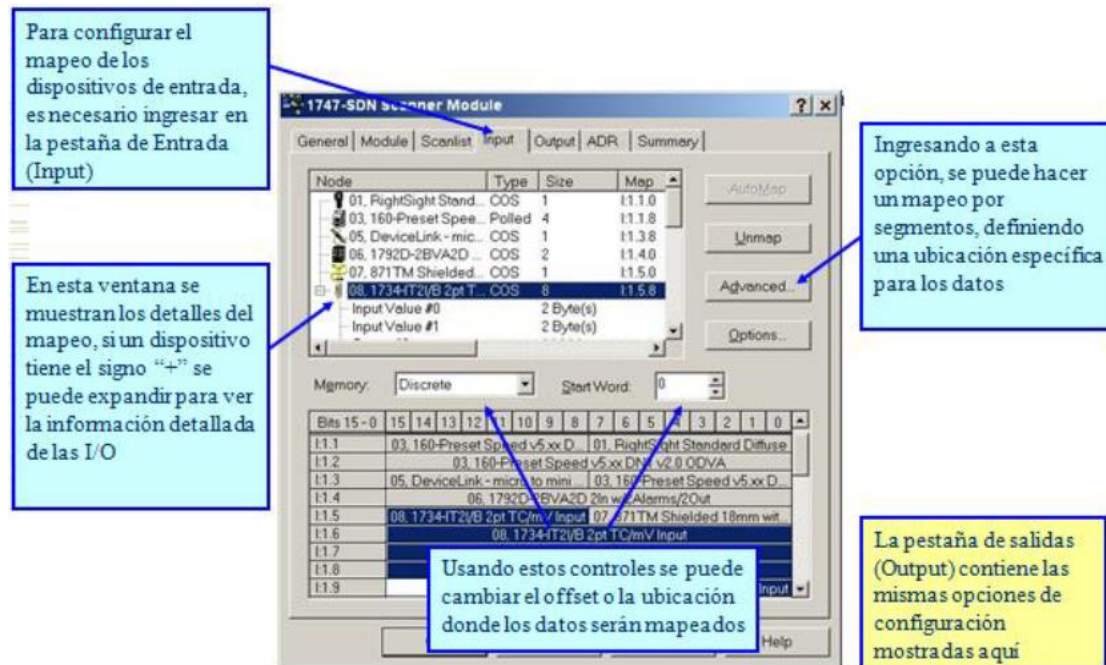


Fig. 3.10 Mapeo de los Dispositivos.

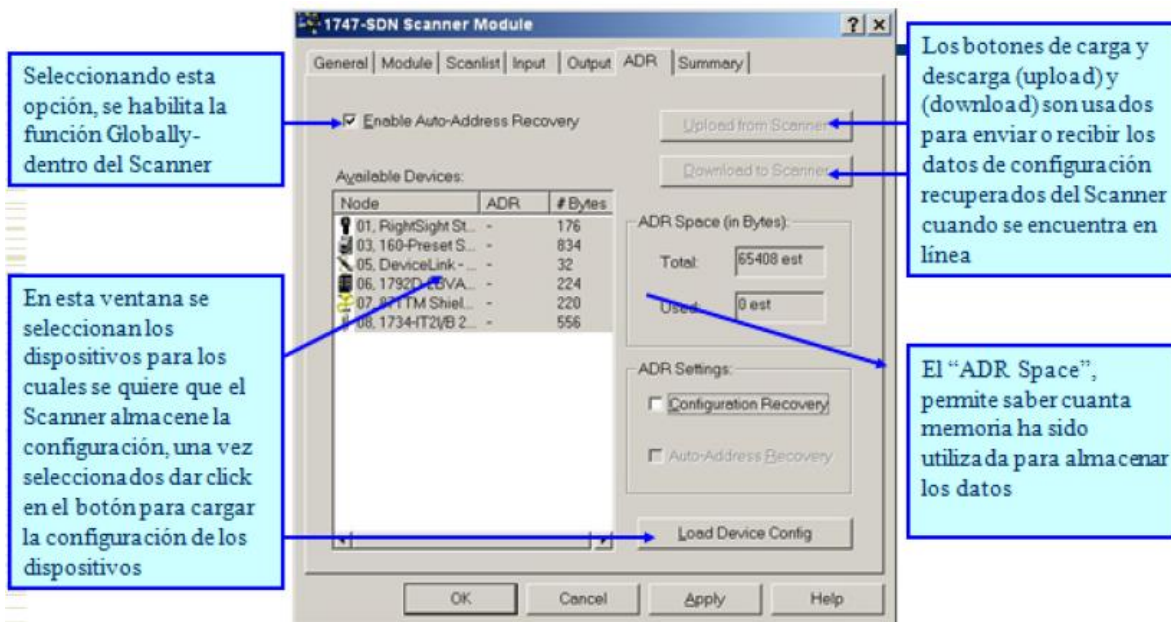


Fig. 3.11 Uso de la recuperación de configuración y “ADR”.

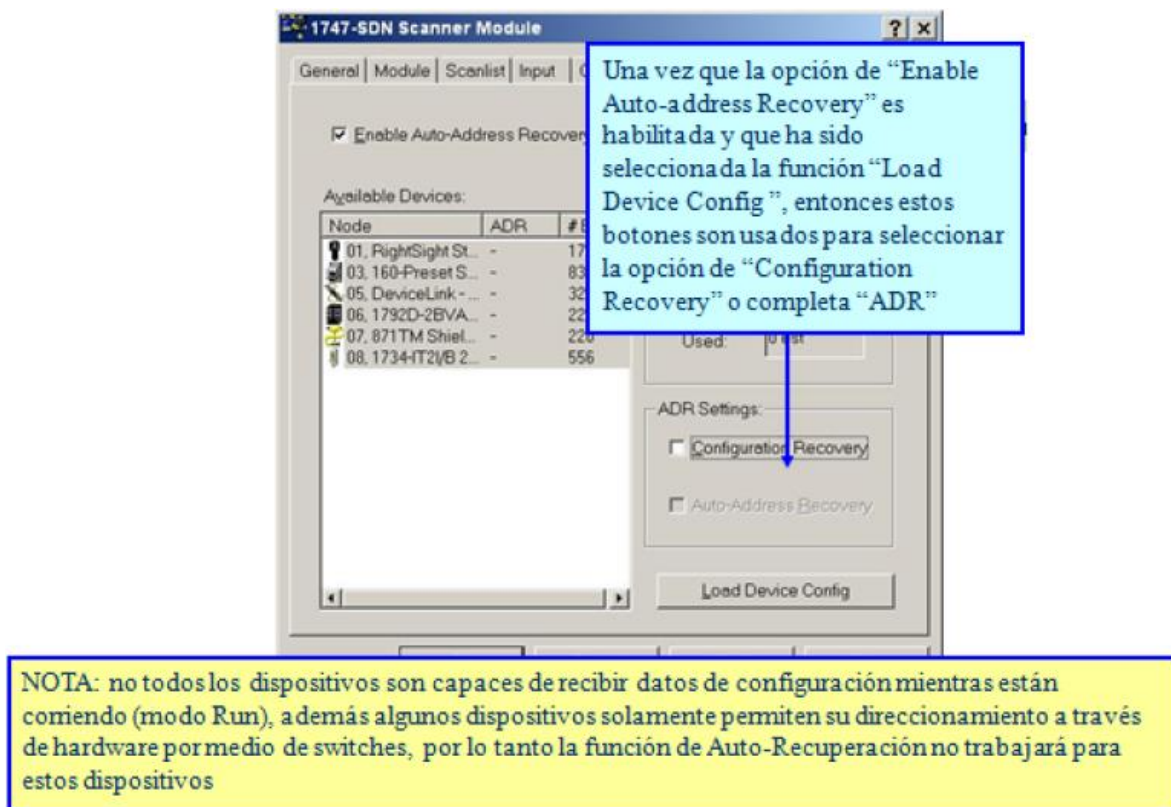


Fig. 3.12 Configuración Auto-Recuperación.

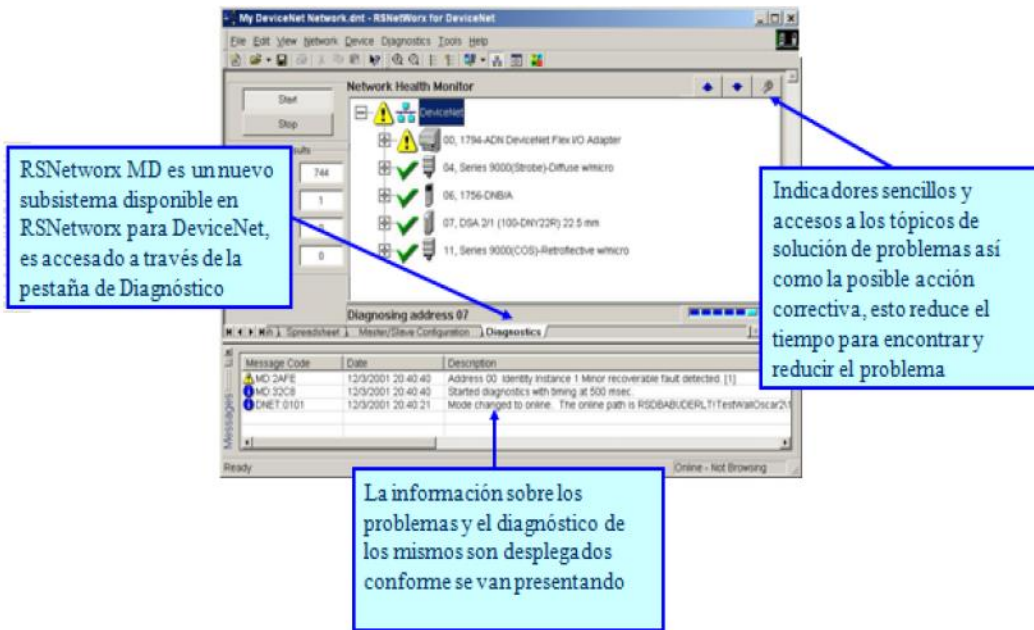


Fig. 3.13 Características de Mantenimiento y Diagnóstico.

3.3 Configuración de Entradas y salidas en el RSLogix 500

Una de las características avanzadas de RSLogix 500 es la capacidad de que el software de programación establezca una conexión de comunicaciones con el controlador y lea cuales módulos de E/S están conectados al controlador. Esta capacidad reduce significativamente la labro de configurar un sistema.

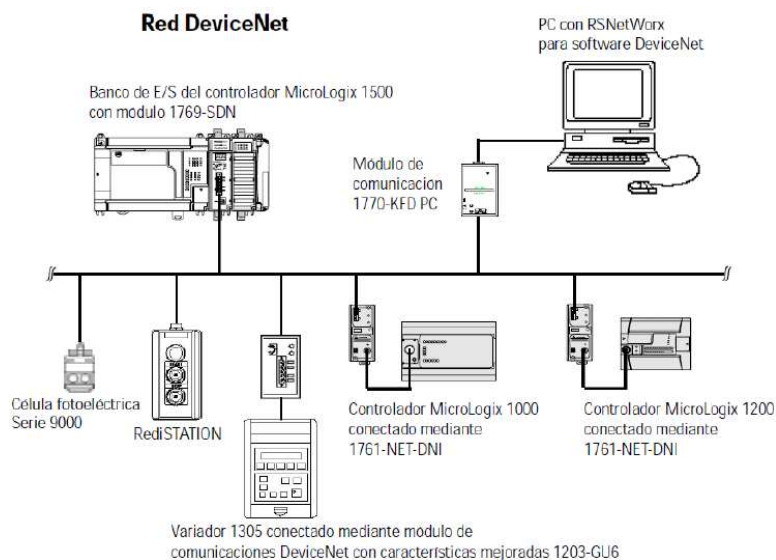


Fig. 3.14 Ejemplo de una red DeviceNet con controlador MicroLogix 1500

3.4 RSLogix 500

El RSLogix 500 permite la programación y configuración de los controladores lógicos programables de la serie CompactLogix. La configuración del controlador se refiere al ajuste de los parámetros de su hardware, tales como velocidad de comunicación del puerto serial, dirección IP para conexión Ethernet, velocidad de escaneo de datos, etc. La programación implica el desarrollo lógico y manejo de módulos en base a la aplicación, para realizar el control de las acciones.

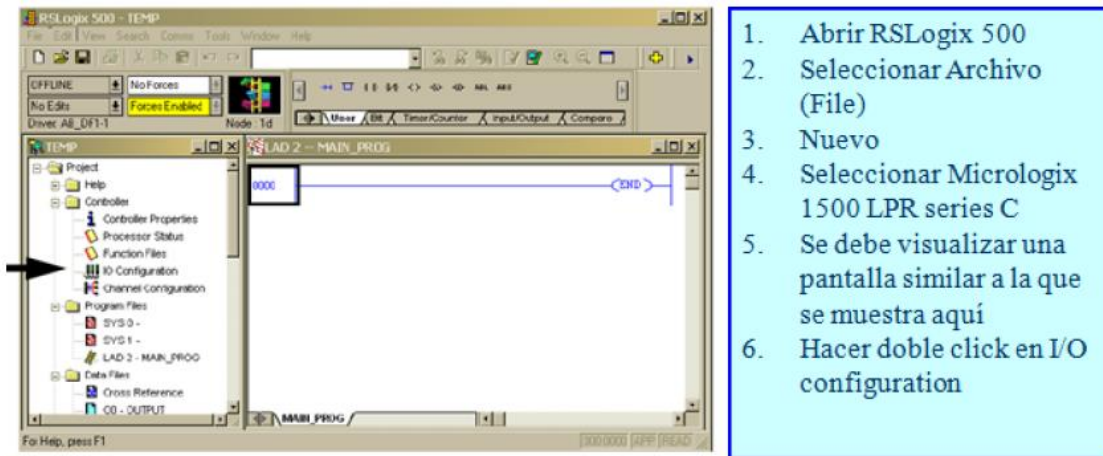


Fig. 3.15 Como iniciar un proyecto en RSLogix 500.

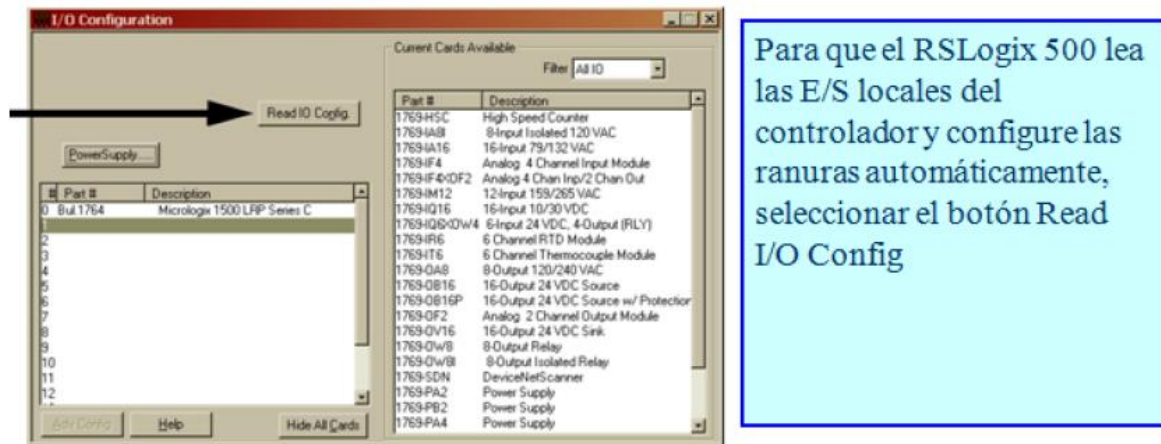
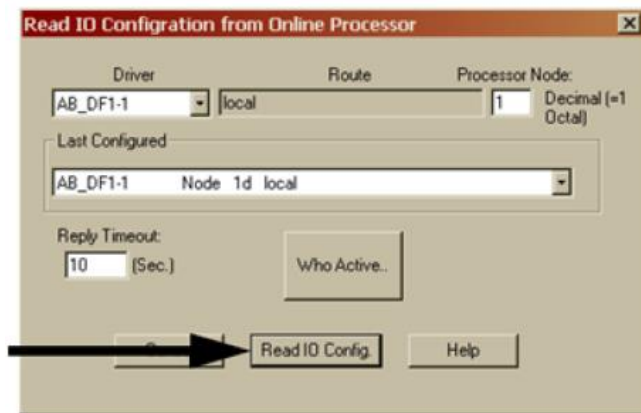
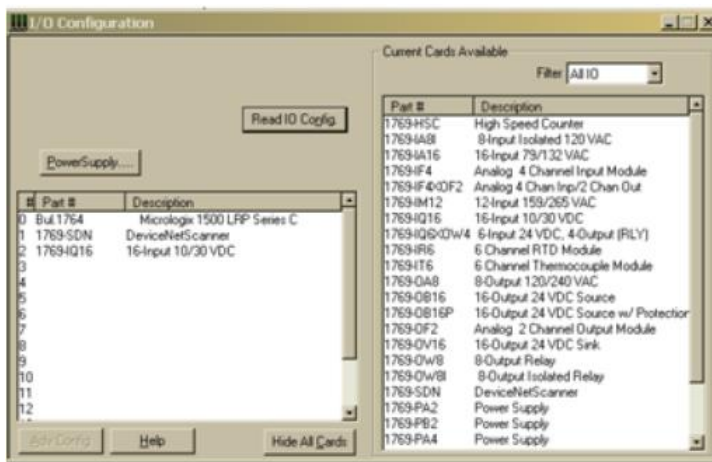


Fig. 3.16 Pantalla de Configuración de Entradas y Salidas.



La pantalla que aparece seguidamente es un diálogo de comunicación que permite seleccionar una ruta de comunicación al controlador MicroLogix usando RSLinx. Si ya se realizó previamente conexión a un controlador, el driver de comunicaciones que se usó será el driver activo. Esta pantalla de diálogo proporciona la capacidad de cambiar el driver o realizar la función **Who Active** en toda la red para ubicar el controlador MicroLogix específico. Si el driver y la ruta de acceso son correctos, seleccione **Read I/O Config**.

Fig. 3.17 Leer configuración de Entradas y Salidas.



Una vez leída la configuración, el RSLinx 500 muestra todos los módulos de E/S conectados al controlador MicroLogix.

En este ejemplo hay un módulo escáner 1769-SDN en la ranura 1 y un módulo de entrada discreta de 16 puntos en la ranura 2.

Fig. 3.18 Entradas y Salidas Instaladas.

3.5 Configuración del módulo escáner 1769-SDN

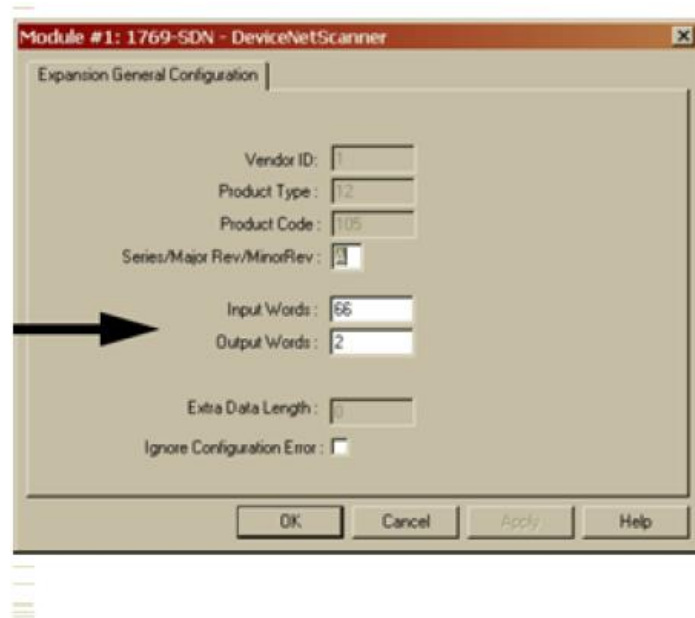
Palabras de entrada: Este es el número de palabras de entrada que el controlador asigna al módulo. El escáner requiere las primeras 66 palabras (0 a 65) para información de estado. Las palabras de datos de entradas esclavas DeviceNet comienzan en la palabra de ranura 66. Se puede

tener un máximo de 180 palabras en entrada para dispositivos esclavos DeviceNet (cantidad máxima de ranuras para entradas del módulo escáner 1769-SDN=246).

Palabras de Salida: Este es el número de palabras de salida que el controlador asigna al módulo. El escáner requiere las primeras 2 palabras (0 a 1) para información de estado. Las palabras de datos de salidas esclavas DeviceNet comienzan en la palabra de ranura 2. Se puede tener un máximo de 180 palabras de salida para dispositivos esclavos DeviceNet (cantidad máxima de ranuras para salidas del escáner=182).

Cambio de la configuración del 1769-SDN

Scanner DeviceNet 1769-SDN, su presencia en la red es indispensable, gestiona el flujo de datos entre el PLC y los dispositivos esclavos.



El cambio (adición o extracción) de la cantidad de datos que el controlador asignó al escáner se realiza en la pantalla de configuración de módulos de expansión. En RSLogix 500, seleccione las opciones **I/O configuration**, **Open** del módulo escáner 1769-SDN y cambie las palabras de entrada o salida según sea necesario. Haga clic en **Save** para guardar el programa y en **Download** para realizar la descarga al controlador.

Fig. 3.19 Cambio de la cantidad de datos.

3.6 Comunicación entre el PLC's

El scanner DeviceNet lo podemos usar para establecer comunicación entre 2 PLC conectados a la misma red.

Los requisitos son:

- Que cada PLC cuente con su propio scanner DeviceNet.
- Cada Scanner debe contar con un nodo propio dentro de la red.
- Habilitar del modo esclavo en alguno de los scanner.
- Configurar la red usando el modo esclavo.

Procedimiento

Una vez que se tiene la red armada y alimentada, con los requisitos anteriores, se procede a leer dicha red.

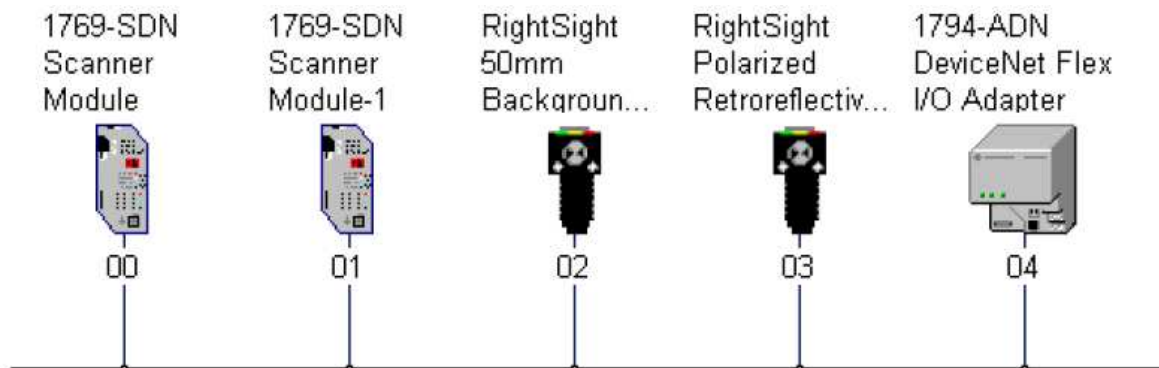


Fig. 3.20 Lectura de la red.

Después se procede a habilitar el modo esclavo en alguno de los Scanner como se indica a continuación

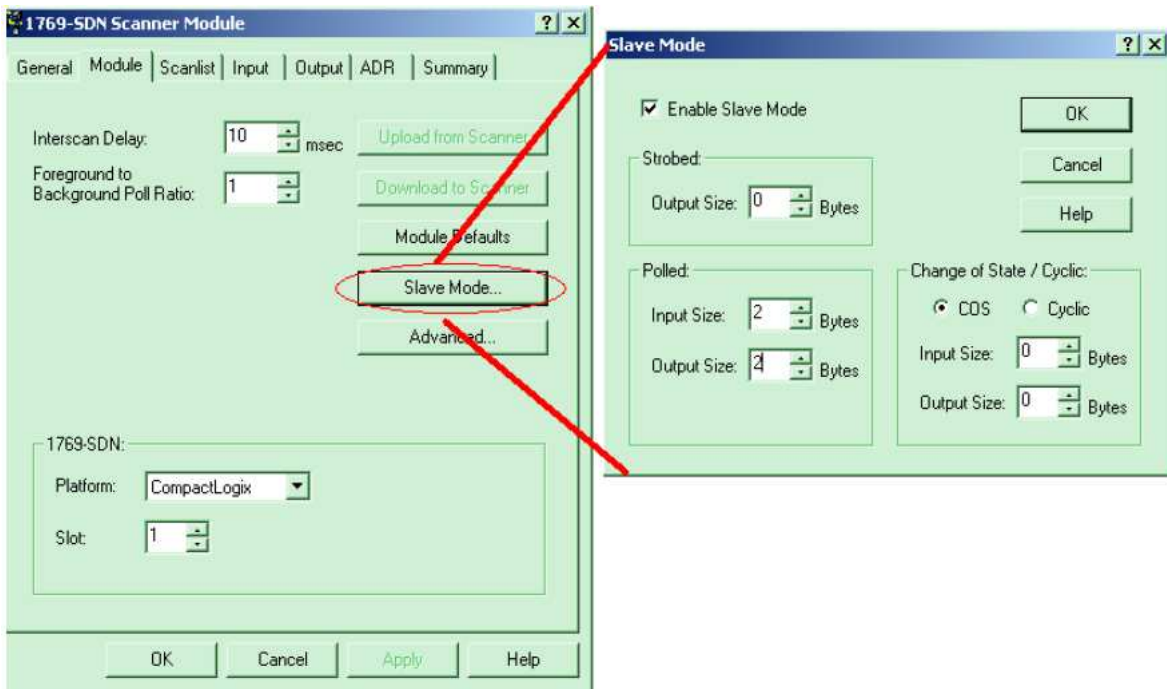


Fig. 3.21 en este módulo 1769-SDN solo son compatibles las formas de comunicación Polled y COS.

3.7 Mapeo de la Red

Scanner nodo 0, en él se mapean sus propias entradas y salidas (modo esclavo) para que estén disponibles para su PLC.

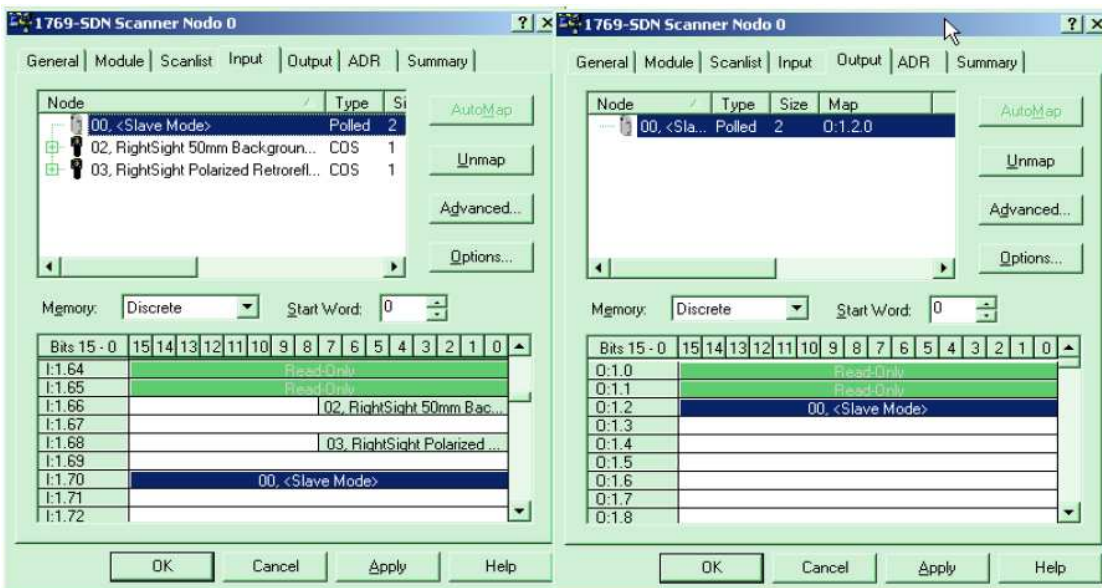


Fig. 3.22 Mapeo de entradas y salidas del scanner nodo 0.

Scanner 1, se mapea el IO del modo esclavo del scanner nodo 0, como cualquier otro dispositivo.

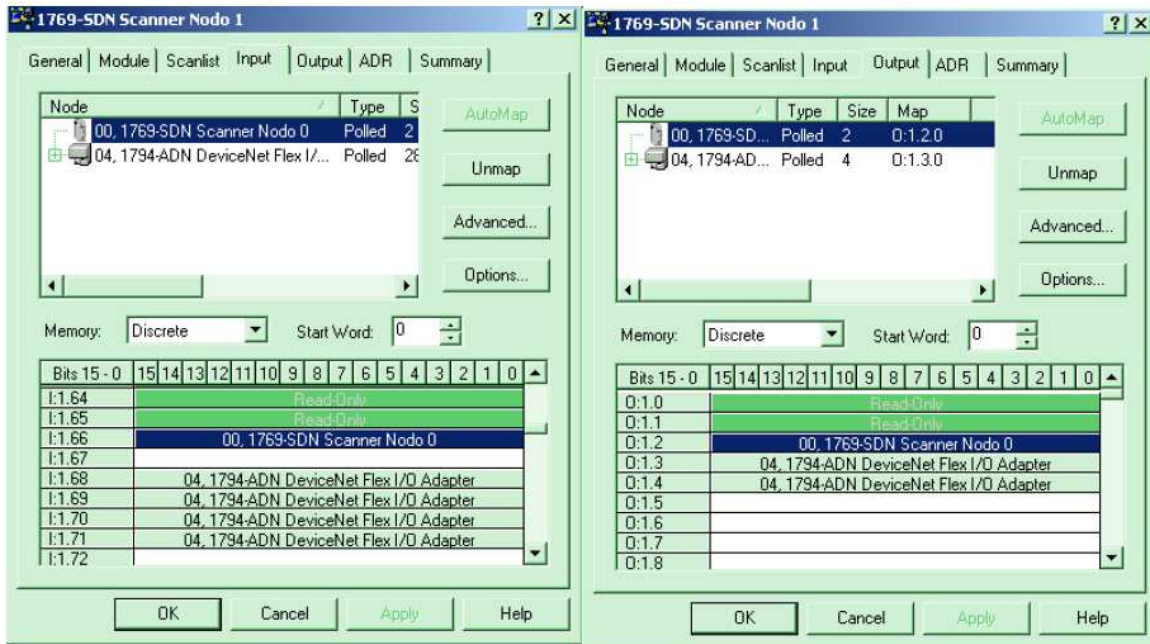


Fig. 3.23 Mapeo del modo esclavo del scanner nodo 0.

3.8 Configuración de módulos IO

Al dar click en propiedades, podemos hacer los ajustes en cada módulo.

En este caso el Módulo 1794-IE4XOE2/A que cuenta con 4 entradas y 2 salidas análogas, podemos ajustar el tipo de señal y rango.

Lo mismo hacemos para los módulos digitales. Para saber que rangos y parámetros soportan, recurrimos a los manuales de usuario de cada módulo.

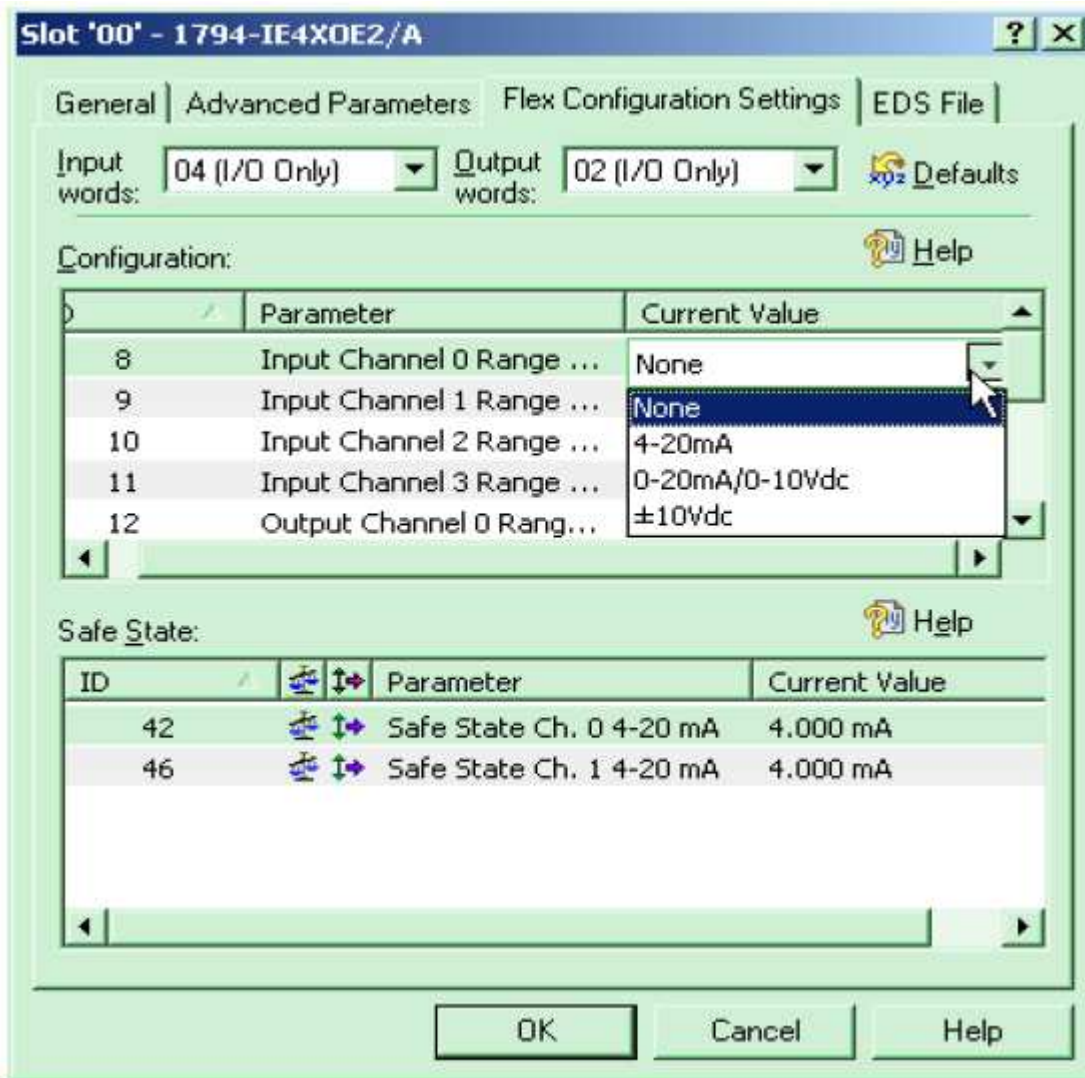


Fig. 3.24 Configuración módulos de I/O

Capítulo 4 Introducción a LabVIEW

LabVIEW es un entorno de programación de flujo de datos gráfica de National Instruments Corp. (NI). La relativa facilidad con la que un complejo software puede ser desarrollado utilizando técnicas de programación gráfica, junto con los dispositivos de hardware que ofrece NI y otros, ha hecho LabVIEW una herramienta de programación popular entre los científicos e ingenieros durante décadas. Las aplicaciones típicas de LabVIEW son los sistemas de control y adquisición de datos para la fabricación y las pruebas de laboratorio.

Aplicaciones de LabVIEW se escriben, o más precisamente, dibujado, en un "diagrama de bloques" y la interfaz de usuario se crean arrastrando y colocando controles e indicadores en el "Panel frontal". El Panel Frontal y Diagrama de bloques son dos partes de una entidad se define como un "instrumento virtual" (o "VI") y almacenados juntos en un solo archivo.

El Panel Frontal y Diagrama de bloques para una aplicación lógica muy simple se muestran abajo. La cifra también incluye las gamas de colores del panel frontal y Diagramas de bloques que contienen una gran selección de elementos de interfaz de usuario y las funciones y estructuras de software.

Aplicaciones de LabVIEW son típicamente escritos y ejecutados en las plataformas Windows, si bien las plataformas Mac OS y Linux también se pueden usar.

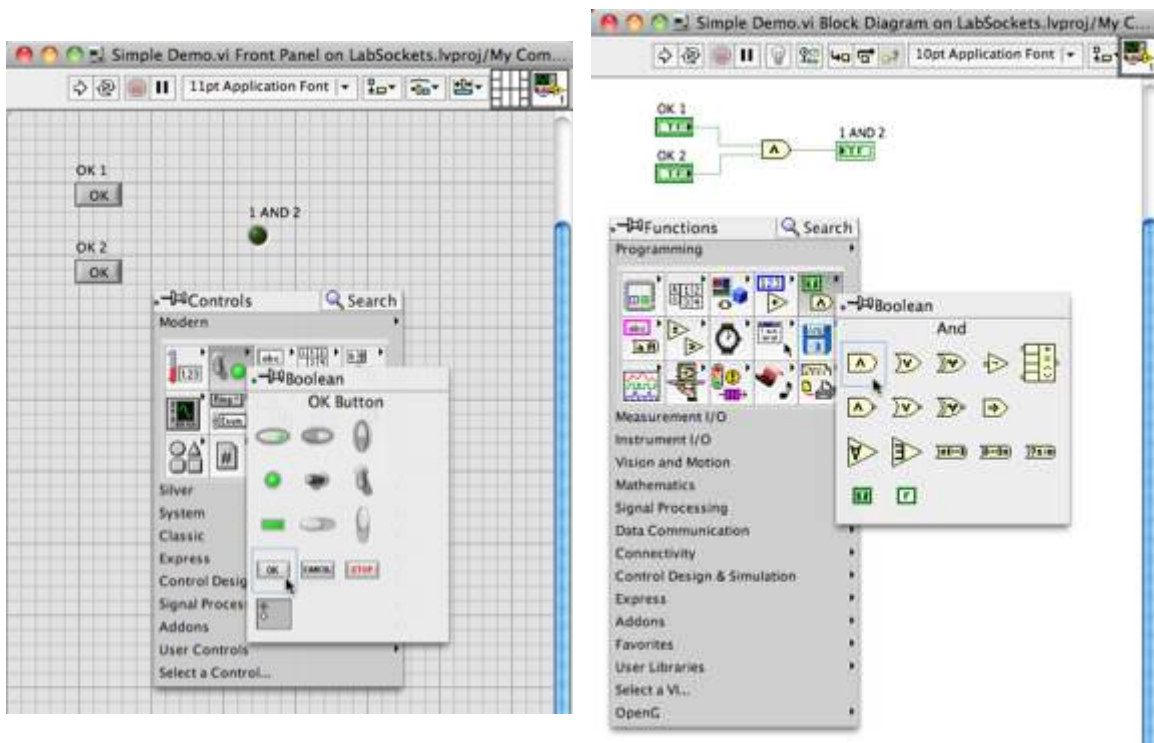


Fig. 4.1 Panel Frontal y Diagrama de bloques

4.1 Conceptos básicos

Instrumentación: conocimiento de la correcta aplicación de los equipos encaminados para apoyar al usuario en la medición, regulación, observación, transformación, ofrecer seguridad, etc, de una magnitud dada en cualquier proceso.

Instrumento de medición: Dispositivo destinado a ser utilizado para hacer mediciones, sólo o en conjunto, con dispositivos complementarios.

Instrumentación virtual: el concepto de instrumentación virtual nace a partir del uso de la computadora personal como “instrumento” de medición de tales señales como temperatura, presión, caudal, etc. El concepto de “instrumentación virtual” va más allá de la simple medición de corriente o voltaje, sino que también involucra el procesamiento, análisis, almacenamiento, distribución y despliegue de los datos e información relacionados con la medición de una o varias señales específicas.

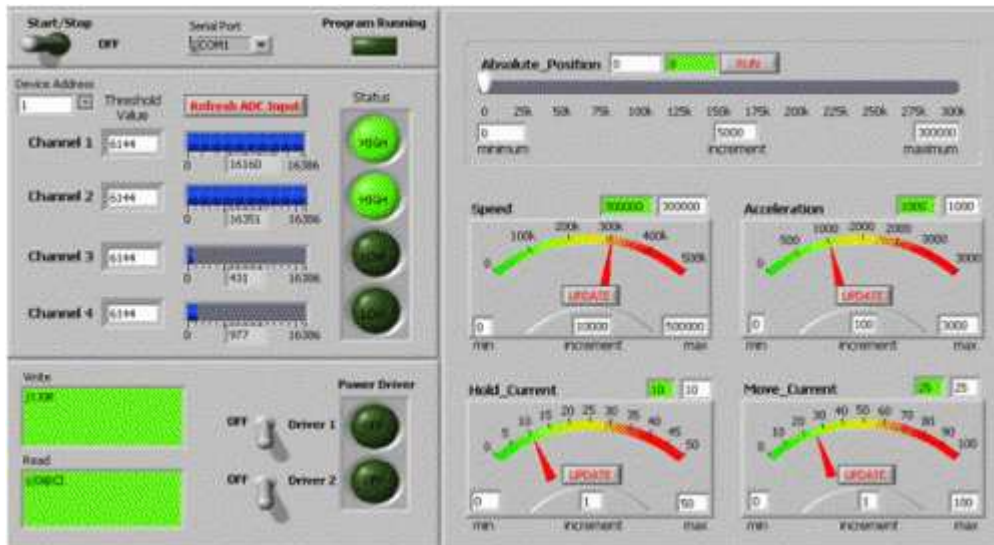


Fig. 4.2 Instrumento virtual en LabVIEW.

El instrumento virtual no se conforma con la adquisición de la señal, sino que también involucra la interfaz hombre-máquina, las funciones de análisis y procesamiento de señales, las rutinas de almacenamiento de datos y la comunicación con otros equipos.

4.2 Poder de la Programación Gráfica

El potencial de la programación gráfica para abordar el problema de la escasez de desarrollador es evidente cuando se considera una aplicación básica de prueba de matemáticas escrito en LabVIEW. El panel frontal y un fragmento del diagrama de bloques de la prueba se muestran a continuación. (Acceso a la prueba en un navegador web se discutirá con más detalle un momento.)

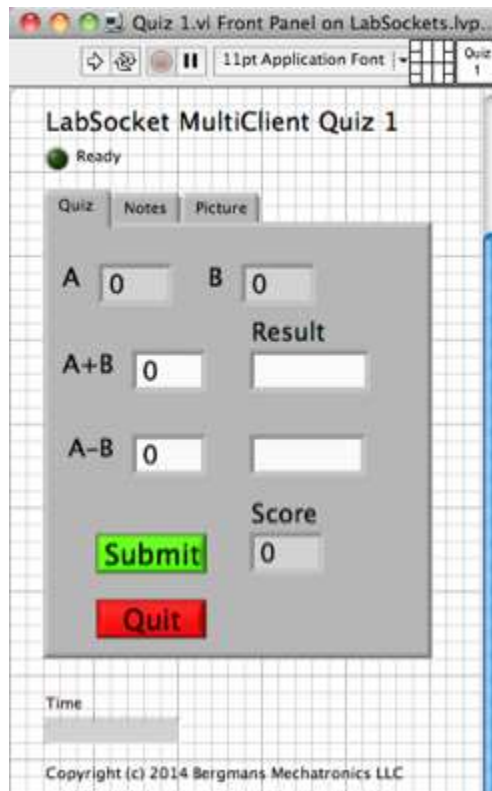


Fig. 4.3 Panel frontal de simple aplicación de preguntas

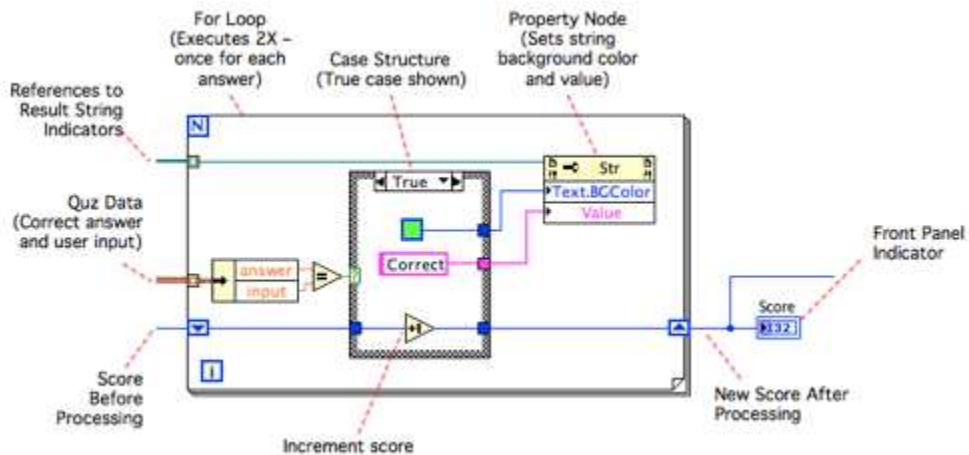


Fig. 4.4 Fragmento de Math Quiz Diagrama de bloques Mostrando Procesamiento de Datos Lógica.

El fragmento de código muestra la lógica utilizada para procesar los resultados de las dos preguntas simples de matemáticas que se presentan al usuario en el cuestionario. En el lado izquierdo del fragmento es una matriz de referencias a los indicadores de cuerda que se muestran los resultados de cada pregunta de la prueba. Dentro de cada iteración del bucle For, la respuesta correcta y la respuesta del usuario se introducen en un comparador. La salida del comparador determina entonces cómo se actualizan la interfaz de usuario y los datos de la puntuación:

Respuestas del partido - caso "True" se ejecuta. Las entradas al nodo de propiedad se establecen de forma que el color de fondo de la cadena de respuesta se establece en verde y la cadena "correcta" durante ese resultado. La puntuación del usuario también se incrementa.

Las respuestas no coinciden - caso "False" (no se muestra) se ejecuta. Las entradas de nodo de propiedad se establecen de manera que el color de fondo para la cadena de respuesta se establece en rojo y "Wrong" se muestra la cadena. La puntuación del usuario no se cambia.

Al salir del bucle For, la puntuación del usuario también se actualiza la interfaz de usuario.

Este fragmento de código ilustra uno de los más notables beneficios de un lenguaje gráfico de flujo de datos como LabVIEW - el código para una tarea moderadamente complicada (por ejemplo, el procesamiento de datos; actualizaciones de la interfaz de usuario, y, de gestión estatal) es fácil de desarrollar, entender y modificar. Este código puede ser fácilmente comprendido incluso por un

programador novato y se podría crear a partir de cero por alguien con conocimientos de programación modestos y un nivel básico de conocimiento de LabVIEW.

La extensión de esta capacidad para el desarrollo de una aplicación web sólo requiere el uso del sistema de LabSocket, como se describe en las dos secciones siguientes.

4.3 HMI



Fig. 4.5 Plataforma LabVIEW

Para construir una interfaz hombre máquina, solo requerimos de una PC, una tarjeta de adquisición de datos con acondicionamiento de señales (PCMCIA, ISA, XT, PCI, etc.) y el software apropiado, los tres elementos clave en la conformación de un HMI.

En el HMI, el software es la clave del sistema. Con este sistema indicado, podríamos construir un osciloscopio “personalizado”. Con la interfaz gráfica que uno desee, agregándole inclusive más funcionalidad.

Sin embargo, este mismo sistema puede también ser utilizado en la medición de temperatura, o en el control de arranque/parada de una bomba centrífuga. Es allí donde radica uno de los principales beneficios de la interfaz hombre-máquina, su flexibilidad.

4.4 Estructura del software labVIEW

LabVIEW es un lenguaje de programación de alto nivel, un entorno de programación gráfica en el que se pueden crear aplicaciones de una forma rápida, sencilla y enfocada al uso en instrumentación.

National Instruments es la empresa desarrolladora de tipo gráfico, y enfocada al uso en instrumentación.

Pero como lenguaje de programación, debido a que cuenta con todas las estructuras, puede ser usado para elaborar cualquier algoritmo que se desee, en cualquier aplicación, como en análisis, telemática, juegos, manejo de textos, etc.

Los programas en LabVIEW son llamados instrumentos virtuales, o VI, debido a que su experiencia y operación imita a los instrumentos físicos, tales como osciloscopios y multímetros. LabVIEW posee un conjunto extenso de herramientas para adquisición, análisis, despliegue y almacenamiento de datos, así como herramientas para ayudarle a resolver problemas con su código.

Los VI's de LabVIEW poseen tres componentes el panel frontal, el diagrama de bloques y el icono y panel conector.

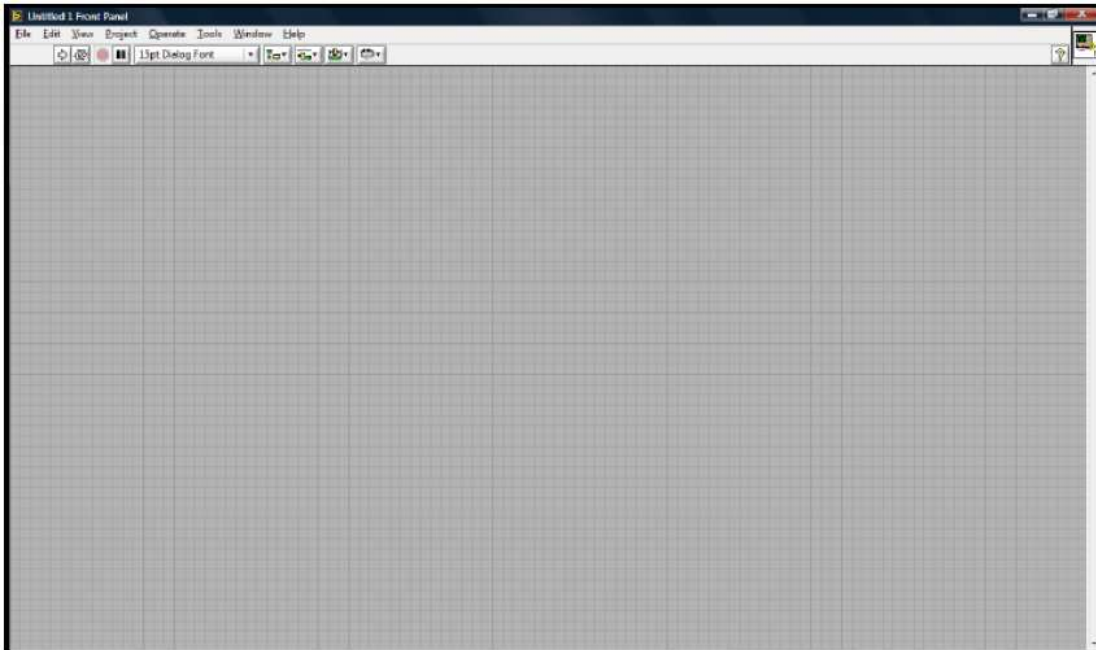


Fig. 4.6 Panel Frontal.

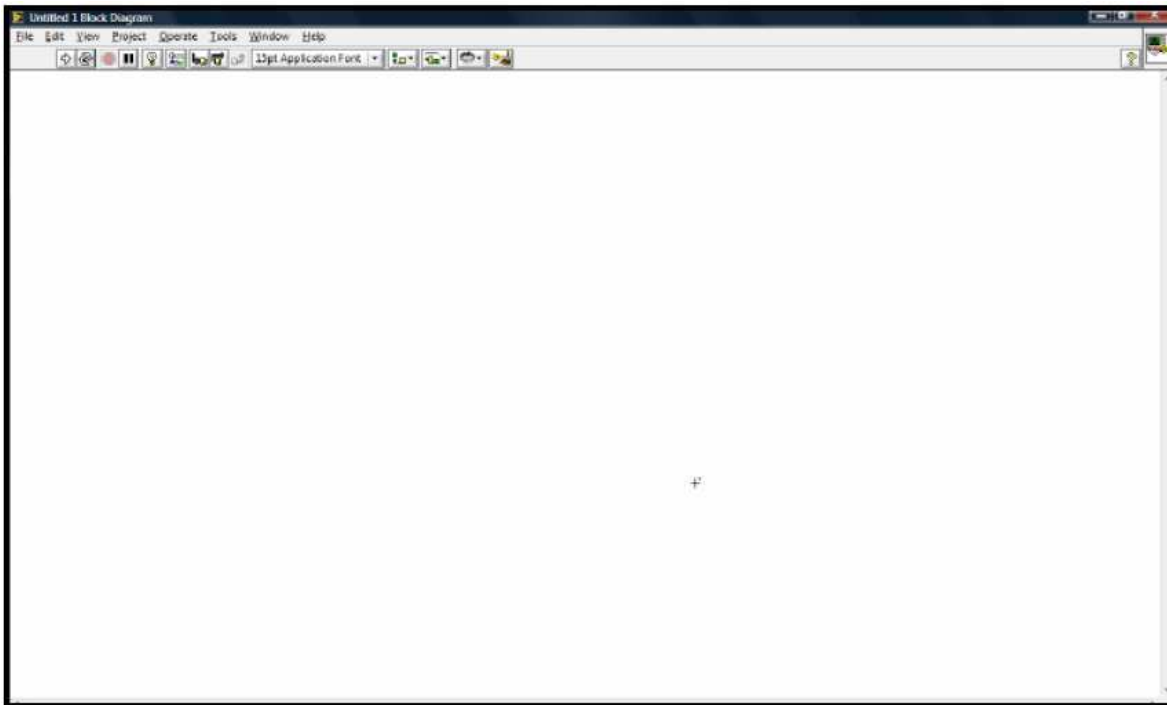


Fig. 4.7 Diagramas de bloques.



Fig. 4.8 Panel del conector

4.5 Protocolos de comunicación

4.5.1 Intercambio Dinámico de Datos (DDE)

Es una tecnología de comunicación entre varias aplicaciones bajo Microsoft Windows, DDE permite que una aplicación abra una sesión con otra, enviar comandos al servidor de aplicaciones y recibir respuestas.

Aunque es apto para las últimas versiones de Windows, ha sido reemplazado por su mucho más poderoso sucesor OLE (Objeto Linqueado e incrustado) y OLE automation, sin embargo, todavía se usa en varios sitios dentro de Windows, por ejemplo en la asociación de archivos. En particular, este no permite incorporar una interfaz del servidor dentro de la aplicación cliente, tampoco soporta la incorporación de un servidor de datos dentro del archivo cliente (por ejemplo: almacenamiento estructurado); y para usar DDE se tienen que conocer los comandos de DDE que el servidor soporta, lo cual no ha sido generalmente estandarizado. Así, para emplear toda la funcionalidad del DDE, se debe agregar código especial en cada aplicación cliente para cada servidor que este quiera controlar, o la aplicación cliente debe facilitar un lenguaje de script o macro, un uso común de DDE fue para desarrollar aplicaciones personalizadas para controlar software disponible, ejemplo: una aplicación escrita en C o algún otro lenguaje debía usar DDE para abrir una hoja de cálculo Microsoft Excel y llenarla con datos, por medio de una conversación con Excel y el envío de comandos DDE.

4.5.2 Objeto Vinculado e incrustado (OLE)

OLE es un sistema de objeto distribuido y un protocolo desarrollado por Microsoft, esta tecnología permite que un objeto (como una hoja de cálculo) para ser incorporado (y vinculados) dentro de otro documento (un documento de procesador de textos).

OLE permite a un editor encargarse a otro la elaboración de parte de un documento y posteriormente volverlo a importar. Por ejemplo, un sistema de publicación de escritorio puede enviar un poco de texto a un procesador de textos o una imagen a un editor de bitmap usando OLE.

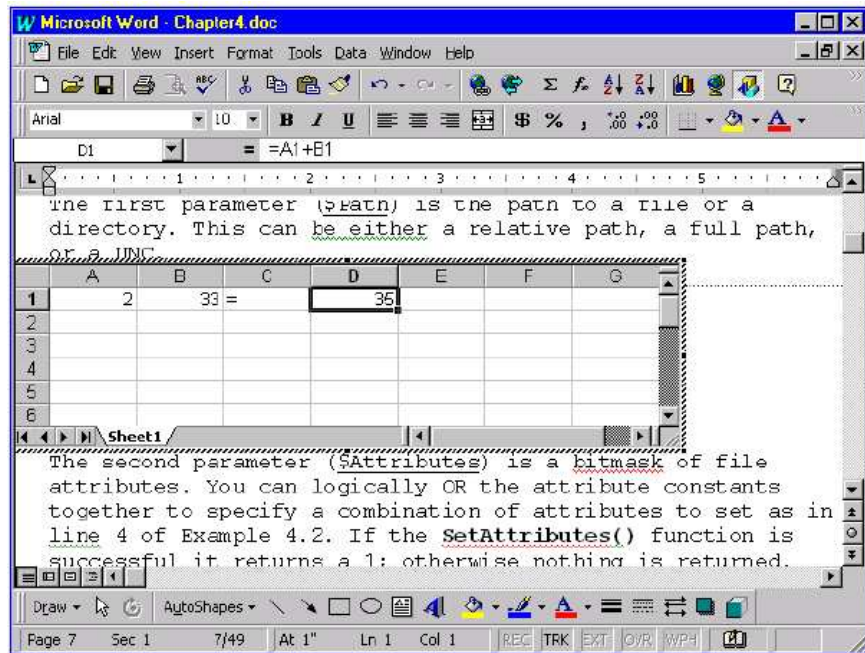


Fig. 4.9 Tecnología OLE

La ventaja principal de usar OLE, además de que el tamaño del archivo es menor, es la de poder crear un archivo principal. Se puede hacer una referencia a los datos de ese archivo, con lo que todo cambio posterior en el archivo principal se reflejará en el documento referenciado.

4.5.3 OLE Automation

En los programas de aplicación de Microsoft Windows, OLE Automation (posteriormente renombrado por Microsoft a Automation), es un mecanismo formal de comunicación entre procesos basado en COM (Component Object Model). Facilita una infraestructura que permite que aplicaciones llamadas controles Automáticos (Automation Controllers) puedan acceder, manipular y compartir objetos de automatización (por ejemplo, propiedades o métodos de otras aplicaciones).

El predecesor DDE (Intercambio dinámico de datos) fue un antiguo mecanismo de control entre aplicaciones. Como con DDE, en OLE automation el controlador es el “cliente” y, la aplicación que exporta los objetos de automatización, el servidor.

4.5.4 OLE para control de proceso (OPC)

Este producto especifica los parámetros necesarios para la comunicación en tiempo real entre diferentes aplicaciones y diferentes dispositivos de control de diferentes proveedores.

Las aplicaciones necesitan una manera común de acceder a los datos de cualquier fuente, como un dispositivo o una base de datos.

OPC es un “protocolo” de comunicaciones abierto que permite la comunicación entre aplicaciones informáticas y que permite la interoperabilidad entre diferentes fabricantes de software y hardware. El estándar OPC consta de varias especificaciones que permiten la obtención y envío de datos en tiempo real (DA), datos históricos (HDA), alarmas y eventos (A&E), y varios otros menos utilizados.

Los fabricantes de software no tienen que adaptar los drivers ante cambios de hardware.

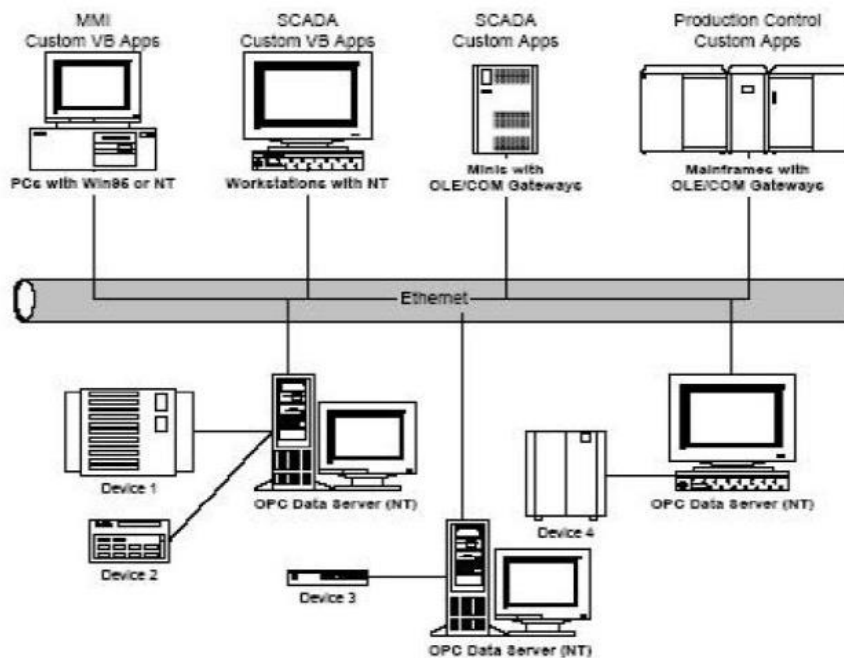


Fig. 4.10 OPC-integración de sistemas

Especificación	Contenido
OPC Introducción	Descripciones generales y aplicación de las especificaciones OPC.
OPC Definiciones comunes e Interfaces	Definiciones de términos utilizados en las especificaciones OPC.
OPC Acceso de datos	Definición de un interfaz para la lectura y escritura de datos en tiempo real.
OPC Alarmas y Eventos	Definición de un interfaz para el monitoreo de alarmas y eventos.
OPC Datos Históricos	Definición de un interfaz para el acceso de datos históricos.
OPC Seguridad	Definición de un interfaz para la utilización de políticas de seguridad.
OPC y XML	Integración del OPC y el XML para construcción de aplicaciones Web.
OPC Intercambio de Datos (DX)	Comunicaciones entre servidores – servidores de datos en los procesos.
OPC Interfaz y comandos de Ejecución	Definición de un interfaz para el intercambio de comandos y su ejecución.

Fig. 4.11 Especificaciones más importantes de la tecnología OPC

Capítulo 5 LabSocket

5.1 Introducción

El sistema LabSocket permite el acceso remoto a las aplicaciones de LabVIEW de escritorio o los navegadores web para móviles, sin necesidad de plugins del navegador o un motor de tiempo de ejecución de cliente. El sistema crea automáticamente una representación la página web de una aplicación de LabVIEW ("Target VI") del panel frontal y dinámicamente sincroniza el contenido de la página web y el panel frontal.

LabSocket-Basic es la más sencilla de las tres versiones Labsocket disponibles .En esta versión, un conjunto de VI's que implementan el sistema de LabSocket básico (el "LabSocket Soporte VIs") funciona en paralelo a la aplicación de LabVIEW está extendiendo a la web (el "Objetivo VI"). El Soporte LabSocket VIs son llamados mediante la colocación de un solo VI en el diagrama de bloques del Objetivo VI.

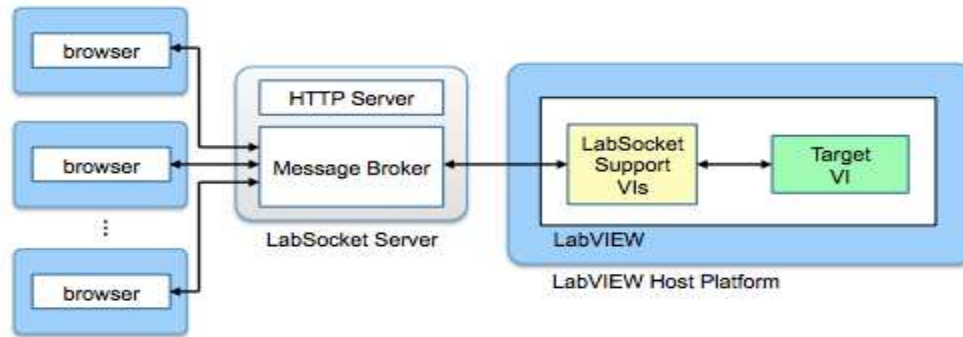


Fig. 5.1 LabSocket-Basic-Extender una Meta VI a uno o varios navegadores.

5.2 Teoría de Operación

El Soporte LabSocket VIs operan en dos modos. En primer lugar, en el modo **Screenscrape**, se genera código HTML y JavaScript que representa el panel frontal Objetivo VI y publicados en un servidor HTTP.

Después de realizar el screenscrape, el sistema entra en modo de **sincronización dinámica**. En este modo, las propiedades de los elementos en el panel frontal del objetivo VI se monitorizan continuamente. Cuando se produce un cambio a una propiedad de un elemento, un mensaje que contiene la información actualizada propiedad se transmite a través del socket TCP / IP a un intermediario de mensajes. El intermediario de mensajes reenvía este mensaje mediante una conexión WebSocket al navegador que a su vez actualiza la propiedad elemento navegador correspondiente. Cuando el usuario cambia el valor de un control en el navegador, se envía una actualización a través de WebSocket y socket TCP / IP para el Apoyo LabSocket VIs. El Soporte VIs luego actualizar el elemento de panel frontal correspondiente en el Target VI.

En otras palabras siguiendo la referencia de la figura a continuación el sistema de la nube al cliente LabSocket-Basic funciona como sigue. Al iniciar el sistema, la interfaz de LabSocket realiza una screenscrape de la VI que el desarrollador desea extender a uno o más navegadores. Este VI se refiere como el "Target VI". El screenscrape implica la topografía de los elementos del panel frontal del VI Target y generar una página web en forma de HTML y JavaScript de código que representa el panel frontal Objetivo VI. Los datos de la página web se publica a continuación para el cliente "Servidor Cloud". Cuando un navegador se apunta a esta página en el servidor, una versión estática

del panel frontal se representa en el navegador. Tenga en cuenta que la nube de servidor del cliente puede ser cierto servidor de la nube, como una instancia de Amazon Web Services EC2, o un servidor físico situado dentro de la red LAN del cliente.

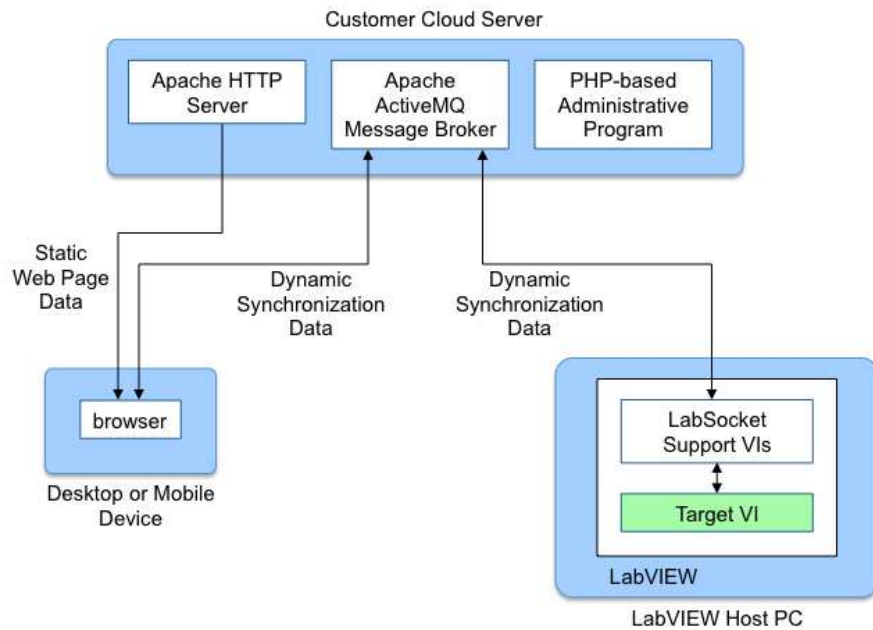


Fig. 5.2 Panorámica LabSocket-Basic, sistema Nube Cliente

Después de realizar el screenscape, la Interfaz LabSocket entra en el modo de sincronización dinámica. En este modo, las propiedades de los elementos en el panel frontal del objetivo VI se monitorizan continuamente. Cuando se produce un cambio a una propiedad de un elemento, un mensaje que contiene la información actualizada de cada propiedad se transmite de inmediato a través del socket TCP / IP a un intermediario de mensajes ActiveMQ en el servidor del cliente. El intermediario de mensajes reenvía este mensaje mediante una conexión WebSocket al navegador que a su vez actualiza la propiedad elemento navegador correspondiente.

Los eventos generados por el usuario en el navegador se transmiten a la Target VI de una manera similar. Cuando el usuario actualiza el valor de un elemento en el navegador, un controlador de eventos basado en JavaScript envía un mensaje a través de WebSocket y socket TCP / IP a la interfaz LabSocket. La interfaz a continuación, actualiza el elemento de panel frontal correspondiente en el Target VI.

Los desarrolladores avanzados de LabVIEW apreciarán el hecho de que cuando la Interfaz LabSocket actualiza el valor de un elemento del panel frontal, también genera un evento de cambio de valor de ese elemento. Como resultado, las aplicaciones de LabVIEW que emplean una estructura de eventos responderán de forma idéntica a los eventos generados por el usuario en el panel frontal Objetivo VI o en el navegador.

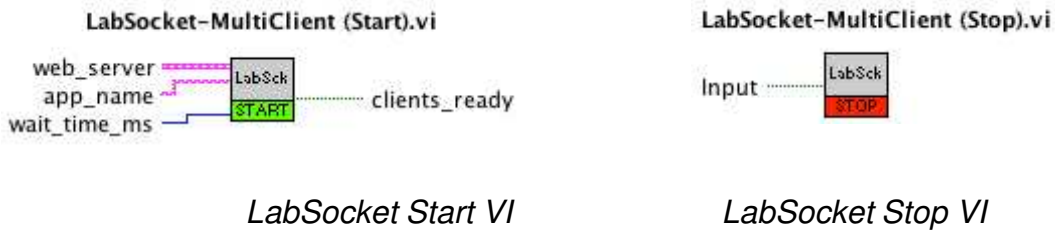


Fig. 5.3 VIs debe ser incluida en el diagrama de bloques Target VI para iniciar y terminar LabSocket Support Vis.

5.2.1 Cliente Mapping Target VI

Si más de un navegador se conecta simultáneamente a la Meta VI, los mismos contenidos aparecen en cada navegador y cada usuario puede controlar el destino VI. En algunas circunstancias, puede ser necesario para cada cliente de navegador para establecer una conexión con una instancia única de la Objetivo VI. LabSocket-MultiClient, se presenta a continuación, se ha desarrollado para cumplir con este requisito.

5.2.2 Servidor Labsocket

Para aplicaciones de producción, la versión comercial del software LabSocket normalmente utiliza el corredor del servidor HTTP y el mensaje contenido en un "LabSocket Server" máquina virtual (VM). Esta máquina virtual también se puede utilizar con el paquete de evaluación LabSocket. Uno de los beneficios de la utilización de esta máquina virtual es que permite que todo el sistema LabSocket que se configurar fácilmente en el seno de una red LAN del cliente. El servidor de VM LabSocket se suministra de forma gratuita con todas las versiones de evaluación y comerciales del sistema LabSocket.

Para el uso inicial del paquete de evaluación LabSocket o para la comprobación del sistema comercial, el corredor de servidor HTTP y mensajes en un servidor de nube existente operado por

Bergmans Mecatrónica LLC también se puede utilizar. Esta opción es ideal para iniciarse rápidamente con cualquier versión del sistema. El LabSocket Servidor sección contiene más información sobre el servidor VM LabSocket y ubicaciones físicas del servidor HTTP y el intermediario de mensajes.

5.2.3 MultiClient LabSocket

LabSocket-MultiClient proporciona una asignación de uno a uno entre cada cliente de explorador y una *instancia única* del Objetivo VI. Esta versión del sistema está diseñado para aplicaciones en las que los usuarios entran o recuperan información específica del usuario, como las encuestas, la entrada de datos, etc

5.2.4 Funcionalidad del MultiClient

Como se muestra en la figura siguiente, un VI de alto nivel conocido como el "VI Server" se utiliza para identificar el prototipo Objetivo VI y para invocar el Soporte LabSocket VIs utilizado por el sistema LabSocket-multicliente.

El Soporte LabSocket VIs realizan la misma función inicial Screenscrape como LabSocket-Basic. El comportamiento del sistema es ligeramente diferente durante la sincronización dinámica: Como cada cliente navegador se conecta al sistema, el Soporte VIs crear e iniciar una copia del prototipo Objetivo VI y luego mantener la sincronización entre el navegador y cada instancia de destino VI. Cuando un cliente de navegador desconecta del sistema, se detiene y se suprime la instancia de destino VI correspondiente.

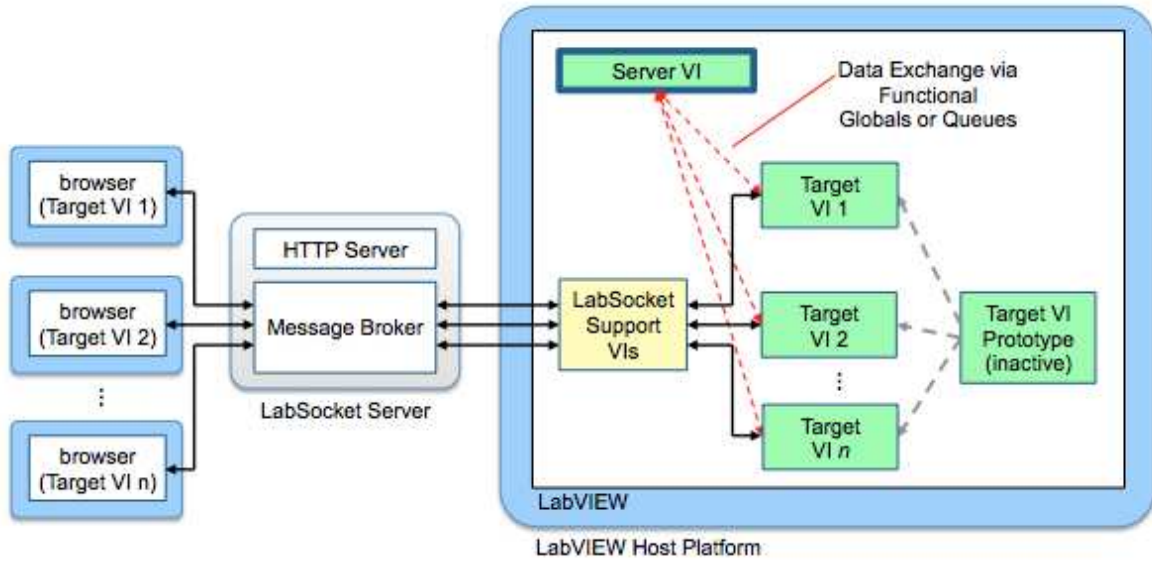


Fig. 5.4 LabSocket-MultiClient - Extender instancias únicas de un Meta VI Separar Navegadores.

5.3 Requisitos del sistema

Los navegadores que utilizan el sistema LabSocket deben ser compatibles con el HTML5 Estándar de interfaz de WebSocket. Afortunadamente, más moderno móvil y de escritorio navegadores son compatibles con este estándar. Una lista completa de WebSockets compatibles con navegadores se muestra en la siguiente figura:



Fig. 5.5 WebSockets compatibles con navegadores

Para los usuarios que requieren soporte para navegadores antiguos como Internet Explorer 9.0 y antes, un producto comercial de puerta de enlace puede ser proporcionado para emular el comportamiento de la interfaz WebSocket.

5.4 Instalación del software LabSocket-Basic

1. Start LabVIEW en la plataforma LabVIEW anfitrión.
2. Instalar paquete VIPM LabSocket-Basic
 - a. Iniciar VIPM seleccionando Herramientas> Administrador VI Package.
 - b. En la ventana del Administrador JKI Paquete seleccionar el archivo llamado VIPM - LabSocket Básica
 - c. En la ventana LabSocket paquete, haga clic en Instalar
 - d. VIPM mostrará LabSocket-Basic como el único elemento de una lista de productos destinados a ser instalados. Haga clic en Continuar
 - e. Después de la instalación, LabSocket se mostrará como el único elemento de una lista de resultados de la última acción. Haga clic en Finalizar
 - f. En la ventana LabSocket paquete, seleccione Archivo> Cerrar
 - g. El paquete LabSocket ahora debe ser visible en el paquete JKI VI Ventana del Administrador.
 - h. En la ventana del Administrador JKI VI Package, seleccione Archivo> Salir
 - i. Salir LabVIEW

5.5 Demostración de Aplicación de LabSocket

En esta sección se muestra cómo utilizar el sistema LabSocket acceder a un sencillo Objetivo VI desde un navegador.

1. Crear un nuevo LabVIEW VI. Desde la ventana principal de LabVIEW, seleccione Archivo> Nueva VI
2. Un componente clave del sistema LabSocket es un LabSocket VI llamado Start.vi. Este VI llama al Soporte LabSocket VIs que a su vez

- i) realizar la screenscape panel frontal; y,
- ii) sincronizar los paneles de navegador y los controles e indicadores frontales.

Haga clic en el diagrama de bloques del nuevo VI y seleccione Complementos> LabSocket> LabSocket Start.vi (Figura 5.6).

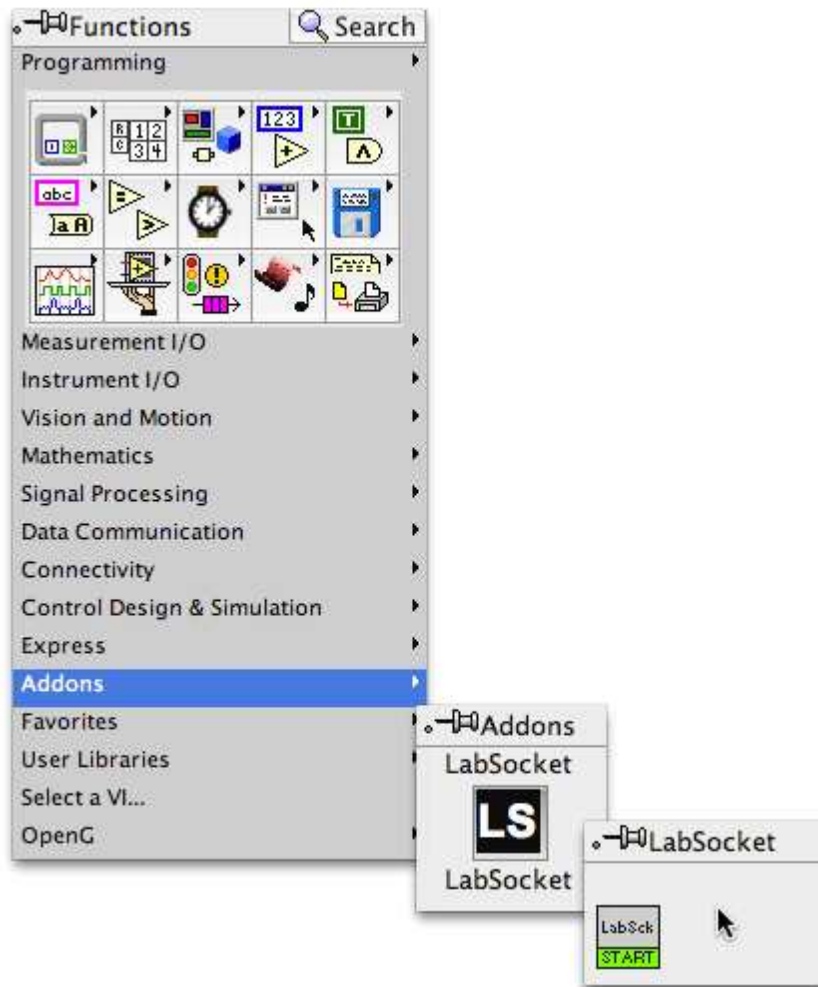


Fig. 5.6 Seleccionar LabSocket Start.vi en la paleta de funciones.

3. alambre los siguientes tres terminales de LabSocket Start.vi (Figura 5.7):

- Dirección IP - un clúster que contiene dos cadenas está conectado a esta entrada:
 - *LabSocket_Server* es la dirección IP del servidor LabSocket. dos opciones están disponibles aquí:
 - i) Introduzca la dirección IP del servidor virtual LabSocket Máquina relación a la plataforma LabVIEW Host.

ó

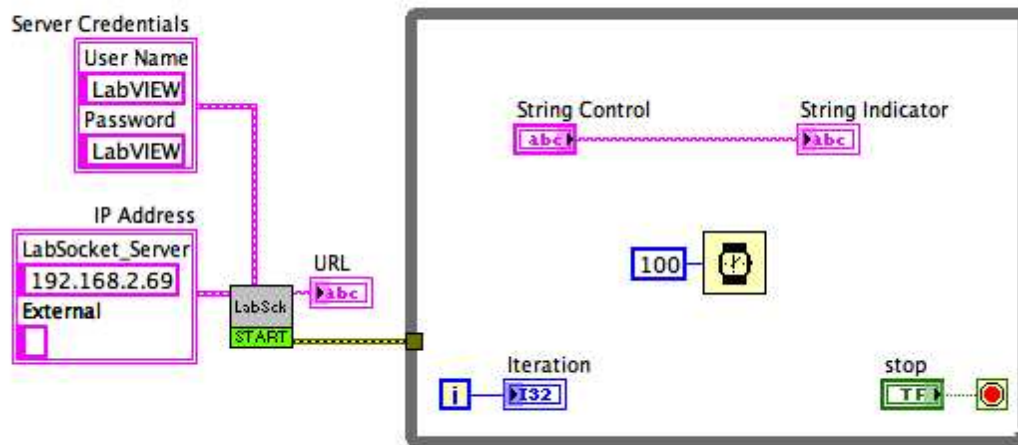
ii) Introduzca la cadena "labsocket.com" (sin comillas) para utilizar el software LabSocket Server en la nube existente de BML servidor. Esta opción está pensada para usuarios que quieren empezar a utilizar rápidamente el sistema y no para propósitos de producción.

- *External* es la dirección IP del servidor LabSocket VM en relación al navegador. Deje este valor en blanco si esta dirección es la misma que la dirección IP LabSocket_Server o estás usando BML de servidor de la nube.

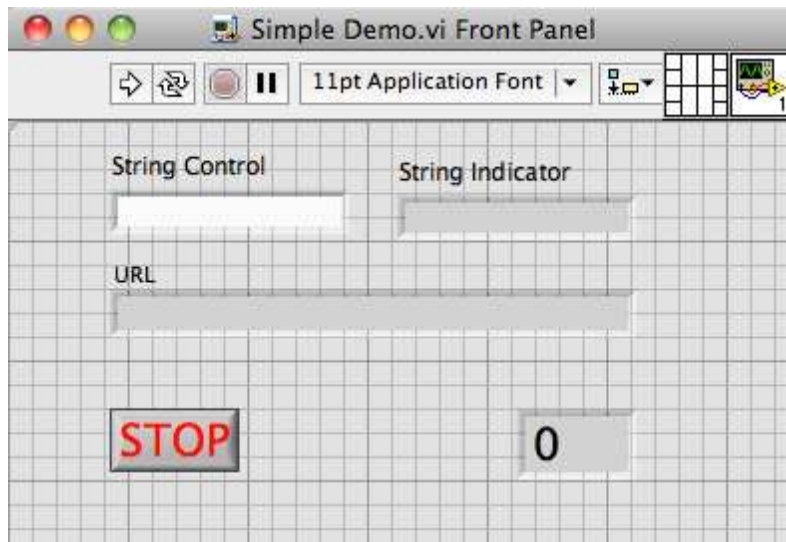
- Credenciales de servidor - un grupo de dos cadenas de texto que contienen cada uno la cadena "LabVIEW" está conectado a esta entrada. Este es el valor por defecto para esta entrada y coincide con la configuración predeterminada del intermediario de mensajes.

- URL - alambre indicador cadena a este terminal. Este indicador mostrará la URL a la que el destino VI se puede acceder con un navegador. El URL se basa en el nombre de archivo de la VI. Si "labsocket.com" se utiliza como la dirección IP del servidor LabSocket, un número de cuatro dígitos al azar es anexa a la dirección URL.

4. Añadir un bucle while, una Espere VI, y los controles e indicadores simples como se muestra a continuación (Figura 3). Cablear el terminal de error de LabSocket Start.vi a la borde del bucle while. Este fuerza de alambre LabSocket Start.vi para completar ejecución antes de los mientras que el inicio de bucle.



a) Diagrama de Bloques



b) Panel Frontal

Fig. 5.7 Simple – Demo.vi

5. establecer la acción mecánica de los botones para cualquiera "Switched Cuando se pulsa" o "Switched cuando Liberado". Guarde el archivo en el disco. En esta guía, él nombre de archivo Demo.vi simple se utiliza, aunque se acepta cualquier nombre.
6. Inicie el VI presionando el botón Run VI.
7. Si esta es la primera vez "LabSocket Start.vi" se ha quedado en esta plataforma, se le pedirá que introduzca una cadena de licencia. Introduzca la cadena de licencia proporcionado a usted por Bergmans Mecatrónica.
8. Aparecerá la ventana Estado LabSocket. La URL a la que el destino VI se puede acceder se mostrará en la página Registro de esta ventana (Figura 4). Esta URL que coincida con la salida de la terminal de URL de LabSocket Start.vi.

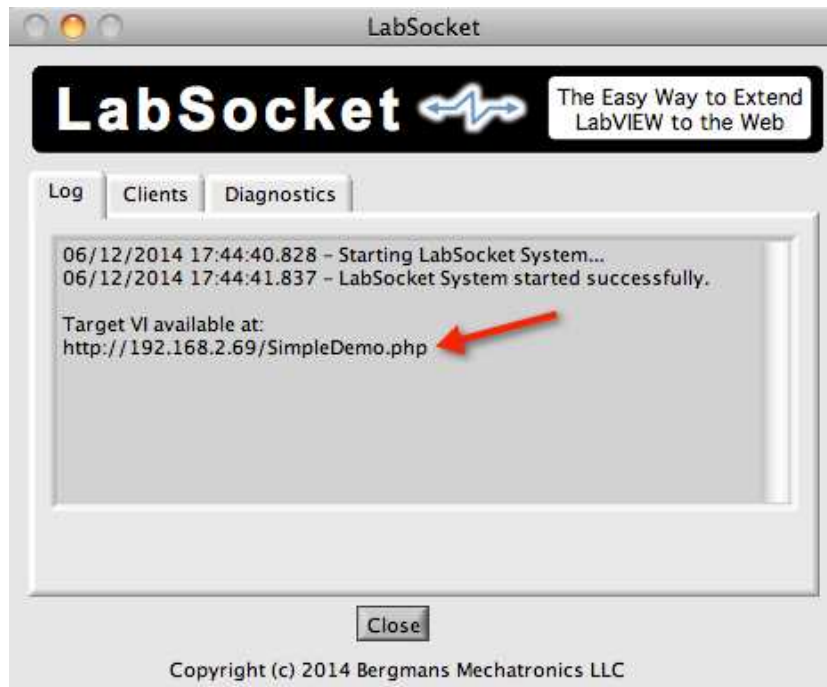


Fig. 5.8 LabSocket ventana de estado (Página de Registro)

9. Cuando un navegador se apunta a la URL que aparece en la LabSocket Ventana de estado, se mostrará automáticamente una representación del Frente Panel. Todos los controles e indicadores compatibles en el panel frontal también sincronizados con los del navegador (Figuras 5.9 y 5.10). Como parte del proceso de conectar el navegador a la VI, una ventana pop-up aparece en el explorador que muestra el nombre de la aplicación y el Número del Navegador Específico "ID de cliente" del navegador.

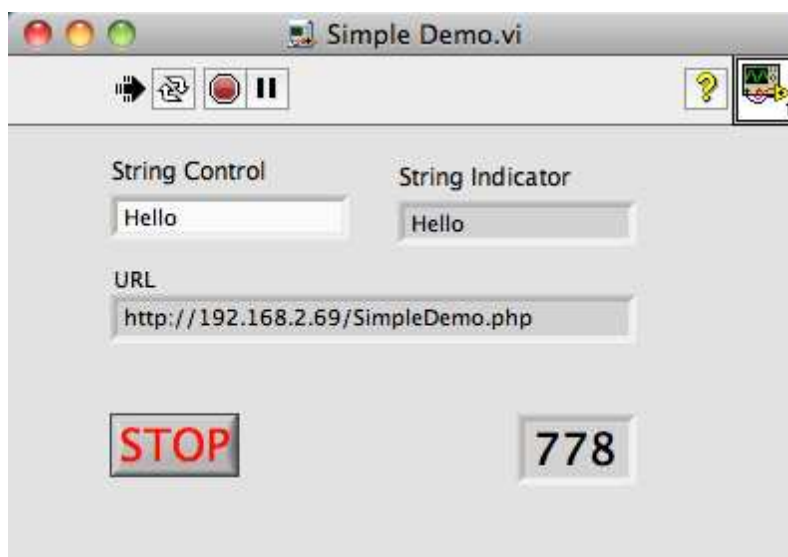


Fig. 5.9 Panel Frontal Target VI "SimpleDemo.vi"

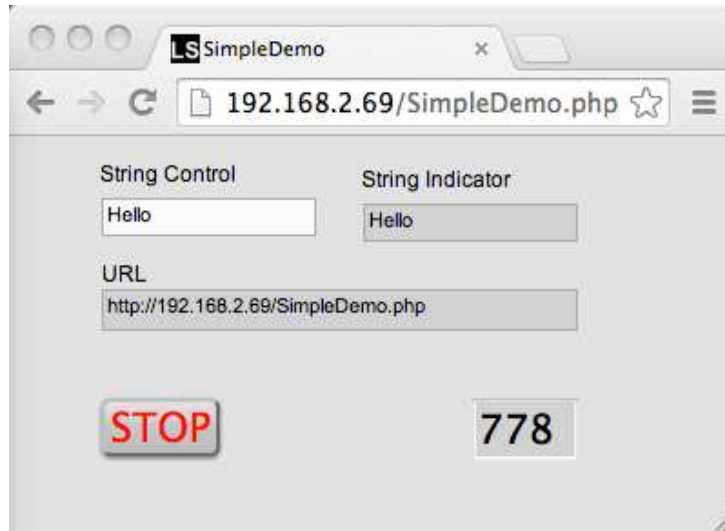


Fig. 5.10 Representación del Navegador de *Target VI* "SimpleDemo.vi"

10. La ventana Estado LabSocket mencionada anteriormente cuenta con otras dos páginas: La página Clientes (Figura 5.11) muestra el ID de cliente de cada uno conectado navegador en la columna "Browser ID". La columna "Last Heartbeat Echo" muestra la hora de la última respuesta del navegador para la inspección técnica periódica señales que se transmiten a cada navegador conectado. (Si un eco de señal no se devuelve desde un navegador en más de 10 segundos, el cliente se considera inactivo y elimina de la lista los clientes en la Ventana de estado.)

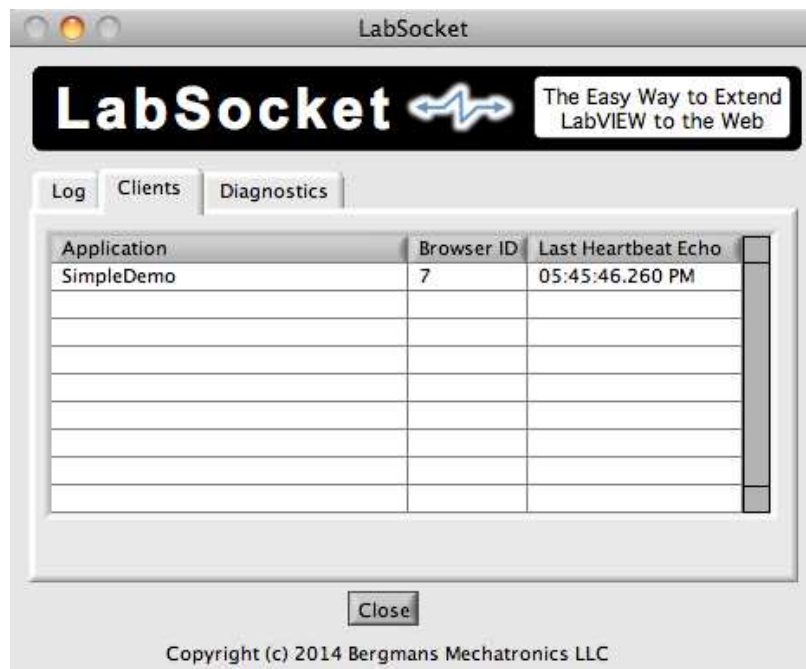


Figure 5.11 Ventana de Estado de LabSocket– Pagina para clientes

La página de diagnósticos (Figura 5.12) de la ventana de estado muestra LabSocket un valor "Período Loop". Este valor es el período de tiempo requerido para actualizar el panel frontal del objetivo VI con los cambios enviados desde el navegador más el tiempo necesario para detectar cambios en el panel frontal Target VI y enviar estos cambios a cada cliente de navegador conectado.

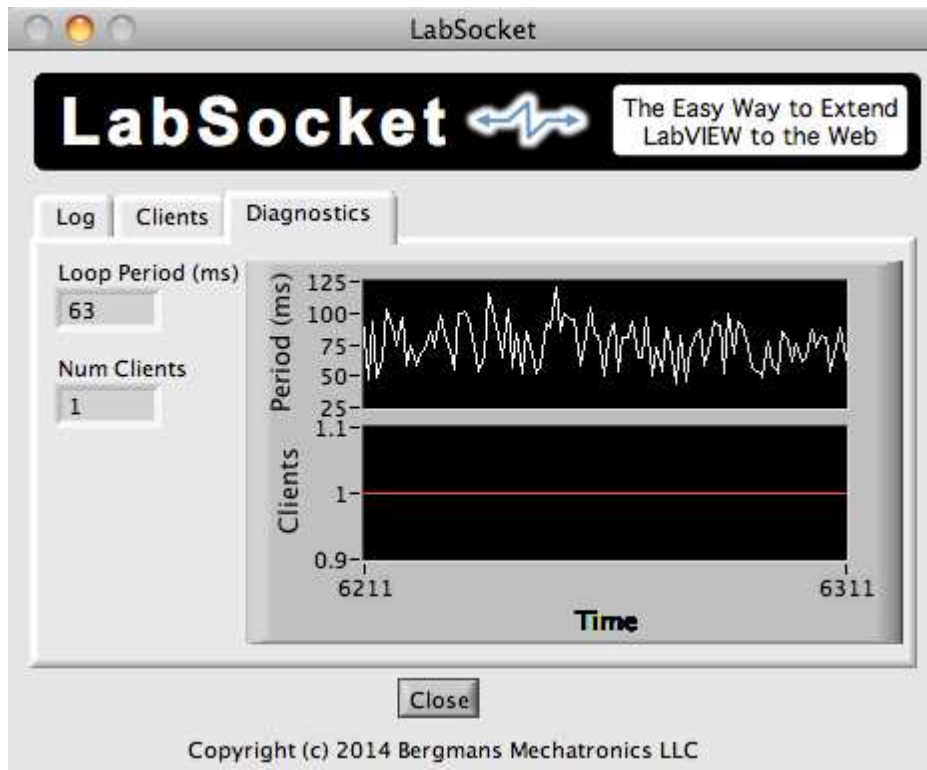


Figure 5.12 Ventana de Estado de LabSocket – Pagina de Diagnósticos

La página Diagnósticos también incluye una exhibición de la cantidad de clientes actualmente conectado a la Target VI

11 Haga clic en Cerrar para cerrar la ventana de estado y suspender la sincronización del Objetivo VI y navegador. Al cerrar la ventana de estado no se detendrá el Target VI y detener el Target VI no cierra la Ventana de estado.

5.6 Comunicación entre instancias de servidor VI y Target VI

Mecanismos de comunicación entre procesos de LabVIEW existentes, tales como las variables globales, globales funcionales o colas pueden ser utilizados para transmitir datos entre las instancias de destino VI y VI el servidor.

Un escenario en el que se podría utilizar esta capacidad es permitir a un administrador de prueba para monitorear el progreso de cada participante en el concurso. Una demostración de la utilización de una cola de LabVIEW para monitoreo prueba se muestra a continuación. Tenga en cuenta que en esta demostración, el Servidor de VI en sí se ha extendido a la web mediante el uso de LabSocket-Basic.

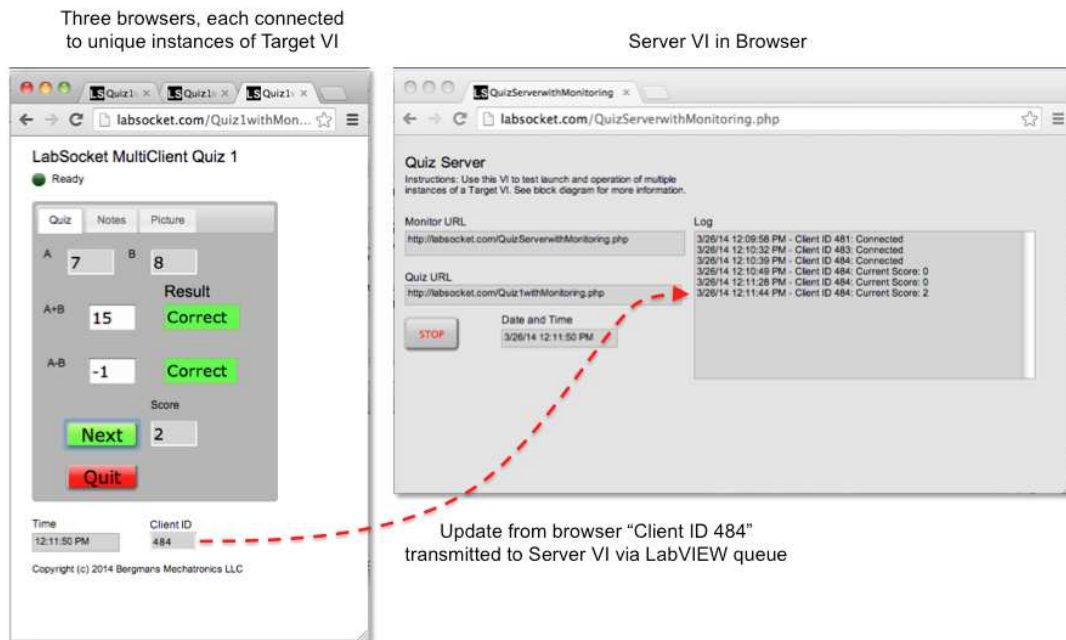


Fig. 5.13 Usando LabVIEW para supervisar las instancias Target VI en Servidor VI.

5.7 Aplicación del Labsocket

Extender LabVIEW para la Web mediante LabSocket

El Sistema LabSocket es un paquete adicional para LabVIEW que permite el acceso remoto a una aplicación de LabVIEW desde el escritorio o navegadores web para móviles. Para acceder a una aplicación de LabVIEW, o "objetivo del instrumento virtual (VI)", a través de Internet, el software

LabSocket funciona sobre la plataforma LabVIEW anfitrión en paralelo al Target VI, como se muestra en la siguiente figura. El Soporte LabSocket VIs operan discretamente y realizan dos funciones clave:

Al iniciar el sistema, el soporte VIs de realizar automáticamente un **Screenscrape** del panel frontal Objetivo VI. Esta función implica la generación de una página web con código HTML y JavaScript que representa el panel frontal Objetivo VI. Este código se envía entonces a un servidor HTTP en la plataforma LabSocket Server.

Después de realizar el screenscrape, el Soporte VIs entrar en **el modo de sincronización dinámica**. En este modo, las propiedades de los elementos en el panel frontal del objetivo VI se sincronizan continuamente con aquellos en el navegador.

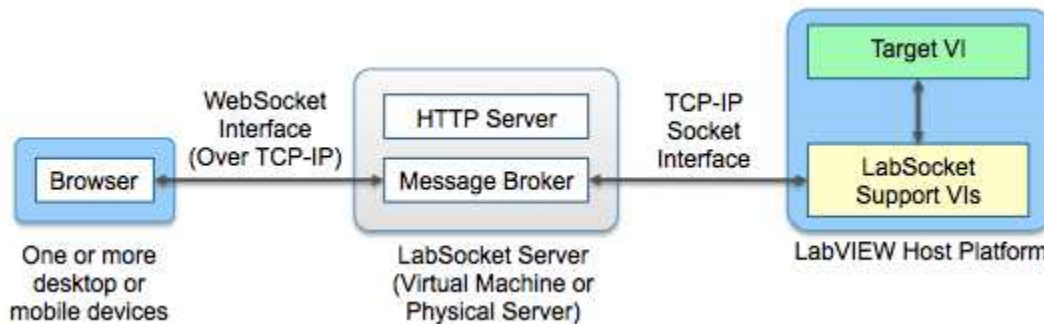


Fig. 5.14 Sistema LabSocket general

Los mensajes entre el navegador y soporte VI pasan a través de un Message Broker en la plataforma LabSocket Server. De especial interés es la interfaz WebSocket entre el navegador y Message Broker que permite la comunicación continua y bidireccional con el navegador. El uso de esta interfaz contribuye a el nombre del sistema: "LabSocket = LabVIEW + WebSocket".

LabSocket: Una herramienta de creación Web basado en LabVIEW

El propósito original del sistema LabSocket era simplemente permitir que los desarrolladores de LabVIEW para hacer las aplicaciones disponibles en la web. Una nueva versión del sistema denominado "**LabSocket-MultiClient**" existe ahora en la que cada cliente del navegador se conecta automáticamente a una instancia única del LabVIEW Objetivo VI. Con esta asignación uno a uno entre el navegador y la instancia de destino VI, la combinación de LabVIEW y LabSocket permite el desarrollo fácil y rápido de aplicaciones web dinámicas.

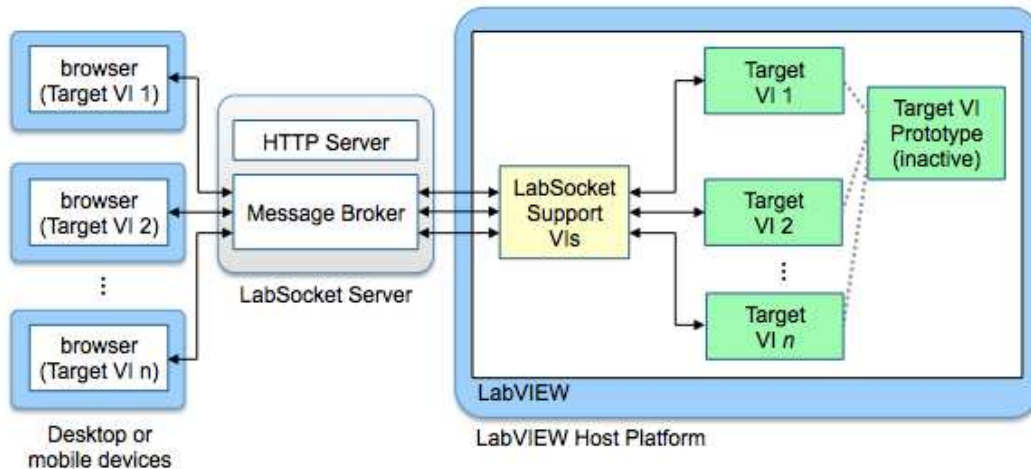


Fig. 5.15 Sistema LabSocket-MultiClient general

La siguiente figura ilustra el sistema LabSocket-multicliente en la acción como una herramienta de creación web. En esta figura, la aplicación de prueba de matemáticas simple descrito anteriormente se utiliza como el destino VI y tres navegadores se han apuntado a la dirección del VI. Tenga en cuenta que los navegadores están operando de forma independiente el uno del otro como lo demuestra el hecho de que los datos de preguntas y respuestas, y las puntuaciones de los usuarios, en cada navegador son diferentes. Cada navegador muestra el mismo valor de marca de tiempo desde que el concurso VI siempre muestra la hora actual en el panel frontal.

Claramente de diferentes estados de aplicación también son evidentes en esta figura - el navegador en el centro está a la espera de que el usuario introduzca los datos del cuestionario y pulse el botón "Enviar", mientras que los navegadores izquierdo y derecho están a la espera de que el usuario pulse el botón "Siguiente" para iniciar una nueva ronda de la prueba.

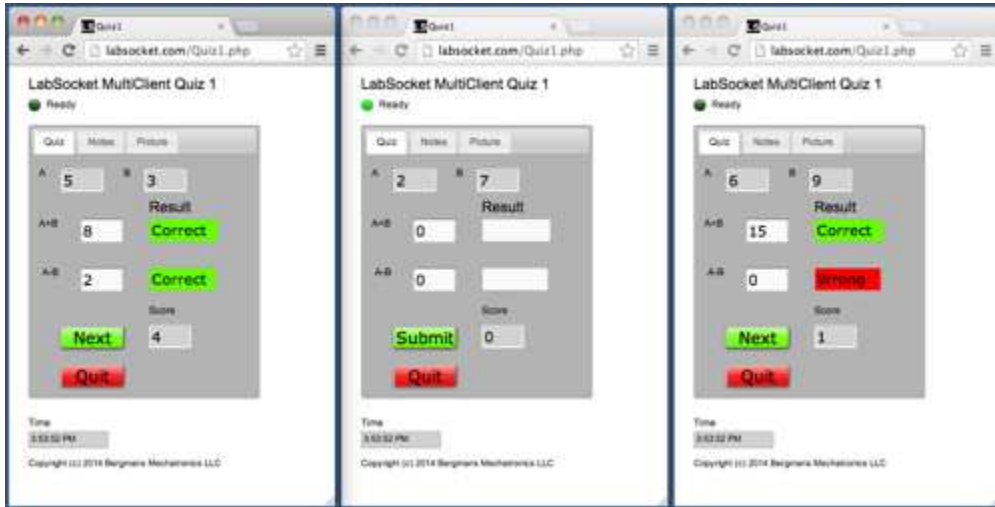


Fig. 5.16 Tres Navegadores Cada conectados simultáneamente a instancias únicas de concurso de aplicaciones

Conclusión

Logramos familiarizarnos del protocolo de comunicación DeviceNet, conociendo las ventajas que este nos presenta con respecto de los otros, dándonos un amplio conocimiento donde se expusieron las principales características y fundamentos de uno de los más utilizados en la industria.

Conocimos la labor principal de un sistema basado en computadores que nos permite supervisar y controlar a distancia un proceso de cualquier tipo, las diferencias de los sistemas de control distribuido.

Vimos que la interpretación de los procesos en una computadora facilita el manejo de los procesos, sin tener que correr el riesgo de estar físicamente abriendo o cerrando válvulas, o checando cuando realizar una acción, en este caso conocimos un poco de la interfaz LabVIEW donde este nos da la información necesaria al usuario.

Uno de los principales temas es el Labsocket desarrollado por LabVIEW en donde nos permite disponer de cualquier dato que el usuario desee por medio de la internet o los navegadores web para móviles y así el usuario pueda disponer, modificar, monitorizar, o realizar cambios desde donde se encuentre.

Bibliografía

- Vicente Guerrero, Ramón L. Yustle, Luis Martínez, Comunicaciones Industriales, Primera edición, septiembre 2009.
- <http://bibdigital.epn.edu.ec/bitstream/15000/10020/2/PARTE%202.pdf>
- www.smar.com/devicenet.asp
- Aquilino Rodríguez Penin, Sistemas SCADA, Tercera Edición, 2012.
- www.instrumentacionycontrol.net/cursos-libres/automatizacion/curso-supervision-procesos-por-computadora/item/271-los-buses-de-campo-directo-al-grano.html
- www.isa.uniovi.es/docencia/redes/Apuntes/tema9.pdf
- Aquilino Rodríguez Penin, Comunicaciones Industriales, Marcombo, 2008.
- <http://labsocket.com/technical%20details.html>
- Nuria Oliva Alonso, Redes de comunicaciones industriales, Editorial UNED, 2013.
- www.info-labview.org
- Ian Sommervill, Ingeniería del Software, Séptima edición, pag. 241_258, 2005.
- www.ni.com/labviewse



Centro Universitario, 11 de Agosto, 2014.

C. OMAR PABLO PÉREZ GUEVARA,
 Pasante (s) del área de Ingeniería en automatización,
PRESENTE .

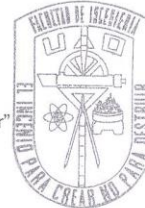
En respuesta a la solicitud que Usted (es) plantea (n) sobre la ASIGNACIÓN de ASESOR, necesario para la elaboración de la TESINA, y requisito para la acreditación del CURSO DE REDES INDUSTRIALES DE PLC´s, se le (s) designa al (la) DR. JUVENAL RODRÍGUEZ RESÉNDIZ, para desarrollar el tema: "CONFIGURACIÓN DE UNA RED DEVICENET CON HMI EN LABIEW MEDIANTE LABSOCKET", por lo que deberá (n) dirigirse al catedrático para ponerse de acuerdo y realizar el trabajo correspondiente tomando en cuenta los lineamientos que para su elaboración otorga el Centro de Educación Continua.

Por lo anterior me permito hacer de su conocimiento que la tesina deberá entregarse debidamente en el formato de disco compacto otorgado por la Mesa de Profesiones de la Dirección de Servicios Académicos; aprobada por su asesor y la coordinación del área a más tardar el día 09 de Mayo del 2014.

ATENTAMENTE

"El ingenio para crear, no para destruir"

M. en I. RUBÉN RAMÍREZ JIMÉNEZ,
 COORDINADOR



CENTRO DE
 EDUCACIÓN
 CONTINUA

FIRMA DE ACEPTACIÓN DEL ASESOR
 DR. JUVENAL RODRÍGUEZ RESÉNDIZ

Firma: 
 Fecha: 11 Agosto 2014
 De enterado



Cp. Archivo.
 Cpr*



Centro Universitario, Agosto 2014.

A QUIEN CORRESPONDA:

Hago constar que el (os) pasante (s) de la Licenciatura de Ingeniería en Automatización C. OMAR PABLO PÉREZ GUEVARA, aprobó (n) la TESINA del CURSO DE REDES INDUSTRIALES DE PLC´s impartido por el Centro de Educación Continua para lo cual presenta (n) el trabajo titulado: "CONFIGURACIÓN DE UNA RED DEVICENET CON HMI EN LABVIEW MEDIANTE LABSOCKET"

Aprobada con fines de Titulación.

DR. JUVENAL RODRÍGUEZ RESÉNDIZ
ASESOR DE TESINA