

**UNIVERSIDAD TÉCNICA DE AMBATO**

*FACULTAD DE INGENIERÍA EN SISTEMAS*

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO  
MONITOREADO POR EL SOFTWARE INTOUCH USANDO PLC”**

**Autores: Guilbert Paúl Basantes Hidalgo  
Elizabeth del Carmen Robalino Morán**

**Director: Ing. Julio Cuji**

**Asesor : Msc. Edison Alvarez**

**Tesis de Grado, previa a la Obtención del Título de  
Ingeniero en Sistemas**

**Ambato – Ecuador**

**Enero 2005**

# AGRADECIMIENTOS

A Dios por guiarnos en todos los momentos y por darnos la fuerza constante para seguir adelante.

A la Facultad de Ingeniería en Sistemas por habernos recibido en sus aulas durante nuestra carrera, formándonos no solo en el aspecto académico sino también en el aspecto moral.

A nuestro Director de Tesis Ing. Julio Cuji y al Msc. Edison Alvarez por su valiosa ayuda en el desarrollo de la Tesis.

A todas las personas que colaboraron de una u otra forma para el desarrollo y finalización de este trabajo.

# DEDICATORIA

Paúl

A Dios por darme la fuerza para seguir adelante.

A mi hijo Martín que ha sido la luz en los días de agonía

A mis padres por confiar en mi y darme siempre el apoyo necesario.

A mis hermanos a quienes quiero y respeto mucho.

A mi familia, tíos y primos por llenar mi vida y apoyarme en mis malos momentos.

Elizabeth

A mi Hijo Martín que es la personita que da alegría a mi vida

A mi familia por la ayuda que me brindaron de una u otra forma

# PROLOGO

El siguiente trabajo tiene por objetivo el Diseño e Implementación de un Sistema Domótico monitoreado por el Software Scada Intouch mediante PLC, para la automatización y control de todos los procesos en una vivienda y con la finalidad de realizar una herramienta de seguridad y confort para el hogar para evitar el robo simulando presencia en viviendas deshabitadas; en períodos cortos de fin de semana o en estancias más largas y que permita monitorizar los dispositivos de una vivienda desde un PC permitiendo informar en todo momento de los eventos que sucedan en la vivienda y poder actuar en consecuencia.

# CONTENIDO

## PRELIMINARES

CARÁTULA.....	i
AGRADECIMIENTO.....	ii
DEDICATORIA.....	iii
PROLOGO.....	iv
CONTENIDO.....	vi
INTRODUCCION.....	vi

## *CAPITULO I*

### *GENERALIDADES*

1.1. La Domótica.- Antecedentes e Importancia.....	2
1.2. Los Sistemas Domóticos.....	4
1.2.1. Descripción del Sistema Domótico.....	6
1.2.1.1. Tipo de Arquitectura.....	7
1.3. Tipos de Sistemas Domóticos.....	9
1.3.1. Sistemas mediante corrientes portadoras.....	9

1.3.2. Sistemas con sus propia Red de Datos.....	11
1.4. Partes de un Sistema Domótico.....	14
1.5. Principales Sensores y Actuadores utilizados en Domótica.....	18

## ***CAPITULO II***

### ***LOS AUTÓMATAS PROGRAMABLES Y EL SOFTWARE***

#### ***SCADA INTOUCH.***

2.1. LOS AUTÓMATAS PROGRAMABLES.....	22
2.1.1. Definición de autómatas programables.....	22
2.1.2. El autómata programable Simatic S7-200.....	24
2.1.2.1. Requisitos del Sistema.....	25
2.1.3. Montaje y Cableado del Simatic.....	27
2.1.3.1. Espacio necesario para montar una CPU 224.....	28
2.1.3.2. Montar una CPU 224 en un armario eléctrico.....	29
2.1.3.3. Montar una CPU 224 en un perfil soporte.....	30
2.1.4. Programación del Simatic.....	31
2.1.4.1. Crear una solución de Automát..con un Micro-PLC....	32
2.1.4.2. Estructurar el proceso o la instalación.....	33

2.1.4.3.	Especificar las unidades funcionales.....	34
2.1.4.4.	Diseñar los circuitos de seguridad cableados.....	34
2.1.4.5.	Definir las estaciones de operador.....	36
2.1.4.6.	Crear los planos de configuración del PLC.....	37
2.1.4.7.	Elaborar un lista de nombres simbólicos.....	38
2.1.4.8.	Programas S7-200.....	38
2.1.4.9.	Lenguajes de programación para las CPUs S7-200.....	40
2.1.4.9.1.	Elementos Básicos de KOP.....	41
2.1.4.9.2.	Operaciones de AWL.....	43
2.1.4.9.3.	Editor FUP (Diagrama de Funciones).....	44
2.1.4.10.	Elementos básicos para estructurar un programa.....	46
2.1.4.10.1.	Estructurar el programa.....	47
2.1.4.10.2.	Programa Ej. con sub y rutinas Inter.....	50
2.1.4.11.	El ciclo de la CPU.....	52
2.1.4.11.1.	Leer las entradas digitales.....	53
2.1.4.11.2.	Ejecutar el programa.....	55
2.1.4.11.3.	Procesar las peticiones de comunicación..	55
2.1.4.11.4.	Efectuar el autodiagnóstico de la CPU.....	56
2.1.4.11.5.	Escribir las salidas digitales.....	56
2.1.4.11.6.	Interrumpir el ciclo.....	57
2.1.4.11.7.	Imagen del proceso entradas y salidas.....	58
2.1.4.11.8.	Control directo de las entradas y salidas..	60

2.1.4.12.	Ajustar el modo de operación de la CPU.....	60
2.1.4.12.1.	Cambiar modo de operación con selector.....	61
2.1.4.12.2.	Cambiar modo de operación con STEP7.....	62
2.1.4.12.3.	Cambiar modo opera. desde programa.....	63
2.1.4.13.	Determinar una contraseña para la CPU.....	63
2.1.4.13.1.	Restringir el acceso a la CPU.....	64
2.1.4.13.2.	Configurar la contraseña para la CPU.....	65
2.1.4.13.3.	Qué hacer si se olvida la contraseña.....	67
2.1.5.	Funciones Básicas.....	69
2.2.	EL SOFTWARE SCADA INTOUCH.....	70
2.2.1.	Introducción a Scada Intouch.....	70
2.2.1.1.	Creación de una Aplicación.....	71
2.2.2.	Entorno de Desarrollo.....	73
2.2.2.1.	Elementos de WindowMarker.....	73
2.2.2.1.1.	Menús de WindowMarker.....	73
2.2.2.2.	Usando WindowMarker.....	75
2.2.2.2.1.	Tipos de Ventanas.....	75
2.2.2.2.2.	Los Elementos Wizard.....	77
2.2.3.	El Diccionario de Tagnames.....	78
2.2.3.1.	Definición de Tagname.....	78
2.2.3.2.	Definición de los Tagnames.....	79
2.2.3.3.	Definición de las Características.....	81



2.2.3.4.	Campos a Rellenar del Tagname.....	82
2.2.4.	Los Animation Links.....	84
2.2.4.1.	¿Qué son las Animation Links? .....	85
2.2.4.2.	Animando Objetos.....	85
2.2.4.3.	Edición de Links y de Tags.....	85
2.2.4.3.1.	Sustituir Texto.....	85
2.2.4.3.2.	Sustituir Tagnames.....	86
2.2.4.3.3.	Borrar Tagnames.....	86
2.2.5.	Intouch QUICKSCRIPTS.....	87
2.2.5.1.	Tipos de Scripts.....	87
2.2.5.2.	Funciones Scripts.....	88
2.2.5.3.	Uso de Variables Internas.....	88
2.2.6.	Alarmas y Eventos.....	89
2.2.6.1.	Tipos de Alarmas.....	90
2.2.6.2.	Prioridades de las Alarmas.....	91
2.2.6.3.	Grupos de Alarmas.....	91
2.2.6.4.	Cómo crear grupos de Alarmas.....	92
2.2.6.5.	Definición de Condición de Alarma en Tagname.....	92
2.2.6.6.	Configuración de un Objeto de Alarmas.....	94
2.2.6.7.	Configuración de Alarmas/Eventos.....	96
2.2.6.8.	Campos de las Alarmas.....	97
2.2.7.	Curvas Históricas y Reales.....	99

2.2.7.1.	Curvas en Tiempo Real.....	100
2.2.7.1.1.	Creación de una Curva Tiempo Real.....	100
2.2.7.1.2.	Configuración de Curva Tiempo Real....	100
2.2.7.2.	Curvas Históricas.....	101
2.2.7.2.1.	Configurar Historical Logging.....	102
2.2.7.2.2.	Utilización de Wizard de Curva Hist.....	102
2.2.8.	Comunicaciones I/O SERVERS.....	103

### ***CAPITULO III***

## ***DISEÑO Y DESARROLLO DEL PROTOTIPO Y DEL SISTEMA***

3.1.	DISEÑO Y DESARROLLO DEL PROTOTIPO.....	106
3.1.1.	Diseño del Prototipo.....	106
3.1.1.1.	Procesos a automatizar.....	106
3.1.1.2.	Entradas y Salidas.....	109
3.1.1.3.	Diseño de los Circuitos.....	111
3.1.2.	Desarrollo del Prototipo.....	112
3.1.2.1.	Construcción del Prototipo.....	112
3.1.2.1.1.	Conexiones Eléctricas del Tablero de Control.....	113

3.1.2.1.2.	Construcción de la Tarjeta Electrónica.....	113
3.1.2.2.	Programación del PLC Simatic S7-200.....	121
3.1.2.2.1.	Desarrollo del Programa en el PLC .....	124
3.1.2.3.	Conexión de los Dispositivos con PLC Simatic S7-200.....	132
3.1.2.3.1.	Comunicación PLC-INTOUCH .....	133
3.1.2.3.2.	Configuración del DRIVER de Comunicación....	135
3.1.2.3.3.	Configuración del Puerto de Comunicaciones.....	136
3.1.2.3.3.1	Configuración del Topic.....	138
3.2.	DISEÑO Y DESARROLLO DEL SISTEMA.....	140
3.2.1.	Visión General del Sistema.....	140
3.2.2.	Objetivos del Sistema.....	141
3.2.3.	Alcance del Sistema.....	142
3.2.4.	Requerimientos hardware y software.....	143
3.2.5.	Descripción del Sistema.....	144
3.2.6.	Análisis del Sistema.....	145
3.2.7.	Diseño de la aplicación.....	148
3.2.7.1.	Presentación.....	148
3.2.7.2.	Botón de Registro Usuario.....	150
3.2.7.3.	Menú Principal del Sistema.....	152
3.2.7.3.1.	Estado del Sistema.....	152
3.2.7.3.2.	Módulo de Control.....	154
3.2.7.3.3.	Módulo Control de Tiempo.....	155

3.2.7.3.4.	Módulo Alarmas y Eventos.....	156
3.2.7.3.5.	Módulo Gráficas.....	162
3.2.7.3.6.	Ventana Históricos.....	178
3.2.7.3.7.	Módulo Simulación.....	179
3.2.7.3.8.	Módulo Reportes.....	186
3.2.7.3.9.	Módulo Ayuda.....	189
3.2.8.	Herramientas a utilizar.....	189
3.2.9.	Implementación. ....	190
3.3.	PRUEBAS Y RESULTADOS DEL PROTOTIPO CONSTRUIDO. ....	191
3.4.	PRUEBAS Y RESULTADOS DEL PROTOTIPO COMPLETO.....	191

## ***CAPITULO IV***

### ***CONCLUSIONES Y RECOMENDACIONES***

4.1.	CONCLUSIONES.....	194
4.2.	RECOMENDACIONES.....	196

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

Manual de Usuario

Varios

# ***CAPITULO I***

## ***GENERALIDADES***

## La Domótica .- Antecedentes e Importancia

A mediados de la década de 1980 a 1990 surge el concepto de Edificio Inteligente y con ello atrajo la atención de constructores de edificios y del mercado inmobiliario. Esta nueva propuesta integró todos los aspectos de comunicación dentro del edificio, seguridad, control del sistema de temperatura del edificio y la administración de la energía.

En la actualidad, al estudio de edificios inteligentes se le llama Inmótica y se define como *el estudio de la estructura de un edificio que facilita a usuarios y administradores, herramientas y servicios integrados a la administración y la comunicación*. El diseño de estas estructuras cubre las necesidades reales de los usuarios y administradores, haciendo uso de todos los posibles adelantos tecnológicos, incluyendo además, factores humanos, ergonómicos y ambientales.

**Cuando se popularizó esta estructura;  
principalmente en Europa, Estados Unidos y  
Japón; las personas constructoras de estos  
edificios se dieron cuenta que podían realizar**

**lo mismo en las casas donde ellos habitaban;  
fue así como surgieron las casas inteligentes y  
a su estudio se le llamó Domótica.**

El término Domótica viene del latín “domus” que significa casa y de la palabra “automática”, por lo tanto la domótica se refiere a una casa automática o como se le ha llamado más comúnmente una casa inteligente. En inglés a la domótica se le conoce más como “home networking” o “smart home”. Una casa inteligente es aquella cuyos elementos o dispositivos están integrados y automatizados y que a través de otro dispositivo remoto o inclusive interno se pueden modificar sus estados o los mismos dispositivos están diseñados para realizar ciertas acciones cuando han detectado cambios en su propio estado.

Una definición más técnica del concepto sería: "conjunto de servicios de la vivienda garantizado por sistemas que realizan varias funciones, los cuales pueden estar conectados entre sí y a redes interiores y exteriores de comunicación. Gracias a ello se obtiene un notable ahorro de energía, una eficaz gestión técnica de la vivienda, una buena comunicación con el exterior y un alto nivel de seguridad".

*La Domótica es importante porque sus beneficios son múltiples, y en general cada día surgen nuevos.*

Por ello se cree conveniente agruparlos en los siguientes apartados:

- El ahorro energético gracias a una gestión tarifaria e "inteligente" de los sistemas y consumos.
- La potenciación y enriquecimiento de la propia red de comunicaciones.
- La más contundente seguridad personal y patrimonial.
- La gestión remota (Vigilancia: Vía teléfono, radio, Internet, etc.) de instalaciones y equipos domésticos.

Como consecuencia de todo lo expuesto anteriormente se consigue un nivel de confort muy superior y la calidad de vida aumenta considerablemente.

Los Sistemas Domóticos.



Los Sistemas Domóticos son sistemas de control de las instalaciones de una vivienda. En un Sistema Domótico se integran todas las funciones de gestión, control, regulación y comprobación de todos los elementos asociados a dicha instalación. Los sistemas domóticos permiten disponer de una vivienda dotada de un Sistema de Control Doméstico capaz de realizar cualquier función que se le pida. Comunicarse, proteger su casa, controlar a distancia sus aparatos domésticos, es decir, vivir mejor

Los Sistemas Domóticos se encargan de gestionar principalmente los siguientes cuatro aspectos del hogar:

*Energía eléctrica:* En este campo, la domótica se encarga de gestionar el consumo de energía, mediante temporizadores, relojes programadores, termostatos, etc. También se aprovecha de la tarifa nocturna, mediante acumuladores de carga.

*Confort:* La domótica nos proporciona una serie de comodidades, como pueden ser el control automático de los servicios de: Calefacción, Agua caliente, Refrigeración, Iluminación y la gestión de elementos como accesos, persianas, toldos, ventanas, riego automático, etc.

*Seguridad:* La seguridad que nos proporciona un sistema domótico es más amplia que la que nos puede proporcionar cualquier otro sistema, pues integra tres campos de la seguridad que normalmente están controlados por sistemas distintos:

*Seguridad de los bienes:* Gestión del control de acceso y control de presencia, así como la simulación de presencia. Alarmas ante intrusiones.

*Seguridad de las personas:* Especialmente, para las personas mayores y los enfermos. Mediante el nodo telefónico, se puede tener acceso (mediante un pulsador radiofrecuencia que se lleve encima, por ejemplo) a los servicios de ambulancias, policía, etc.

*Incidentes y averías:* Mediante sensores, se pueden detectar los incendios y las fugas de gas y agua, y, mediante el nodo telefónico, desviar la alarma hacia los bomberos, por ejemplo. También se pueden detectar averías en los accesos, en los ascensores, etc.

*Comunicaciones:* Este aspecto es imprescindible para acceder a multitud de servicios ofrecidos por los operadores de telecomunicaciones. La domótica tiene una característica fundamental, que es la integración de sistemas, por eso hay nodos (pasarela residencial) que interconectan la red domótica con diferentes dispositivos, como Internet, la red telefónica, etc.

## 1.2.1. Descripción del Sistema Domótico

Para poder clasificar técnicamente un sistema de automatización de viviendas, es necesario tener claros una serie de conceptos técnicos, como son: tipo de arquitectura, medio de transmisión, velocidad de transmisión y protocolo de comunicaciones.

### 1.2.1.1. Tipo de Arquitectura

La arquitectura de un sistema domótico, como la de cualquier sistema de control, especifica el modo en que los diferentes elementos de control del sistema se van a ubicar. Existen dos arquitecturas básicas: la arquitectura centralizada y la distribuida.

Arquitectura centralizada: Es aquella en la que los elementos a controlar y supervisar (sensores, luces, válvulas, etc.) han de cablearse hasta el sistema de control de la vivienda (PC o similar). El sistema de control es el corazón de la vivienda, en cuya falta todo deja de funcionar, y su instalación no es

compatible con la instalación eléctrica convencional en cuanto que en la fase de construcción hay que elegir esta topología de cableado.

Arquitectura distribuida: Es aquella en la que el elemento de control se sitúa próximo al elemento controlar.

Hay sistemas que son de arquitectura distribuida en cuanto a la capacidad de proceso, pero no lo son en cuanto a la ubicación física de los diferentes elementos de control y viceversa, sistemas que son de arquitectura distribuida en cuanto a su capacidad para ubicar elementos de control físicamente distribuidos, pero no en cuanto a los procesos de control, que son ejecutados en uno o varios procesadores físicamente centralizados.

En los sistemas de arquitectura distribuida que utilizan como medio de transmisión el cable, existe un concepto a tener en cuenta que es la topología de la red de comunicaciones. La topología de la red se define como la distribución física de los elementos de control respecto al medio de comunicación (cable).

Cada elemento del sistema tiene su propia capacidad de proceso y puede ser ubicado en cualquier parte de la vivienda. Esta característica proporciona al instalador domótico una libertad de diseño que le posibilita adaptarse a las características físicas de cada vivienda en particular.

#### Tipos de Sistemas Domóticos.

Los Sistemas Domóticos se pueden clasificar atendiendo a diversos criterios. Una clasificación muy intuitiva es la basada en el tipo de soporte utilizado para transmitir los datos de comunicación entre dispositivos. Atendiendo a este criterio, actualmente existen claramente dos tipos de sistemas domóticos:

- Sistemas basados en corrientes portadoras utilizando la red de baja tensión ya instalada en los edificios.
  
- Sistemas que utilizan una red de cableado especial para la interconexión de los dispositivos y su comunicación.

### 1.3.1. Sistemas mediante corrientes portadoras.

La característica más importante de estos sistemas es la utilización de la red de baja tensión previamente instalada en el edificio (red eléctrica convencional). De este modo el sistema domótico realiza la gestión técnica de la instalación (iluminación, calefacción, riego, etc.) utilizando como soporte de la información las líneas eléctricas ya existentes. El principio que rige este sistema (puesto que la energía eléctrica se transporta por una corriente sinusoidal de baja frecuencia 50 Hz o 60 Hz) es la utilización de emisores que inyectan en cualquier punto del circuito una señal de baja potencia y de frecuencia en torno a los 120 -140 kHz que se superpondrá a la frecuencia de red. Los receptores dispuestos por la red reciben la señal, la filtran y descifran el mensaje enviado por el emisor para actuar en consecuencia.

*Estos sistemas tienen la gran ventaja de no necesitar un cableado adicional para la instalación domótica, y que su aplicación en edificios ya construidos es inmediata. La desventaja de este tipo de sistemas es la complejidad de sus dispositivos, puesto que son necesarios moduladores/demoduladores en todos los elementos del*

*sistema ya que la información va superpuesta a la red eléctrica. Otro aspecto negativo de este sistema, pese a incorporar filtros para recuperar claramente los datos enviados, es la utilización de la red eléctrica como soporte de transmisión puesto que introduce numerosas interferencias que finalmente pueden influir en el correcto funcionamiento del sistema. Dentro de este sector del mercado existen principalmente tres sistemas comerciales: X-10 de Home Systems, CAD de Legrand y Global Home System de Landis & Gyr.*

#### 1.3.2. Sistemas con su propia red de datos

Dentro de los diferentes sistemas domóticos que necesitan su propia red de cableado para la comunicación y transmisión de datos entre dispositivos, se puede realizar a su vez otra subdivisión atendiendo a la topología de dicha red de datos o de interconexión entre dispositivos: Sistemas centralizados y Sistemas distribuidos.

##### ➤ Sistemas centralizados

En este tipo de sistemas existe un cerebro responsable de la instalación: La unidad central, dicho dispositivo es el encargado de procesar la información recibida desde los diferentes sensores y transmisores, y enviar, en consecuencia, órdenes a los actuadores correspondientes.

La topología de la red de transmisión utilizada en sistemas centralizados es de tipo estrella cuyo nodo principal es la unidad central . No existe comunicación directa entre sensores, transmisores y actuadores. Cuando un elemento sensor transmite una señal a la unidad central, ésta en función de la programación establecida por el usuario, transmitirá a su vez una serie de órdenes a los actuadores. La ventaja fundamental que tienen este tipo de sistemas es su bajo coste, ya que ningún elemento necesita módulos especiales de direccionamiento, ni interfaces para distintos buses.

Además, su instalación es más sencilla y es posible utilizar una gran variedad de elementos convencionales, puesto que los requisitos que se les exigen son mínimos.

El principal inconveniente es su limitada flexibilidad, puesto que las posibles reconfiguraciones serán muy costosas, teniendo en cuenta que



el término “reconfiguración” hace referencia a la posibilidad de añadir o quitar elementos del sistema domótico y no a las posibles variaciones del software realizadas por el usuario.

Además, al ser totalmente imprescindible el módulo de control para el correcto funcionamiento del sistema, si éste se avería, el sistema deja de funcionar completamente; esto reduce la robustez de la instalación. Otro inconveniente añadido es la necesidad de más metros de cableado que en el caso de sistemas distribuidos (basados en bus), con lo que su uso está bastante limitado en grandes instalaciones. El sistema más instalado en el ámbito nacional de este tipo de sistemas domóticos centralizados es el SimonVIS de IHC (Innovation House Control).

#### ➤ Sistemas distribuidos

En este tipo de sistemas, cada elemento posee la capacidad de tratar la información que recibe y actuar en consecuencia de forma autónoma, por tanto no es necesaria la presencia de una unidad central. La configuración de un sistema domótico distribuido tiene una topología de cableado tipo bus .

Entre las ventajas cabe destacar la facilidad de reconfiguración del sistema, lo que incide directamente en el grado de flexibilidad, y sobre todo en el ahorro de cableado de la instalación.

Al tratarse de sistemas más caros también son sistemas más potentes que permiten implementar una gran cantidad de aplicaciones y servicios al usuario.

El inconveniente más serio es el coste de los elementos del sistema. Esto es debido a la necesidad de incluir los protocolos CSMA/CD de comunicación y las técnicas de direccionamiento en cada uno de los elementos, así como la capacidad de procesar información (inclusión de un microcontrolador); lo que implica una necesidad de compatibilidad entre equipos que a su vez repercute en una disminución importante de productos ofertados en el mercado.

Además, como se trata de sistemas más complejos, se plantea una fuerte restricción en la utilización de elementos convencionales, puesto que los requisitos exigidos para su compatibilidad son elevados. Por otra parte, en este tipo de sistemas, el software de programación es un software complicado de utilizar que requiere la participación de un experto para la programación de los diferentes dispositivos. Los sistemas más importantes que utilizan esta configuración son, por ejemplo, el bus EIB, Batibus, EHS, LonWorks, etc.

## Partes de un Sistema Domótico.

Al iniciar con una investigación o desarrollo en el campo de la domótica, es imprescindible tener una correcta idea del estado del arte en la tecnología que hoy

se está utilizando, de manera de enmarcar los desarrollos futuros dentro de las tendencias y normas que ya han sido adoptadas por el mercado.

La figura 1.1 propone un modelo para describir los distintos componentes de un sistema domótico. Esta representación nos permitirá manejar la complejidad, analizando cada componente por separado. Evitaremos así el estudio del sistema como un objeto monolítico.



### *Figura 1.1 Componentes de un sistema Domótico*

Los componentes de este modelo son:

**Tecnologías de Red:** a diferencia de lo que solemos encontrar en los ambientes comerciales, las tecnologías de comunicación que pueden ser utilizadas en los hábitats domotizados, son mucho más heterogéneas. La red en cuestión, debe ser capaz de transportar conversaciones telefónicas, señales de TV y de cámaras de video del sistema de seguridad, música en MP3 u otro formato, contenidos de DVDs y señales de control de electrodomésticos. Así mismo, las compañías proveedoras de servicios, como las de electricidad, podrían necesitar ingresar desde el exterior, para controlar el flujo de energía y leer los medidores en forma remota.

**Un Gateway:** denominado también “home gateway” o “residential gateway” que comunica a los dispositivos domóticos (sensores, cámaras de video, electrodomésticos, etc.) a Internet, permite que los usuarios accedan y controlen funciones de sus casas también desde Internet y actúa, a su vez, como un bridge/router interno permitiendo que dispositivos con distintas tecnologías de comunicación (Ethernet, BlueTooth, Wi-Fi, X-10,etc) interactúen entre sí.

**Dispositivos domóticos, sensores, actuadores:** son los puntos terminales del sistema domótico. Los constituyen los típicos electrodomésticos (heladeras, cocinas, hornos a microondas, etc.), sistemas relacionados con el confort y la seguridad (sensores de temperatura, humedad, luminosidad, interpretadores de voz, aire acondicionado, equipos de audio y video, de vigilancia, etc.), y otros elementos capaces de ser controlados, y automatizado su funcionamiento (ascensores, puertas, ventanas, portero eléctrico, etc.)

**Sistema de Control Centralizado:** este elemento actúa como centro de monitoreo y control de toda la instalación. Desde él es posible obtener el estado de cada uno de los sensores, actuadores y electrodomésticos para modificar o reprogramar su comportamiento.

Una de sus posibles implementaciones podría ser un computador (idealmente una tablet PC) corriendo una aplicación visual que muestre los distintos ambientes y elementos de la casa en una interfase 3D.

**Sistema de Inteligencia:** es el punto más alto del sistema domótico. Está constituido por el conjunto de aplicaciones que, basadas en conceptos de inteligencia artificial (sistemas expertos, redes neuronales, lógica difusa, algoritmos genéticos, computación evolutiva, etc.), toman decisiones de alto nivel

para lograr verdaderos hábitats inteligentes. Por ejemplo, basado en la información brindada por sensores de voz, temperatura y antropométricos, un sistema de este tipo podría inferir qué personas están en una determinada sala de la casa, y buscar entonces una solución de compromiso en la temperatura ambiente o en el sistema de música funcional, que fuera agradable para la mayoría.

#### Principales Sensores y Actuadores utilizados en Domótica.

Son muchos los elementos que componen los distintos sistemas de automatización de viviendas y edificios, desde una central de gestión para sistemas centralizados hasta un mando automático a distancia. Dentro de esta multiplicidad de elementos, empezamos con la definición de dos elementos muy característicos: *los sensores y los actuadores.*

***Los sensores son los elementos que utiliza el sistema para conocer el estado de ciertos parámetros (la temperatura ambiente, la existencia de un escape de agua, etc.).***

***Entre los más comúnmente utilizados se distinguen los siguientes:***

- *Termostato de ambiente, destinado a medir la temperatura de la estancia y permitir la modificación de parámetros de consigna por parte del usuario.*
  
- *Sensor de temperatura interior, destinado a medir únicamente la temperatura de la estancia.*
  
- *Sensor de temperatura exterior, destinado a optimizar el funcionamiento de la calefacción a través de una óptima regulación de su carga y/o funcionamiento.*
  
- *Sondas de temperatura para gestión de calefacción, necesarias para controlar de forma correcta distintos tipos de calefacción eléctrica (por ejemplo, sondas limitadoras para suelo radiante).*
  
- *Sonda de humedad, destinada a detectar posibles escapes de agua en cocinas, baños, etc.*

- *Detector de fugas de gas, para la detección de posibles fugas de gas en cocina, etc.*
  
- *Sensor de presencia, para detección de intrusiones no deseadas en la vivienda.*
  
- *Receptor de infrarrojos.*

*Por otra parte, tenemos los actuadores, que son elementos que utiliza el sistema para modificar el estado de ciertos equipos e instalaciones.*

*Entre los más comúnmente utilizados se distinguen los siguientes:*

- *Contactores (o relés de actuación) de carril DIN.*
  
- *Contactores para base de enchufe.*
  
- *Electroválvulas de corte de suministro (gas y agua).*
  
- *Válvulas para la zonificación de la calefacción por agua caliente.*



➤ *Sirenas o elementos zumbadores, para el aviso de alarmas en curso.*

*Debemos indicar que el número de sensores y actuadores varía en el tiempo, dado que muchas empresas agrupan unos u otros en un solo aparato abaratando costes, por lo que esta información siempre estará sujeta a posibles modificaciones a lo largo del tiempo. Sin embargo, los elementos básicos no varían.*

# INTRODUCCIÓN

Se ha desarrollado el presente trabajo para demostrar la utilización de los autómatas programables, muy utilizados en los procesos industriales, pero esta vez adaptándolos en la automatización de una casa que acoplada a un PLC y manejada por medio de un software para su manipulación vienen a formar un Sistema Domótico que permite el control en forma automática y manual de varias tareas como son: Control Automático de Temperatura, Nivel del Tanque de Agua, de Riego de Jardín, de Accesos, de Presencia Exterior, Interior, Alarma de Incendio, así como el Control Manual y Horario de Iluminación, Control de Persianas, etc.

El Sistema lo desarrollamos mediante el software Intouch 7.1., el cual presenta facilidades de comunicación con el PLC SIMATIC S7-200 CPU 224 (DC/DC/DC), mediante su DRIVER el S7200PPI, el que utiliza el Protocolo DDE para intercambio de datos.

El ingreso de datos se lo puede realizar desde un panel de operador ubicado en la parte del módulo o desde el computador por medio de una interfaz gráfica desarrollada con el software INTOUCH por medio de la cual se realiza la supervisión y adquisición de datos.

Todas las acciones de control son realizadas por medio de un PLC el que procesa las señales de entrada y determina las acciones a tomar para lograr las condiciones de funcionamiento deseado.

## ***CAPITULO II***

***LOS AUTÓMATAS PROGRAMABLES Y***

***EL SOFTWARE SCADA INTOUCH***

## 2.1 Los Automatas Programables

### ***2.1.1. Definición de Automatas programables.***

Los autómatas programables aparecieron en los Estados Unidos de América en los años 1969–1970, y más particularmente en el sector de la industria del automóvil; fueron empleados en Europa alrededor de dos años más tarde. Su fecha de creación coincide, pues con el comienzo de la era del microprocesador y con la generación de la lógica cableada modular.

El autómata es la primera máquina con lenguaje, es decir, un calculador lógico cuyo juego de instrucciones se orienta hacia los sistemas de evolución secuencial. Hay que apreciar que, cada vez más, la universalidad de los ordenadores tiende a desaparecer, el futuro parece abrirse hacia esta nueva clase de dispositivos: máquina para proceso de señales, para la gestión de bases de datos...etc.

*El autómata programable es, pues en este sentido un precursor y constituye para los automatistas un esbozo de la máquina ideal.*

La creciente difusión de aplicaciones de la electrónica, la fantástica disminución del precio de los componentes, el nacimiento y el desarrollo

de los microprocesadores y, sobretodo, la miniaturización de los circuitos de memoria permiten presagiar una introducción de los autómatas programables, cuyo precio es atractivo incluso para equipos de prestaciones modestas, en una inmensa gama de nuevos campos de aplicación.

El autómata programable satisface las exigencias tanto de procesos continuos como discontinuos. Regula presiones, temperaturas, niveles y caudales así como todas las funciones asociadas de temporización, cadencia, conteo y lógica.

También incluye una tarjeta de comunicación adicional, el autómata se transforma en un poderoso satélite dentro de una red de control distribuida.

El autómata programable es un aparato electrónico programable por un usuario programador y destinado a gobernar, dentro de un entorno industrial, maquinas o procesos lógicos secuenciales.

### ***2.1.2. El Autómata Programable Simatic S7 –200***

Es un aparato autónomo compacto que comprende una unidad central de procesamiento (CPU), la fuente de alimentación, así como entradas y salidas digitales.

La CPU ejecuta el programa y almacena los datos para la tarea de automatización o el proceso.

Las entradas y salidas controlan el sistema de automatización. Las entradas vigilan las señales de los aparatos de campo (p.ej. sensores e interruptores) y las salidas vigilan las bombas, los motores u otros dispositivos del proceso.

Los diodos luminosos indican el modo de operación de la CPU (RUN) o si se ha detectado un fallo del sistema (SF).

***La gama S7-200 comprende diversos sistemas de automatización pequeños (Micro-PLCs) que se pueden utilizar para numerosas tareas. La figura 2.1 muestra un Micro-PLC S7-200. Gracias a su diseño compacto, su capacidad de ampliación, su bajo costo***

*y su amplio juego de operaciones, los Micro-PLCs S7-200 se adecúan para numerosas aplicaciones pequeñas de control. Además, los diversos tamaños y fuentes de alimentación de las CPUs ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.*

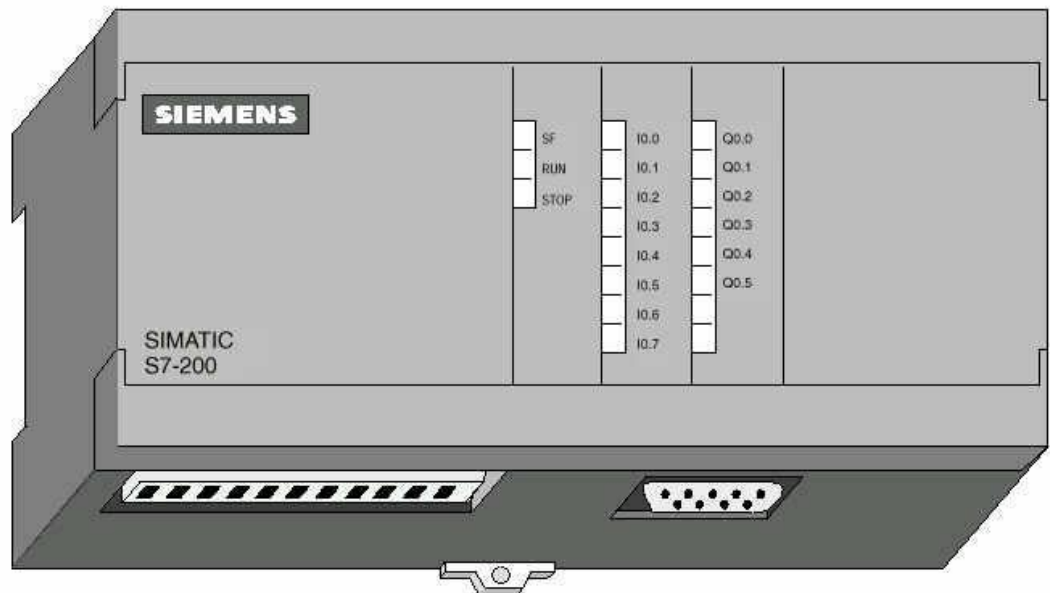


Figura 2.1. Micro-PLC S7-



### **2.1.2.1. Requisitos del sistema**

Como muestra la figura 2.2, el software de programación STEP 7-Micro/WIN se puede utilizar con una estación de desarrollo de programas (PDS 210) para crear y comprobar los programas. El programa definitivo se carga en un cartucho de memoria que se inserta luego en la CPU 224.

Con objeto de crear programas para la CPU 224 se requieren los siguientes equipos:

- Ordenador personal (PC) con el software de programación STEP 7-Micro/WIN.
- Estación de desarrollo de programas (PDS 200).
- Cable de comunicación PC/PPI.
- Cartucho de memoria para transferir el programa a la CPU 224.

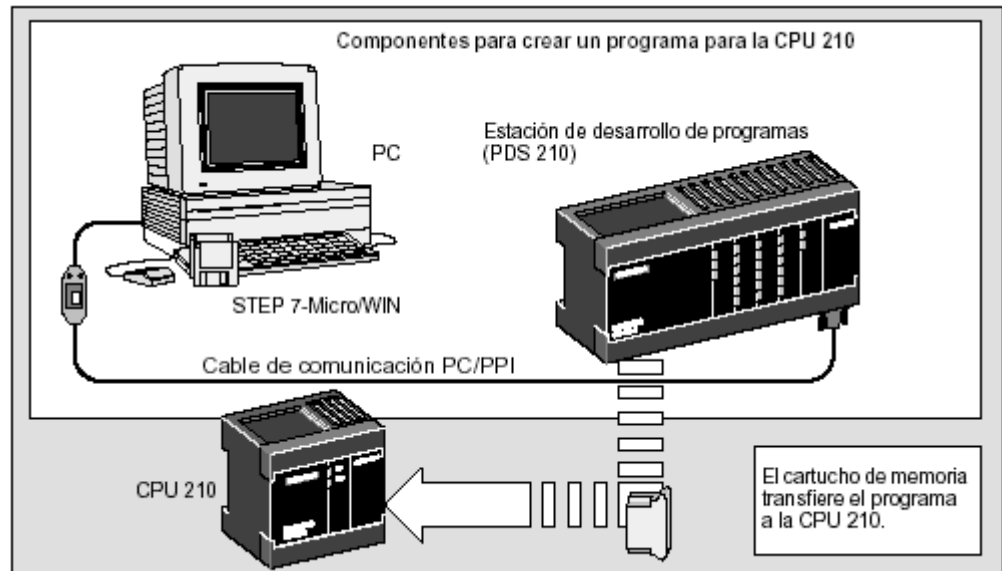
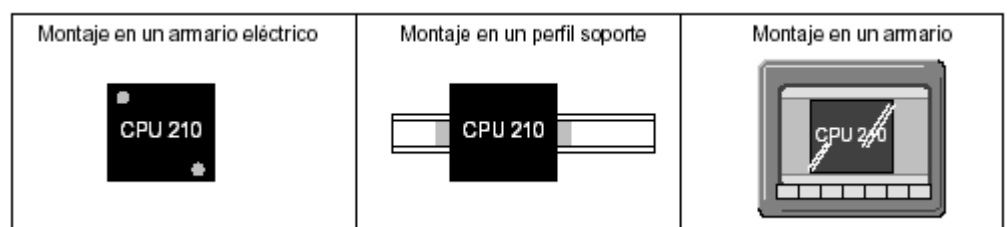


Figura 2.2. Componentes para crear un programa para la CPU 224

### 2.1.3. Montaje y Cableado del Simatic

Como muestra la figura 2.3., la CPU 224 se puede disponer en un armario eléctrico o en un perfil soporte. La CPU 224 se puede montar de forma horizontal o vertical.

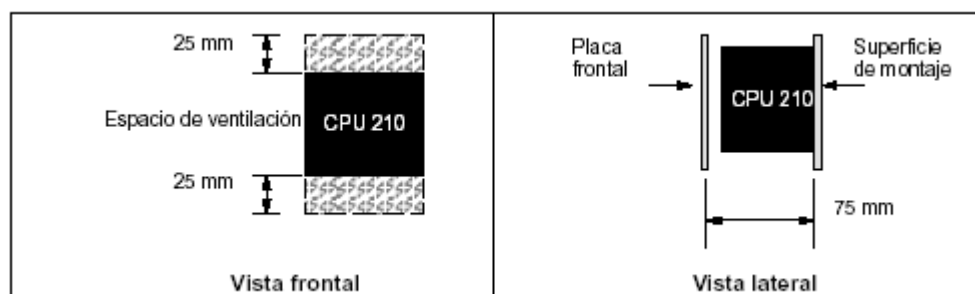


*Figura 2.3. Montaje de la CPU*

### **2.1.3.1. Espacio necesario para montar una CPU 224**

Al configurar la disposición de los módulos en el bastidor se deberán respetar las siguientes reglas:

- Para la CPU 224 se ha previsto la ventilación por convección natural. Por lo tanto, se deberá dejar un margen mínimo de 25 mm por encima y por debajo de las unidades para garantizar su ventilación (Ver figura 2.4). El funcionamiento continuo a una temperatura ambiente máxima y con una carga muy elevada reduce la vida útil de cualquier dispositivo electrónico.
- Si la CPU 224 se instala en un armario eléctrico, el espesor mínimo de éste último deberá ser de 75 mm.



*Figura 2.4. Espacio Necesario para montar la CPU*

- Al configurar la disposición de los aparatos, prevea suficiente espacio para el cableado de las entradas y salidas.

#### **2.1.3.2. Montar una CPU 224 en un armario eléctrico**

Para montar una CPU 224 en un armario eléctrico:

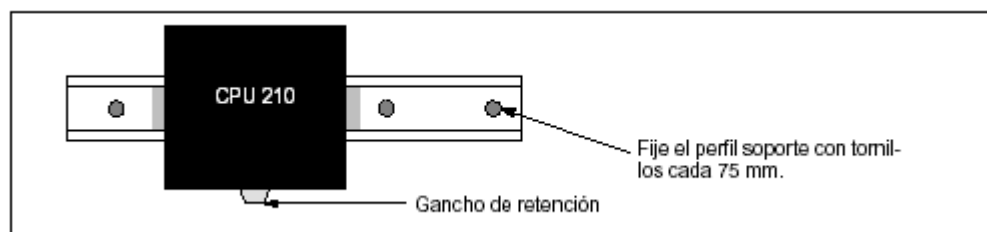
1. Posicione y taladre los orificios de sujeción para los tornillos de tamaño DIN M4 (estándar americano n° 8). En el apartado 1.2 se indican las dimensiones de montaje y otros datos importantes al respecto.
2. Atornille la CPU al tablero, utilizando tornillos de tamaño DIN M4 (estándar americano n° 8).

### 2.1.3.3. Montar una CPU 224 en un perfil soporte

Para montar una CPU 224 en un perfil soporte (Figura 2.5):

1. Fije el perfil soporte sobre el armario eléctrico utilizando tornillos cada 75 mm.
2. Abra el gancho de retención (ubicado en el lado inferior de la CPU) y enganche la parte posterior del módulo en el perfil soporte.
3. Cierre el gancho de retención y asegúrese de que el módulo haya enganchado correctamente en el perfil.

Si los módulos se montan en entornos donde se presenten vibraciones fuertes, o bien en posición vertical, puede ser necesario asegurarlos con frenos.



*Figura 2.5. Montaje de la CPU en un soporte*

Desconecte por completo la alimentación de la CPU 224 y de los equipos conectados antes de instalar o retirar el cableado de campo.

Tome siempre las medidas de seguridad necesarias y asegúrese de que la alimentación de la CPU está desconectada antes de proceder a su montaje.

#### ***2.1.4. Programación del Simatic.***

***Las CPUs S7-200 ofrecen numerosos tipos de operaciones que permiten solucionar una gran variedad de tareas de automatización. Disponen de dos juegos básicos de operaciones. El software de programación STEP 7-Micro/WIN 32 permite elegir entre diferentes editores para crear programas de control utilizando dichas operaciones. Por ejemplo,***

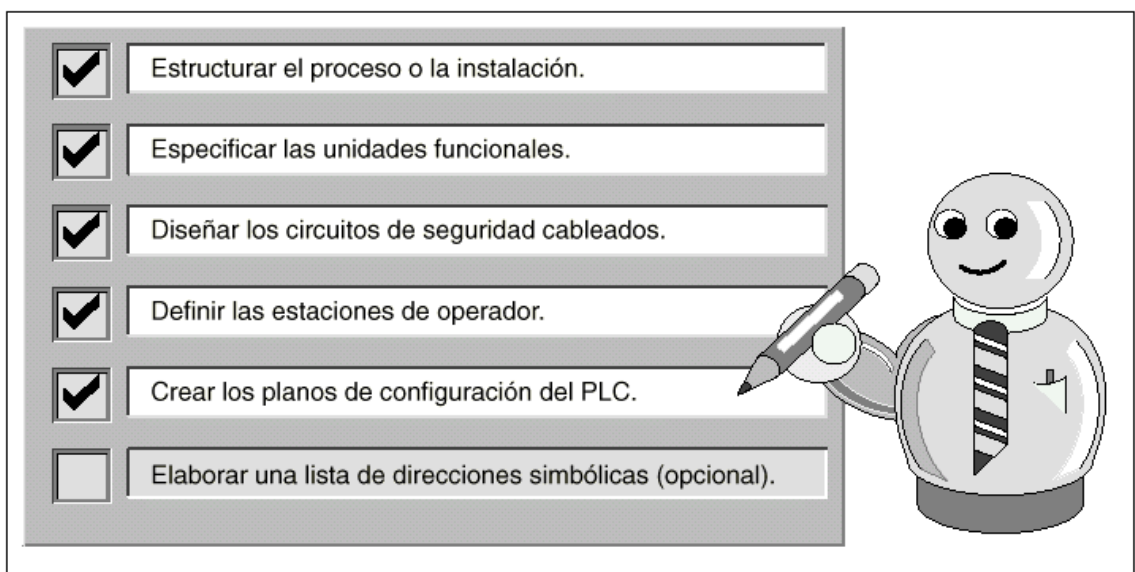
*puede ser que se prefiera crear programas en un entorno de programación gráfica, o que se opte por utilizar un editor textual, similar al lenguaje ensamblador.*

*Antes de comenzar a programar aplicaciones para la CPU S7-200, es recomendable que se familiarice con algunas funciones básicas de la misma.*

#### *2.1.4.1. Crear una solución de automatización con un Micro-PLC*

*Hay diversos métodos para crear una solución de automatización con un Micro-PLC. En el presente apartado se indican algunas reglas generales aplicables a numerosos proyectos. No obstante, también deberá tener en cuenta las reglas de su empresa y sus propias experiencias.*

*La figura 2.6 muestra los pasos básicos al respecto.*



*Figura 2.6 Pasos básicos para una solución de automatización*

#### ***2.1.4.2. Estructurar el proceso o la instalación***

***Estructure el proceso o la instalación en secciones independientes entre sí. Dichas secciones determinarán los límites entre los***



*diversos sistemas de automatización e influirán en las descripciones de las áreas de funciones y en la asignación de recursos.*

*2.1.4.3. Especificar las unidades funcionales*

*Describa las funciones de cada sección del proceso o de la instalación. Incorpore los siguientes aspectos:*

- *Entradas y salidas (E/S)*
  
- *Descripción del funcionamiento*
  
- *Condiciones de habilitación (es decir, los estados que se deben alcanzar antes de*

*ejecutar una función) de cada actuador (electroválvulas, motores, accionamientos, etc.)*

➤ *Descripción de la interface de operador*

➤ *Interfaces con otras secciones del proceso o de la instalación*

#### *2.1.4.4. Diseñar los circuitos de seguridad cableados*

*Determine qué aparatos requieren un cableado permanente por motivos de seguridad.*

*Si fallan los sistemas de automatización, puede producirse un arranque inesperado o un*

*cambio de funcionamiento de las máquinas que controlan. En tal caso, se pueden causar heridas graves o deteriorar objetos. Por lo tanto, es preciso utilizar dispositivos de protección contra sobrecargas electromecánicas que funcionen independientemente de la CPU, evitando así las condiciones inseguras.*

*Para diseñar los circuitos de seguridad cableados:*

- *Defina el funcionamiento erróneo o inesperado de los actuadores que pudieran causar peligros.*
  
- *Defina las condiciones que garanticen el funcionamiento seguro y determine cómo*

*reconocer dichas condiciones, independientemente de la CPU.*

➤ *Defina cómo la CPU y los módulos de ampliación deberán influir el proceso cuando se conecte y desconecte la alimentación, así como al detectarse errores. Estas informaciones se deberán utilizar únicamente para diseñar el funcionamiento normal y el funcionamiento anormal esperado, sin poderse aplicar para fines de seguridad.*

➤ *Prevea dispositivos de parada de emergencia manual o de protección contra sobrecargas electromagnéticas que impidan el*

*funcionamiento peligroso,  
independientemente de la CPU.*

➤ *Desde los circuitos independientes, transmita informaciones de estado apropiadas a la CPU para que el programa y las interfaces de operador dispongan de los datos necesarios.*

➤ *Defina otros requisitos adicionales de seguridad para que el proceso se lleve a cabo de forma segura y fiable.*

**2.1.4.5. Definir las estaciones de operador**

*Conforme a las funciones exigidas, cree planos de las estaciones de operador incorporando los siguientes puntos:*

- *Panorámica de la ubicación de todas las estaciones de operador con respecto al proceso o a la instalación.*
  
- *Disposición mecánica de los aparatos (pantalla, interruptores, lámparas, etc). de la estación de operador*
  
- *Esquemas eléctricos con las correspondientes entradas y salidas de la CPU o de los módulos de ampliación.*

#### ***2.1.4.6. Crear los planos de configuración del PLC***

***Conforme a las funciones exigidas, cree planos de configuración del sistema de automatización incorporando los siguientes puntos:***

- Ubicación de todas las CPUs y de todos los módulos de ampliación con respecto al proceso o a la instalación.***
  
- Disposición mecánica de las CPUs y de los módulos de ampliación (incluyendo armarios, etc).***
  
- Esquemas eléctricos de todas las CPUs y de los módulos de ampliación (incluyendo los números de referencia, las direcciones de***

*comunicación y las direcciones de las entradas y salidas).*

*2.1.4.7. Elaborar una lista de nombres simbólicos*

*Si desea utilizar nombres simbólicos para el direccionamiento, elabore una lista de nombres simbólicos para las direcciones absolutas. Incluya no sólo las entradas y salidas físicas, sino también todos los demás elementos que utilizará en su programa.*

*2.1.4.8. Programas S7-200*

**Referencias a las entradas y salidas en el programa**



*El funcionamiento básico de la CPU S7-200 es  
muy sencillo:*

- *La CPU lee el estado de las entradas.*
  
- *El programa almacenado en la CPU utiliza  
dichas entradas para evaluar la lógica.  
Durante la ejecución del programa, la CPU  
actualiza los datos.*
  
- *La CPU escribe los datos en las salidas.*

*La figura 2.7 muestra la conexión de un  
esquema de circuitos simple con la CPU S7-  
200. En este ejemplo, el estado del interruptor  
de la estación de operador para abrir la válvula*

*de vaciado se suma a los estados de otras  
entradas. Los cálculos de los mismos  
determinan entonces el estado de la salida para  
la electroválvula que cierra la válvula de  
vaciado.*

*La CPU procesa el programa cíclicamente,*

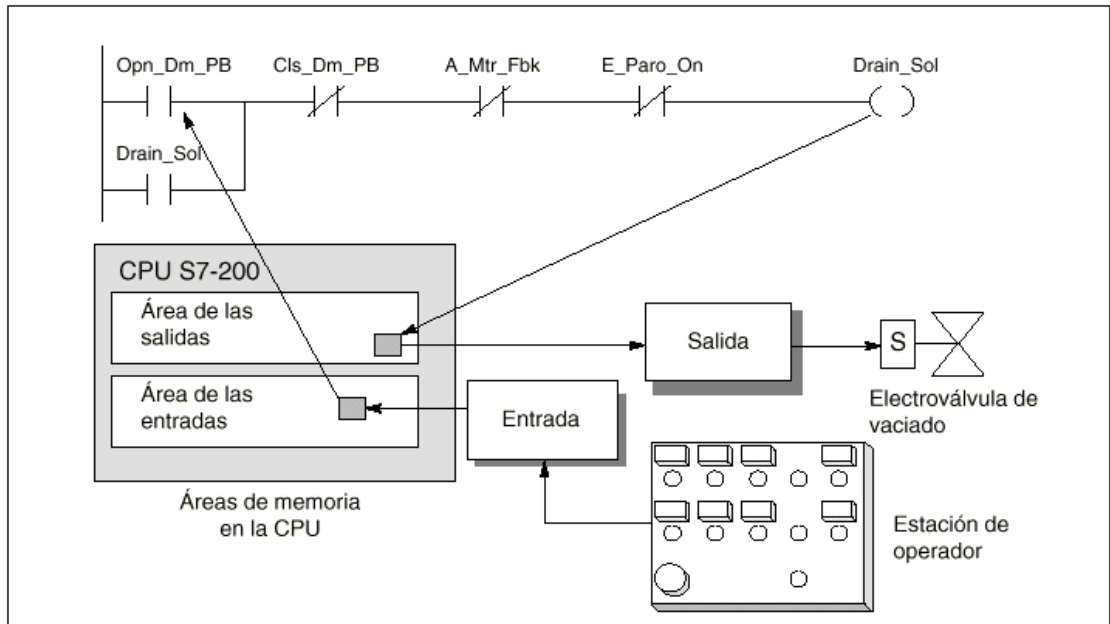


Figura 2.7 Referencias a las entradas v salidas leyendo y escribiendo los datos.

#### *2.1.4.9. Lenguajes de programación para las CPUs S7-200*

*Las CPUs S7-200 (y STEP 7-Micro/WIN)*

*asisten los siguientes lenguajes de*

*programación:*

- *La lista de instrucciones (AWL) comprende un juego de operaciones nemotécnicas que representan las funciones de la CPU.*
  
- *El esquema de contactos (KOP) es un lenguaje de programación gráfico con componentes similares a los elementos de un esquema de circuitos.*

*STEP7-Micro/WIN ofrece además dos representaciones nemotécnicas para visualizar las direcciones y las operaciones del programa: internacional y SIMATIC. Tanto la nemotécnica internacional como la de SIMATIC se refieren al mismo juego de operaciones del S7-200. Hay una correspondencia directa entre las dos representaciones, siendo idénticas las funciones de ambas.*

- *El editor FUP (Diagrama de Funciones) permite visualizar las operaciones en forma de cuadros lógicos similares a los circuitos de puertas lógicas.*

#### 2.1.4.9.1. Elementos básicos de KOP

*Al programar con KOP, se crean y se disponen componentes gráficos que conforman un segmento de operaciones lógicas. Como muestra la figura 2.8, se ofrecen los siguientes elementos básicos para crear programas:*

- *Contactos: un contacto representa un interruptor por el que circula la corriente cuando está cerrado.*
  
- *Bobinas: una bobina representa un relé que se excita cuando se le aplica tensión.*

- *Cuadros: un cuadro representa una función que se ejecuta cuando la corriente circula por él.*
  
- *Segmentos: cada uno de estos elementos constituye un circuito completo. La corriente circula desde la barra de alimentación izquierda pasando por los contactos cerrados para excitar las bobinas o cuadros.*

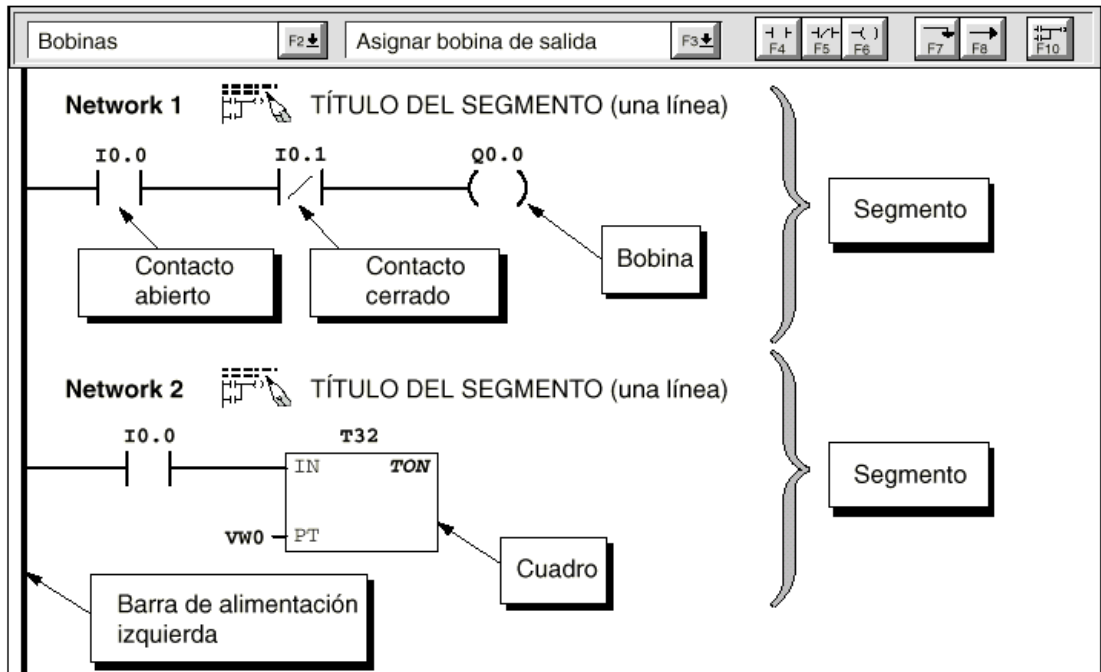


Figura 2.8 Elementos básicos de KOP

2.1.4.9.2. Operaciones de AWL

*La lista de instrucciones (AWL) es un lenguaje de programación en el que cada línea del programa contiene una operación que utiliza una abreviatura*

*nemotécnica para representar una  
función de la CPU.*

*Las operaciones se combinan en un  
programa, creando así la lógica de  
control de la aplicación.*

*La figura 2.9 muestra los elementos  
básicos de un programa AWL.*

```
Editor AWL - proyect1.ob1
//Programa para tren transportador
NETWORK //Marcha motor:
LD "Marcha1" //Si I0.0 está activada (on)
AN "Paro_Eml" //e I0.1 no está activada,
= Q0.0 //poner en marcha el motor del transportador.
NETWORK //Paro emergencia transportador:
LD I0.1 //Si Paro_Eml está activada
O I0.3 //o si Paro_Eml2 está activada,
R Q0.0, 1 //parar el motor del transportador.
NETWORK //Fin del programa
MEND
```

Comience cada comentario con dos barras inclinadas (//).

Operación

Operando

Figura 2.9 Ventana del editor AWL con un programa de ejemplo  
2.1.4.9.3. Editor FUP (Diagrama de Funciones)



El FUP permite visualizar las operaciones en forma de cuadros lógicos similares a los circuitos de puertas lógicas. En FUP no existen contactos ni bobinas como en el editor KOP, pero si hay operaciones equivalentes que se representan en forma de cuadros.



La lógica del programa se deriva de las conexiones, entre dichas operaciones de cuadro.

Ello significa que la salida de una operación (p. ej. Un cuadro AND) se puede utilizar para habilitar otra operación (p. ej. un temporizador) para crear la lógica de control necesaria. Dichas conexiones permiten solucionar numerosos problemas lógicos.

A continuación se indican los aspectos principales a considerar cuando se desee utilizar el editor FUP:

- El estilo de presentación en forma de puertas gráficas se adecua especialmente para observar el flujo del programa.

- El editor FUP se puede utilizar con los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL siempre se puede utilizar para visualizar un programa creado en SIMATIC FUP.

KOP	AWL
<b>Programa principal</b>	
<p>Network 1</p>  <p>Network 2</p> 	<p>Network 1</p> <pre>LD    SM0.1    //Si se activa la            //marca del primer ciclo CALL  0        //Llamar subrutina 0.</pre> <p>Network 2</p> <pre>MEND</pre>
<b>Subrutinas</b>	
<p>Network 3</p>	<p>Network 3</p>

#### ***2.1.4.10. Elementos básicos para estructurar un programa***

La CPU S7-200 ejecuta continuamente el programa para controlar una tarea o un proceso. El programa se crea con STEP

7-Micro/WIN y se carga en la CPU. Desde el programa principal se pueden llamar diversas subrutinas o rutinas de interrupción.

#### 2.1.4.10.1. Estructurar el programa

Los programas para la CPU S7-200 comprenden tres partes básicas: el programa principal, las subrutinas (opcional) y las rutinas de interrupción (opcional). Como muestra la figura 2.10, un programa S7-200 se estructura mediante los siguientes elementos:

➤ *Programa principal: En esta parte del programa se disponen las operaciones que controlan la aplicación. Las operaciones del programa principal se ejecutan de forma secuencial en cada ciclo de la CPU. Para terminar el programa principal, utilice en KOP una*

*bobina absoluta Finalizar programa principal, o en AWL una operación Finalizar programa principal (MEND). Vea 2.10 en la figura.*

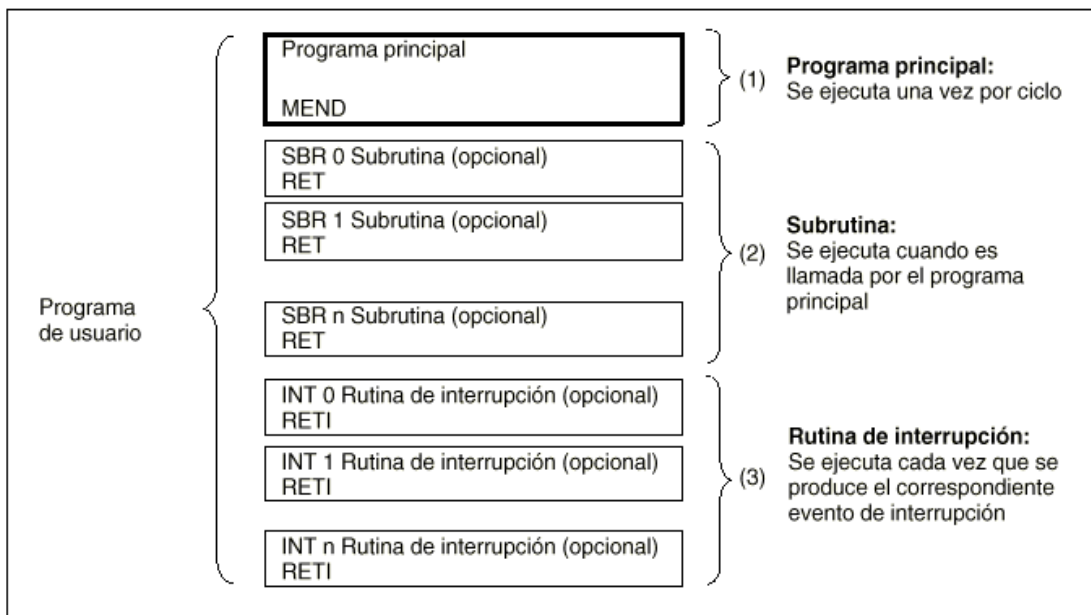
➤ *Subrutinas: Estos elementos opcionales del programa se ejecutan sólo cuando se llaman desde el programa principal. Se deben añadir siempre al final del programa principal (detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una*

*operación Retorno absoluto (RET)  
para terminar cada subrutina.*

➤ *Rutinas de interrupción: Estos elementos opcionales del programa se ejecutan cada vez que se presente el correspondiente evento de interrupción. Se deben añadir siempre al final del programa principal (detrás de la bobina absoluta Finalizar un programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una operación Retorno absoluto desde rutina de interrupción (RETI) para terminar cada rutina de interrupción. Las*

*subrutinas y las rutinas de interrupción se deben añadir detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL. No hay reglas adicionales en lo relativo a su disposición en el programa de usuario. Las subrutinas y las rutinas de interrupción se pueden mezclar a voluntad después del programa principal. No obstante, para que la estructura del programa sea fácil de leer y comprender, es recomendable agrupar al final del programa principal primero todas*

*las subrutinas y, después, todas las rutinas de interrupción.*



*Figura 2.10 Estructura del programa de una CPU S7-200*



#### 2.1.4.10.2. Programa de ejemplo con subrutinas y rutinas de interrupción

El programa de ejemplo representado en la figura 2.11.se muestra una interrupción temporizada que se puede utilizar para leer el valor de una entrada analógica.

KOP	AWL
<b>Programa principal</b>	
<p>Network 1</p> <p>Network 2</p>	<p>Network 1</p> <pre>LD    SM0.1    //Si se activa la                 //marca del primer ciclo CALL  0        //Llamar subrutina 0.</pre> <p>Network 2</p> <pre>MEND</pre>
<b>Subrutinas</b>	
<p>Network 3</p> <p>Network 4</p> <p>Network 5</p>	<p>Network 3</p> <pre>SBR  0        //Comenzar subrutina 0</pre> <p>Network 4</p> <pre>LD    SM0.0    //Marca de funcionamiento                 //continuo, MOVB  100, SMB34 //Ajustar a 100 ms el intervalo                 //de la int. temporiz. 0. ENI                    //Habilitar todos los eventos ATCH  0, 10      //Asociar int. temp. 0 a                 //rutina de interrup. 0.</pre> <p>Network 5</p> <pre>RET          //Fin subrutina.</pre>
<b>Rutinas de interrupción</b>	
<p>Network 6</p> <p>Network 7</p> <p>Network 8</p>	<p>Network 6</p> <pre>INT  0        //Comenzar rutina de int. 0.</pre> <p>Network 7</p> <pre>MOVW AIW4, VW100 //Mostrar entrada analógica 4</pre> <p>Network 8</p> <pre>RETI          //Finalizar rutina                 de interrup.</pre>

Figura 2.11 Programa de ejemplo con una subrutina

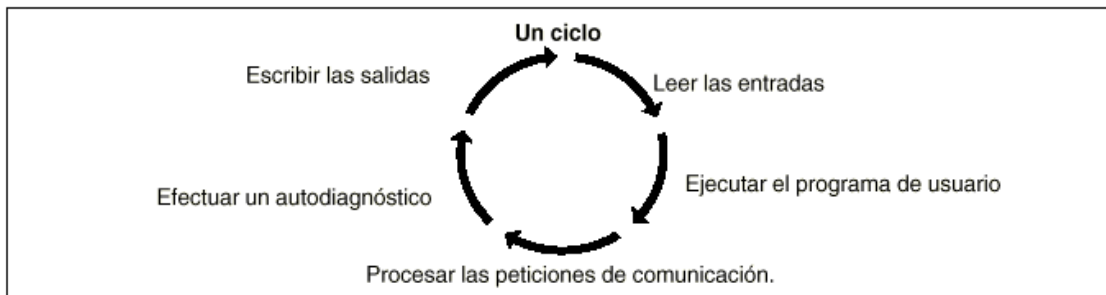
#### 2.1.4.11. El ciclo de la CPU

*La CPU S7-200 se ha previsto para que ejecute cíclicamente una serie de tareas, incluyendo el programa de usuario. Dicha ejecución se denomina ciclo.*

*Durante el ciclo que se muestra en la figura 2.12., la CPU ejecuta la mayoría de las tareas siguientes (o todas ellas):*

- *Lee las entradas*
  
- *Ejecuta el programa de usuario.*
  
- *Procesa las peticiones de comunicación.*
  
- *Efectúa un autodiagnóstico.*

➤ *Escribe las salidas.*



*Figura 2.12: Ciclo de la CPU S7-200*

*La serie de tareas que se ejecutan durante el ciclo depende del modo de operación de la CPU.*

*La CPU S7-200 tiene dos modos de operación: STOP y RUN. Con respecto al ciclo, la principal*

*diferencia entre STOP y RUN es que el programa se ejecuta al estar la CPU en modo*

*RUN, mas no en STOP.*

#### 2.1.4.11.1. Leer las entradas digitales

Al principio de cada ciclo se leen los valores actuales de las entradas digitales y se escriben luego en la imagen del proceso de las entradas.

***La CPU reserva un espacio de la imagen del proceso de las entradas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proporcionan una entrada física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S o utilizarlos en el programa de usuario.***

*Al comienzo de cada ciclo, la CPU pone a 0 estos bits no utilizados en la imagen del proceso. No obstante, si la CPU asiste varios módulos de ampliación y no se está utilizando su capacidad de E/S (porque no se han instalado los módulos de ampliación), los bits de entradas de ampliación no utilizados se pueden usar como marcas internas adicionales.*

*La CPU no actualiza automáticamente las entradas analógicas como parte del ciclo y no ofrece una imagen del proceso para las mismas. A las entradas analógicas*

*se debe acceder directamente desde el programa de usuario.*

2.1.4.11.2. Ejecutar el programa

Durante esta fase del ciclo, la CPU ejecuta el programa desde la primera operación hasta la última (Fin del programa). El control directo de las entradas y salidas permite acceder directamente a las mismas mientras se ejecuta el programa o una rutina de interrupción.

*Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa. Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre el evento (en cualquier punto del ciclo).*

#### 2.1.4.11.3. Procesar las peticiones de comunicación

Durante esta fase del ciclo, la CPU procesa los mensajes que haya recibido por la interface de comunicación.

#### 2.1.4.11.4. Efectuar el autodiagnóstico de la CPU

Durante el autodiagnóstico se comprueba el firmware de la CPU y la memoria del programa (sólo en modo RUN), así como el estado de los módulos de ampliación.

#### 2.1.4.11.5. Escribir las salidas digitales

Al final de cada ciclo, la CPU escribe los valores de la imagen del proceso de las salidas en las salidas digitales.



*La CPU reserva un espacio de la imagen del proceso de las salidas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proveen una salida física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S. No obstante, los bits no utilizados de la imagen del proceso de las salidas se pueden usar como marcas internas adicionales.*

*La CPU no actualiza automáticamente las salidas analógicas como parte del ciclo y no ofrece una imagen del proceso para*

*las mismas. A las salidas analógicas se debe acceder directamente desde el programa de usuario.*

*Cuando el modo de operación de la CPU se cambia de RUN a STOP, las salidas digitales adoptan los valores definidos en la tabla de salidas o conservan su estado actual. Las salidas analógicas conservan su último valor.*

#### 2.1.4.11.6. Interrumpir el ciclo

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan como parte del programa. Las rutinas de interrupción no se

ejecutan como parte del ciclo, sino sólo cuando ocurre el evento (en cualquier punto del ciclo). La CPU procesa las interrupciones según su prioridad y después en el orden que aparecen.

#### 2.1.4.11.7. Imagen del proceso de las entradas y salidas

Por lo general, es recomendable utilizar la imagen del proceso, en vez de acceder directamente a las entradas o salidas durante la ejecución del programa. Las imágenes del proceso existen por tres razones:

➤ *El sistema verifica todas las entradas al comenzar el ciclo. De este modo se sincronizan y "congelan" los valores de estas entradas durante la ejecución del programa. La imagen del proceso actualiza las salidas cuando*

*termina de ejecutarse el programa.*

*Ello tiene un efecto estabilizador en el sistema.*

➤ *El programa de usuario puede acceder a la imagen del proceso mucho más rápido de lo que podría acceder directamente a las entradas y salidas físicas, con lo cual se acelera su tiempo de ejecución.*

➤ *Las entradas y salidas son unidades de bit a las que se debe acceder en formato de bit. No obstante, la imagen del proceso permite acceder a ellas en formato de bits, bytes, palabras y palabras dobles, lo que ofrece flexibilidad adicional.*

*Otra ventaja es que las imágenes del proceso son lo suficientemente grandes para poder procesar el número máximo de entradas y salidas. Puesto que un sistema real comprende tanto entradas como salidas, en la imagen del proceso existe siempre un número de direcciones que no se utilizan. Estas direcciones libres pueden utilizarse como marcas internas adicionales.*

#### 2.1.4.11.8. Control directo de las entradas y salidas

Las operaciones de control directo de las entradas y salidas (E/S) permiten acceder a la entrada o salida

física, aunque el acceso a las E/S se efectúa por lo general a través de las imágenes del proceso. El acceso directo a una entrada no modifica la dirección correspondiente en la imagen del proceso de las entradas. En cambio, el acceso directo a una salida actualiza simultáneamente la dirección correspondiente en la imagen del proceso de las salidas.

#### ***2.1.4.12. Ajustar el modo de operación de la CPU***

***La CPU S7-200 tiene dos modos de operación:***

➤ ***STOP: La CPU no ejecuta el programa.***

***Cuando está en modo STOP, es posible cargar programas o configurar la CPU.***

➤ *RUN: La CPU ejecuta el programa. Cuando está en modo RUN, no es posible cargar programas ni configurar la CPU.*

*El diodo luminoso (LED) en la parte frontal de la CPU indica el modo de operación actual.*

*Para poder cargar un programa en la memoria de la CPU es preciso cambiar a modo STOP.*

#### 2.1.4.12.1. Cambiar el modo de operación con el selector

El modo de operación de la CPU se puede cambiar manualmente accionando el selector ubicado debajo de la tapa de acceso a la CPU:

➤ *Si el selector se pone en STOP, se detendrá la ejecución del programa.*

➤ *Si el selector se pone en RUN, se iniciará la ejecución del programa.*

➤ *Si el selector se pone en TERM (terminal), no cambiará el modo de operación de la CPU. No obstante, será posible cambiarlo utilizando el software de programación (STEP 7-Micro/WIN).*

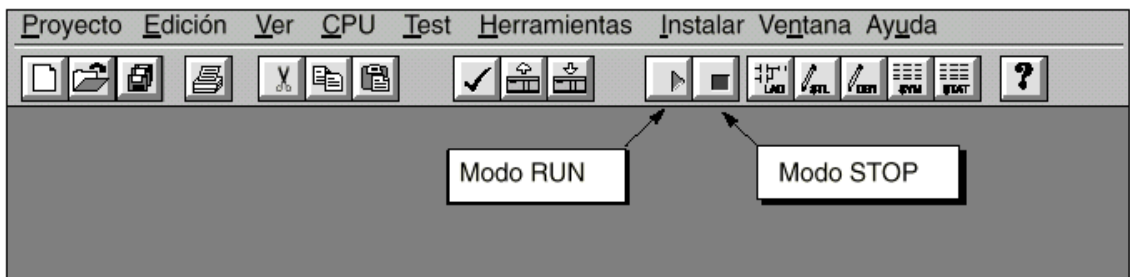
*Si se interrumpe la alimentación estando el selector en posición STOP o TERM, la CPU pasará a modo STOP cuando se le aplique tensión. Si se interrumpe la alimentación estando el selector en posición RUN, la CPU*



*pasará a modo RUN cuando se le  
aplique tensión.*

#### 2.1.4.12.2. Cambiar el modo de operación con STEP 7- Micro/WIN

Como muestra la figura 2.13, el modo de operación de la CPU se puede cambiar también con STEP 7-Micro/WIN. Para que ello sea posible, el selector de la CPU deberá estar en posición TERM o RUN.



*Figura 2.13 Utilizar STEP 7-Micro/WIN para cambiar el modo de operación de la  
CPU*

#### 2.1.4.12.3. Cambiar el modo de operación desde el programa

Para cambiar la CPU a modo STOP es posible introducir la correspondiente operación (STOP) en el programa. Ello permite detener la ejecución del programa en función de la lógica.

#### ***2.1.4.13. Determinar Una Contraseña para La CPU***

***Todas las CPUs S7-200 ofrecen una protección con contraseña para restringir el acceso a determinadas funciones. Con una contraseña se puede acceder a las funciones y a la memoria de la CPU. Si no se utiliza la opción de contraseña, la CPU permite un acceso ilimitado. Si está protegida con una contraseña, la CPU prohíbe todas las operaciones restringidas conforme a la***

***configuración definida al determinar la contraseña.***

2.1.4.13.1. Restringir el acceso a la CPU

Como muestra la tabla 2.1, las CPUs S7-200 ofrecen tres niveles de protección para acceder a sus funciones.

Cada uno de dichos niveles permite ejecutar determinadas funciones sin la contraseña.

***Si se introduce la contraseña correcta, es posible acceder a todas las funciones de la CPU. El ajuste estándar para las CPUs S7-200 es el nivel 1 (privilegios totales).***

*Si se introduce la contraseña a través de una red, no se afecta la protección con contraseña de la CPU. Si un usuario tiene acceso a las funciones restringidas de la CPU, ello no autoriza a los demás usuarios a acceder a dichas funciones. El acceso ilimitado a las funciones de la CPU sólo se permite a un usuario a la vez.*

**Una vez introducida la contraseña, el nivel de protección se conservará aproximadamente durante un minuto después de haber desconectado la unidad de programación de la CPU.**

Tarea	Nivel 1	Nivel 2	Nivel 3
Leer y escribir datos de usuario	No restringido	No restringido	No restringido
Arrancar, detener y rearrancar la CPU			
Leer y escribir el reloj de tiempo real			
Leer los datos forzados en la CPU		Restringido	Restringido
Cargar en la PG el programa de usuario, los datos y la configuración			
Cargar en la CPU		Restringido	
Borrar el programa de usuario, los datos y la configuración <sup>1</sup>			
Forzar datos o ejecutar uno/varios ciclo(s)			
Copiar en el cartucho de memoria			

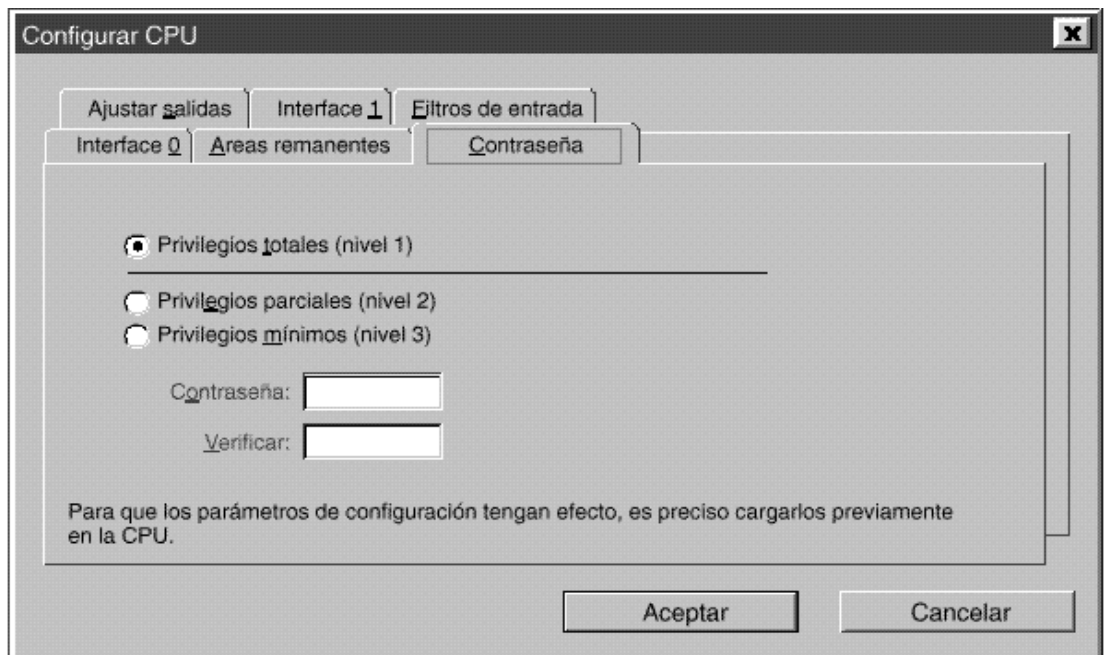
**Tabla 2.1 Restringir el acceso a la CPU S7-200**

2.1.4.13.2. Configurar la contraseña para la CPU

STEP 7-Micro/WIN permite determinar una contraseña para acceder a las funciones de la CPU.

➤ *Elija el comando de menú CPU \_  
Configurar y seleccione la ficha  
"Contraseña" ver la figura 2.14.*

➤ *Indique el nivel de protección deseado. Introduzca y verifique luego la contraseña.*



*Figura 2.14 Configurar una contraseña para la CPU*

2.1.4.13.3. Qué hacer si se olvida la contraseña

Si se olvida la contraseña es preciso efectuar un borrado total de la memoria de la CPU y volver a cargar el programa. Al borrar la memoria de la CPU, ésta pasa a modo STOP y recupera los ajustes predeterminados, con excepción de la dirección de estación y del reloj de tiempo real.

*Si desea borrar el programa de la CPU, elija el comando de menú CPU \_Borrar... para visualizar el cuadro de diálogo "Borrar CPU". Elija la opción "Todo" y confirme su acción haciendo clic en el botón "Aceptar". Entonces se visualizará un cuadro de diálogo donde deberá introducir la contraseña "clearplc" que permitirá iniciar el borrado total.*

*La función de borrado total no borra el programa contenido en el cartucho de memoria. Puesto que en éste último se encuentra almacenado no sólo el programa, sino también la contraseña, es preciso volver a programar también dicho cartucho para borrar la contraseña olvidada.*

#### *Precaución*

*Al efectuarse un borrado total de la CPU, se desactivan las salidas (las salidas analógicas se congelan en un valor determinado).*



*Si la CPU S7-200 está conectada a otros equipos durante el borrado total, es posible que los cambios de las salidas se transfieran también a dichos equipos. Si ha determinado que el "estado seguro" de las salidas sea diferente al ajustado de fábrica, es posible que los cambios de las salidas provoquen reacciones inesperadas en los equipos conectados, lo que podría causar la muerte o heridas graves personales y/o daños materiales.*

*Adopte siempre las medidas de seguridad apropiadas y asegúrese de que su proceso se encuentra en un estado seguro antes de efectuar un borrado total de la CPU.*

### 2.1.5. Funciones Básicas.

En la tabla 2.2. se resumen las funciones más importantes de la CPU 224.

Función		CPU 210
Dimensiones (longitud x ancho x profundidad)		90 x 80 x 62 mm
Cartucho de memoria para cargar el programa en la CPU		Si
Memoria	Tamaño del programa	256 palabras
	Tipo de almacenamiento	EEPROM
	Marcas	48 bits (3 palabras)
Entradas/salidas (E/S)	Entradas integradas	4 entradas digitales
	Salidas integradas	4 salidas digitales
	Módulos de ampliación	No
	Filtro de retardo de entrada DC	15 ms
	Filtro de retardo de salida AC	55 ms
	Entradas con sumidero de corriente/fuente (DC)	Si
Operaciones (36 en total)	Velocidad de ejecución booleana	95 $\mu$ s/operación
	Temporizadores de retardo a la conexión	4
	Resolución	100 ms
	Contadores adelante/atrás	4
	Almacenamiento del valor actual al interrumpirse la alimentación	Si
Funciones adicionales	Salto / meta	Si
	Potenciómetros analógicos	1
	Entradas de interrupción de hardware	1
	Respuesta a interrupciones	20 $\mu$ s on, 40 $\mu$ s off

Tabla 2.2 Funciones Básicas

## 2.2 El Software Scada Intouch

### 2.2.1 Introducción a Scada Intouch

Primeramente podemos definir que el término Scada viene de las siglas “Supervisory Control and Data Acquisition” , es decir adquisición de datos y control de supervisión, por lo tanto InTouch es un paquete de software utilizado para crear aplicaciones de interface hombre-máquina bajo entorno *PC*. InTouch utiliza como sistema operativo el entorno *WINDOWS 95/98/NT/2000*.

El paquete consta básicamente de dos elementos: *WINDOWMAKER* y *WINDOWVIEWER*.

*WINDOWMAKER* es el sistema de desarrollo. Permite todas las funciones necesarias para crear ventanas animadas interactivas conectadas a sistemas de e/s externos o a otras aplicaciones.

*WINDOWS WINDOWVIEWER* es el sistema runtime utilizado para rodar las aplicaciones creadas con *WINDOWMAKER*.

En cualquier pantalla de *WINDOWMAKER* disponemos de una ayuda sensitiva pulsando la tecla F1.



Este es el icono de *WINDOWMAKER* o creador de aplicaciones. Seleccionada la aplicación que desea crear o modificar, pulse sobre este icono para llevar a cabo su trabajo.



Este es el icono de *WINDOWVIEWER* o runtime. Una vez seleccionada la aplicación que desea monitorizar, pulse sobre este icono. Esta aplicación debe haber sido previamente creada, por lo que este icono no estará accesible cuando seleccione nueva aplicación.

Al pulsar sobre el icono de WindowMaker, **InTouch** creará automáticamente subdirectorio con este nombre, e incluirá en él los ficheros de trabajo.

#### 2.2.1.1. Creación de una Aplicación

Una vez instalado el paquete InTouch, ya podemos crear una aplicación. Para ello, es necesario entrar en InTouch desde

WINDOWS presionando dos veces con el ratón sobre el símbolo de InTouch.

En su monitor aparecerá la siguiente pantalla (Ver Figura 2.15.):

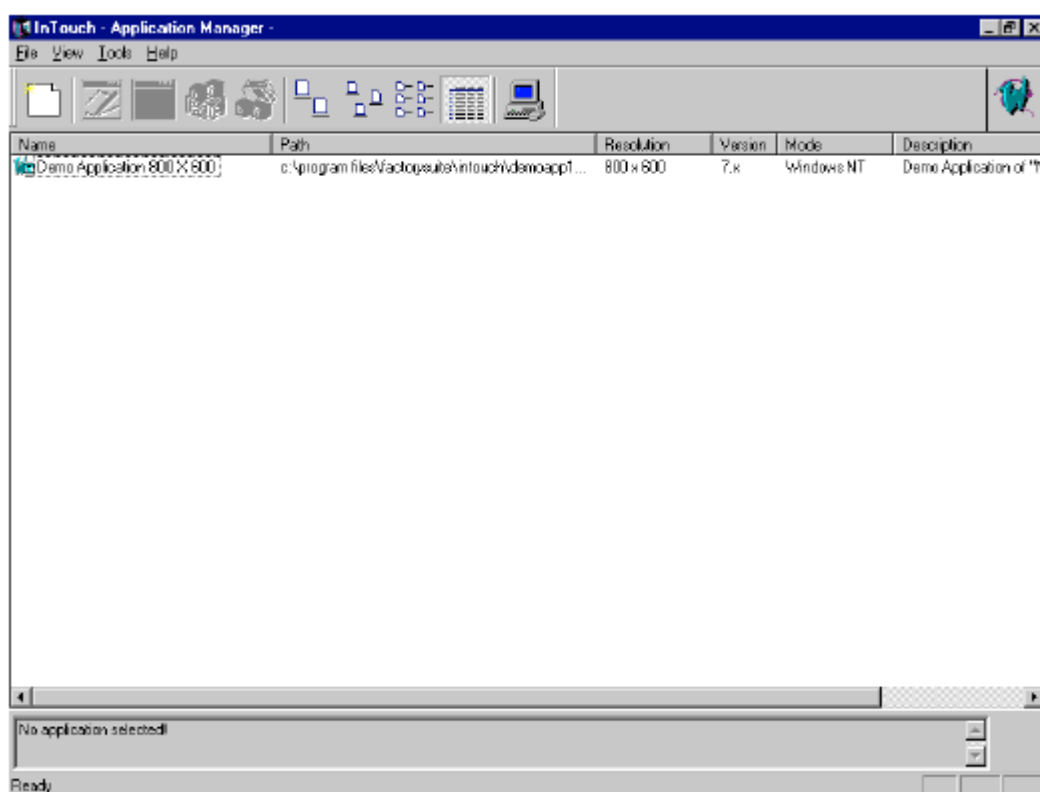


Figura 2.15 Pantalla principal de Application Manager para la entrada a **InTouch**.

Desde aquí podemos seleccionar cualquiera de las aplicaciones de nuestro ordenador previamente creadas, o bien crear una aplicación nueva.

Para ello, seleccione *FILE NEW* para acudir al asistente de generación de aplicaciones, que le permitirá además dar un nombre comentario a la nueva aplicación creada (muy útil tanto para documentación posterior, como para selección desde el application manager).

## 2.2.2. Entorno de Desarrollo

### 2.2.2.1. Elementos de WindowMarker

#### 2.2.2.1.1. Menús de WindowMarker

<b>File.-</b> Manejo de ficheros y de ventanas. Contiene los siguientes elementos:	
New Window	Crear una nueva ventana
Open Window	Abrir una ventana existente
Save Window	Salvar una ventana
Close Window	Cerrar una ventana. Si no está salvada, <b>InTouch</b> nos consultará

Delete Window	Borra una ventana
Save Window As	Salva una ventana con un nombre distinto
Save All Windows	Salva todas las ventanas abiertas
WindowViewer	Salta o ejecuta el programa WindowViewer
Print	Permite imprimir: Información sobre los tags, ventanas y también scripts
Export Window	Exporta ventana a otra aplicación <b>InTouch</b>
Import	Importa ventana de otra aplicación <b>InTouch</b>
Exit	Salte de WindowMaker

**Edit.-** Contiene una serie de comandos para editar los objetos de la ventana. Con estas funciones, podemos editar los objetos que se encuentren seleccionados

Undo	Permite deshacer la última acción de edición. Dispone de hasta 25 niveles
Nothing to redo	Rehace la última acción de edición deshecha. Dispone de hasta 25 niveles
Duplicate	Duplica el/los objeto/s seleccionado/s
Cut	Cortar al portapapeles
Copy	Copiar al portapapeles
Paste	Pegar al portapapeles
Erase	Borrar

Import Image	Importar imagen
Paste Bitmap	Pegar Bitmap
Bitmap Original Size	Devolver al bitmap su tamaño original tras haberlo pegado
Edit Bitmap	Editar bitmap tras haberlo pegado
Select All	Seleccionar todo
Links	Links
Enlarge radius	Agrandar el radio de una curva del objeto seleccionado
Reduce radius	Reducir el radio de una curva del objeto seleccionado
Reshape Object	Deformar el objeto
Add Point	Añadir punto (en un objeto polígono o polilínea)
Del point	Borrar punto (en un objeto polígono o polilínea)
Symbol Factory	Llamada al programa Symbol Factory

*Tabla 2.3 Menús de WindowMarker*

#### 2.2.2.2. Usando WindowMaker

##### 2.2.2.2.1. Tipos De Ventanas.

**InTouch** trabaja con ventanas o pantallas. Estas ventanas disponen de:



- Elementos animados
  
- Tendencias gráficas y alarmas
  
- Lógica Asociada

Antes de empezar a dibujar es necesario definir la pantalla sobre la que vamos a trabajar.

Las pantallas pueden ser de tres tipos (Ver Tabla 2.4.):

<i>Replace</i>	Cierra cualquier otra ventana que corte cuando aparece en pantalla, incluyendo ventanas tipo <i>popup</i> u otras tipo <i>replace</i>
<i>Overlay</i>	Aparece sobre la ventana desplegada. Cuando cerramos una ventana tipo <i>overlay</i> , cualquier ventana que estuviera <i>escondida</i> bajo la <i>overlay</i> será restablecida. Seleccionando cualquier porción o parte visible de una ventana debajo de la <i>overlay</i> , provocará que esta ventana pase a ser considerada activa
<i>Popup</i>	Similar a la <i>overlay</i> , pero en el caso de <i>popup</i> la ventana siempre queda por encima de las demás, y no desaparece ni aunque pinchemos con el ratón sobre otra. Normalmente será necesario hacer desaparecer la ventana <i>popup</i> antes de que aparezca otra.

*Tabla 2.4. Tipos de Ventanas de Intouch*

#### 2.2.2.2.2. Los Elementos Wizard.

**WIZARDS**, en su más básico concepto, podría ser definido como "elementos inteligentes" que permiten que las aplicaciones **InTouch** puedan ser generadas de un modo más rápido y eficiente. La versión 7.1 de **InTouch** dispone de los elementos **WIZARDS** que permiten crear rápidamente un objeto en la pantalla. Haciendo doble click sobre el objeto podemos asociarle links (animación), asignarlo a tagnames o incluso incluir una lógica en ese objeto. Si agrupamos varios de estos objetos, podemos crear un elemento completo, acabado y programado, que lo podemos utilizar tantas veces como se desee.

Además de estos **WIZARDS** "sencillos", es posible utilizar otros más "complejos" que provoquen operaciones en background, tales como crear/convertir una base de datos, importar un fichero AutoCad,

configurar módulos de software (p.e. recetas, SPC), etc.

Ello es posible gracias a la herramienta Wonderware Extensibility Toolkit (opcional de **InTouch**). La mayoría de **WIZARDS** son escalables y configurables en tamaño. Ello le permitirá modificar y poder ajustar los dibujos ya hechos a un tamaño necesario para su ventana.

### 2.2.3. El Diccionario de Tagname

#### 2.2.3.1. Definición de Tagname

El diccionario de tagnames es el corazón de **InTouch**. Durante el runtime, este diccionario contiene todos los valores de los elementos en la base de datos. Para crear esa base de datos, **InTouch** necesita saber qué elementos la van a componer. Debemos, por lo tanto, crear una base de datos con todos aquellos datos que necesitemos para nuestra aplicación. A cada uno de estos datos (tags) debemos asignarle un nombre.

Al final, dispondremos de un diccionario con todos los tagnames o datos que nosotros mismos hemos creado.

#### *Acceso*

A este diccionario se accede desde el menú */Special/Tagname Dictionary*.

#### 2.2.3.2. Definición de Tagnames.

Desde el diccionario de tagnames definimos los tagnames y sus características. Existen diversos tipos de tagnames, según su función o características (Tabla 2.5). Básicamente se dividen en:

<i>MEMORY</i>	Tags registros internos de <b>InTouch</b>
<i>I/O</i>	Registros de enlace con otros programas
<i>INDIRECT</i>	Tags de tipo indirecto
<i>GROUP VAR</i>	Tags de los grupos de alarmas
<i>HISTTREND</i>	Tag asociado a los gráficos históricos
<i>TagID</i>	Información acerca de los tags que están siendo visualizados en una gráfica histórica

*Tabla 2.5. Tipos de Tagnames según su función*

De los 3 primeros tipos, disponemos de:

<i>Discrete</i>	Puede disponer de un valor 0 ó 1
<i>Integer Discrete</i>	Puede disponer de un valor 0 ó 1
<i>Integer Tagname</i>	32 bits con signo. Su valor va desde -2.147.483.648 hasta 2.147.483.647
<i>Real Tagname</i>	en coma flotante. Su valor va entre $\pm 3.4e38$ . Todos los cálculos son hechos en 64 bits de resolución, pero el resultado se almacena en 32 bits
<i>Message Tagname</i>	alfanumérico de hasta 131 caracteres de longitud
<i>Real</i>	Tagname en coma flotante. Su valor va entre $\pm 3.4e38$ . Todos los cálculos son hechos en 64 bits de resolución, pero el resultado se almacena en 32 bits
<i>Message</i>	Tagname alfanumérico de hasta 131 caracteres de longitud

*Tabla 2.6. Tipos de Tagnames*

Existe otro tipo de tags, los System Tagnames. Se trata de tags del sistema, que nos dan información acerca de parámetros tales como fecha/hora, errores de impresora, actividad del Historical Logging, etc. Todos estos tags empiezan por el signo \$

#### 2.2.3.3. Definición de las Características

Desde el diccionario de tagnames definimos sus características  
(Ver Tabla 2.7)

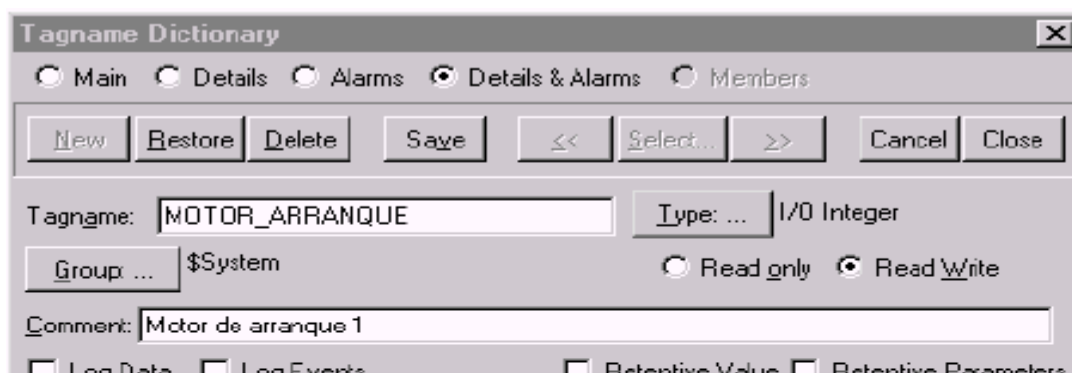
- Main
  
- Details
  
- Alarms
  
- Details&Alarms
  
- Members

<i>Main</i>	Visualiza las características principales del tagname
<i>Details</i>	Visualiza las características del tag que va a crear (valor mínimo/máximo, etc.)
<i>Alarms</i>	Visualiza las condiciones de alarma del tag
<i>Details&amp;Alarms</i>	Le permitirá de visualizar las características del tagname tanto de detalles como de alarma
<i>Members</i>	Visualiza Miembros caso de ser supertag

*Tabla 2.7. Definición de las Características*

#### 2.2.3.4. Campos a Rellenar del Tagname

Una vez seleccionado el tipo de tagname y qué características debemos definir, un submenú aparecerá para que rellenemos los campos de ese tagname. Ver figura 2.16



<input type="radio"/> Read only	<input checked="" type="radio"/> Read Write	Permite lectura/escritura o sólo lectura del registro
<input checked="" type="checkbox"/> Log Data		Graba el valor del tag al fichero de históricos cuando varía más que lo especificado en <i>Log Deadband</i>
<input type="checkbox"/> Log Events		Activa la grabación de eventos para ese tag
<input type="checkbox"/> Retentive Value		Permite que el valor current del registro sea retentivo
<input checked="" type="checkbox"/> Retentive Parameters		Permite retener los cambios del registro de cualquier campo de límites de alarmas
Initial Value:	20	Selecciona el valor inicial del registro
Min EU:	-10	Introduzca el valor en unidades de ingeniería del registro equivalente al mínimo recibido

Max EU:	150	Introduzca el valor en unidades de ingeniería del registro equivalente al máximo recibido
Deadband:	0	Permite definir cuánto debe cambiar el valor de un registro para ser actualizado en pantallas
Min Raw:	-32768	Valor mínimo en el rango de valores enteros del valor I/O
Max Raw:	32767	Valor máximo en el rango de valores enteros del valor I/O
Access Name: ...		Seleccione el programa de acceso
<input type="checkbox"/> Conversion <input checked="" type="radio"/> Linear <input type="radio"/> Square Root		Seleccione si quiere una conversión lineal o de raíz cuadrada
<input checked="" type="checkbox"/> Use TagName as Item Name		Seleccione esta opción para displayar el tagname como nombre del item I/O



*Figura 2.16. Campos a Rellenar del Tagname*

#### 2.2.4. Los Animation Links

Tras haber creado un objeto gráfico o un símbolo, éste puede ser animado mediante las *Animation Links*. Las *Animation Links* provocan que el objeto cambie de apariencia reflejando cambios en los valores de la base de datos. Por ejemplo, una válvula puede cambiar de color de acuerdo a si está o no activa.

##### 2.2.4.1. ¿Qué son las Animation Links?

Para asignar una *animation link* a un objeto, éste deberá estar seleccionado. Haciendo dos veces click sobre el objeto o símbolo deseado entramos directamente en el menú de *Animation Links*.

#### 2.2.4.2. Animando Objetos

Una vez que un objeto gráfico es creado dentro de la ventana de aplicación, éste puede ser “animado” es decir puede cambiar su color, visibilidad, ubicación en la ventana, orientación, ser sensible al clic del mouse, etc. Para lograr estos efectos es necesario definir *Animation Links* a dicho objeto.

#### 2.2.4.3. Edición de Links y de Tags.

##### 2.2.4.3.1. Sustituir Texto

Desde *SPECIAL ->SUBSTITUTE STRINGS* es posible modificar el texto que tengamos seleccionado.

##### 2.2.4.3.2. Sustituir Tagnames

Cuando duplicamos un objeto, este se convierte en una exacta réplica del duplicado. Si necesitamos usar un tagname distinto para el nuevo objeto, podemos utilizar *SPECIAL ->SUBSTITUTE TAGS*

#### 2.2.4.3.3. Borrar Tagnames

Es posible borrar tagnames de Intouch bajo las siguientes condiciones:

1. El tag a borrar no puede estar utilizado.
2. WindowViewer debe estar cerrado
3. En WindowMaker no debe tener abierta ninguna ventana
4. Hay que ejecutar *SPECIAL ->UPDATE USE COUNTS*
5. Ejecutar *SPECIAL->DELETE UNUSED TAGS*

6. Volver a ejecutar *SPECIAL->UPDATE USE COUNTS*

### 2.2.5. Intouch QUICKSCRIPTS

#### 2.2.5.1. Tipos de Scripts

**InTouch** permite crear una lógica interna con condiciones, cálculos, etc. Esta lógica puede estar asociada a:

1. Toda una aplicación (*APPLICATION SCRIPTS*)
2. Una sola ventana (*WINDOW SCRIPTS*)
3. Una tecla (*KEY SCRIPTS*)
4. Una condición (*CONDITION SCRIPTS*)
5. Cambio de un dato (*DATA CHANGE SCRIPTS*)
6. Asociadas a un Active

## 7. Funciones Usuario (QUICKFUNCTIONS)

### 2.2.5.2. Funciones de Scripts

La lógica (*SCRIPT*) de **InTouch** es un programa que nos permitirá llevar a cabo acciones determinadas mediante una estructura *IF...THEN...ELSE*.

La lógica estará activa de acuerdo al tipo de lógica elegida (por aplicación, por ventana, etc.).

### 2.2.5.3. Uso de Variables Internas.

Es posible declarar variables internas mediante la instrucción *DIM*

DIM variablelocal [As tipo-dato]

Ej. DIM VarLoc1 As Integer;

VarLoc1 podrá utilizarse en todas las QuickScripts como variable de cálculo o de apoyo no podrá ser visualizada en pantalla.

Esta variable no ocupa tag

Si no especificamos [As tipo-dato], por defecto será declarada como integer (entero).

#### 2.2.6. Alarmas y Eventos

InTouch permite la visualización de alarmas distribuidas (gestión de las alarmas bajo una estructura cliente/servidor en una red de ordenadores). Nosotros daremos únicamente tratamiento a las alarmas locales.

InTouch soporta la visualización, archivo (en disco duro o en base de datos relacional) e impresión de alarmas tanto digitales como analógicas, y permite la notificación al operador de condiciones del sistema de dos modos distintos: Alarmas y Eventos.

Una alarma es un proceso anormal que puede ser perjudicial para el proceso y que normalmente requiere de algún tipo de actuación por parte

del operador. Un evento es un mensaje de estado normal del sistema que no requiere ningún tipo de respuesta por parte del operador.

#### 2.2.6.1. Tipos de Alarmas

Las alarmas pueden dividirse en los siguientes tipos:

<b>Condición de Alarma</b>	<b>Tipo</b>
Discrete	DISC
Deviation Major	LDEV
Deviation Minor	SDEV
Rate-Of-Change	(ROC) ROC
SPC	SPC
Value - LoLo	LOLO
Value - Lo	LO
Value - Hi	HI
Value - HiHi	HI

#### 2.2.6.2. Prioridades de las Alarmas

A cada alarma de cada tag puede asociarse un nivel de prioridad (importancia) de 1 a 999 (Prioridad 1 es más crítica). Ello permite filtrar alarmas en displays, en impresora o en disco duro.

#### 2.2.6.3. Grupos de Alarmas

Intouch dispone de un cómodo sistema para prioridades de alarmas.

Cuando creamos un tagname de alarma, le asignamos un grupo de alarmas. Estos grupos o "jerarquía" de alarmas permiten significar qué alarmas son más importantes, a la vez que permiten reconocer un grupo de alarmas en lugar de todas a la vez. Al crear un tag, lo asociamos a un grupo (si no lo hacemos, el tag de alarma queda automáticamente asociado al grupo principal, llamada *\$SYSTEM*).

#### 2.2.6.4. Cómo Crear Grupos de Alarmas

Los grupos de alarmas se crean desde /Special/Alarm Groups.



Desde aquí también "emparentamos" unos grupos con otros, hasta crear toda la "jerarquía"

#### 2.2.6.5. Definición de una Condición de Alarma en un Tagname.

Para definir un tagname debemos seleccionar ALARMS en el momento de definir el tagname.

Details  Alarms  Both  None

Para las alarmas discretas disponemos de las siguientes posibilidades:

Dictionary - Tag Name Definition  Details  Alarms  Both  None

New Restore Delete Save Previous Next Select Cancel Done

Tag Name:  Type: ... Memory Discrete

Group: ... \$System

Comment: AccessLevel

Log Data  Log Events  Retentive Value

Alarm State:  
 Off  On  None

Alarm State:  
 Off  On  None

*Off:* Define alarma cuando el registro está desactivado

*On:* Define alarma cuando el registro está activado

*Priority:* Aparece al definir on/off. Entre 1 y 999. Permite definir la prioridad de la alarma

Figura 2.17. Definición de una Condición de Alarma en un Tagname.

Para las alarmas analógicas, disponemos de las siguientes posibilidades (Ver figura 2.18):

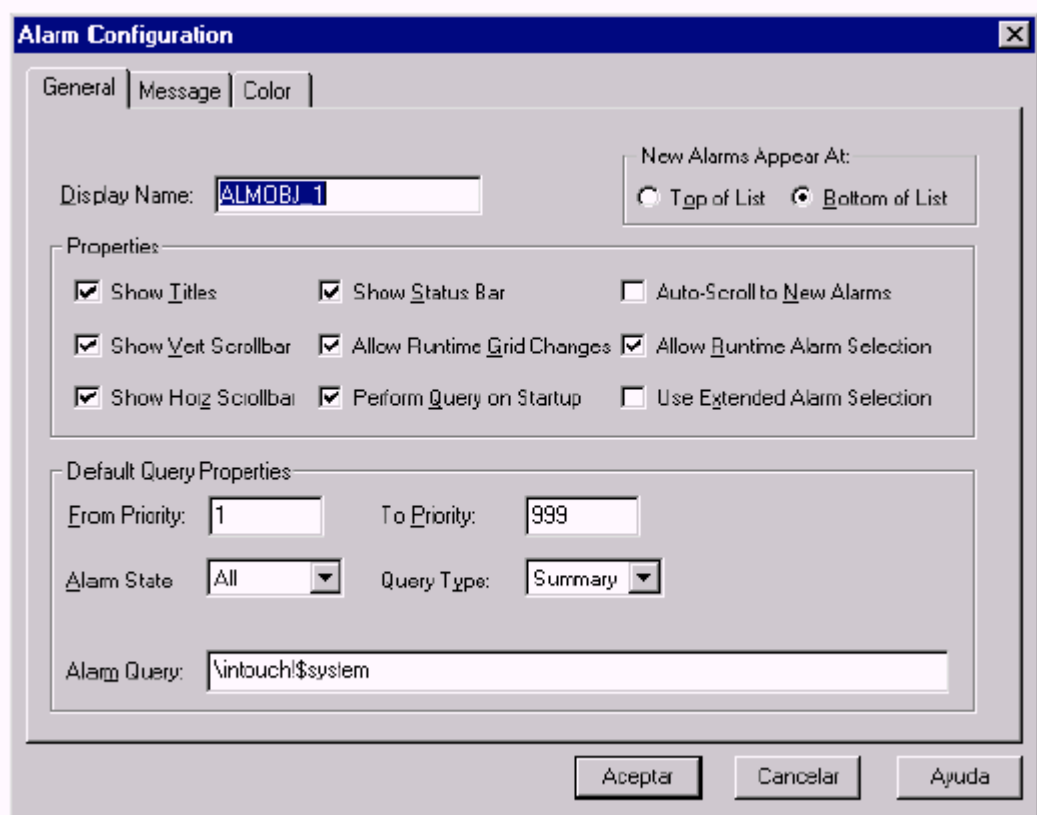
<i>Alarm Value</i>	Valores límite de la alarma. 4 niveles
<i>Pri</i>	Prioridad de la alarma
<i>Minor/Major Deviation</i>	Se utiliza para detectar cuándo el valor analógico es una desviación mayor o menor del valor especificado en <i>Target</i>

<i>% Deviation</i>	Porcentaje de desviación permitido al tagname con respecto al <i>Target</i> para que se produzca una alarma por desviación mayor o menor
<i>Target</i>	Valor de referencia para los porcentajes mayor/menor de desviación
<i>Rate of Change</i>	Este tipo de alarma detecta cuándo el valor de la alarma varía en exceso de acuerdo a una cantidad de tiempo

Figura 2.18. Definición de una Condición de Alarma Analógica

### 2.2.6.6. Configuración de un Objeto de Alarmas

La Figura 2.19.muestra la Configuración de un Objeto de Alarmas



*Figura 2.19. Configuración de un Objeto de Alarmas*

La Tabla 2.6. describe los Objetos presentes en la Figura 2.19.

<i>Display Name</i>	Nombre unívoco del display que estamos parametrizando
<i>New Alarms Appear At</i>	Alarmas nuevas al principio del display (Top), o al final (Bottom)
<i>Show Titles</i>	Muestra o no títulos
<i>Show Vert Scroll Bar</i>	Muestra o no la barra de scroll vertical
<i>Show Horz Scroll Bar</i>	Muestra o no la barra de scroll horizontal
<i>Show Status Bar</i>	Muestra o no la barra de estado
<i>Allow Runtime Grid Changes</i>	Permite o no cambios en la grilla de visualización durante el runtime
<i>Perform Query on Startup</i>	Ejecuta consulta sobre alarmas al arrancar
<i>Auto Scroll to New Alarms</i>	Se posiciona automáticamente en la nueva alarma
<i>Allow Runtime Alarm Selection</i>	Permite selección de alarmas en runtime
<i>Use Extended Alarm Selection</i>	Usa selección extendida de alarmas
<i>From ... To Priority</i>	Niveles de prioridad de alarma mayor y menor que se displayarán en el objeto de alarmas
<i>Alarm State</i>	Define si queremos ver todas las alarmas (all) o sólo las no reconocidas (unack)
<i>Query Type</i>	Seleccione <i>Alarm Summary</i> para displayar las alarmas activas actualmente. Seleccione <i>Alarm History</i> para displayar las alarmas activas y también aquellas que han dejado de serlo.
<i>Alarm Query</i>	Tipo de query. Por defecto es \InTouch!\$system (alarmas del propio nodo, del grupo \$system).- Es posible cambiar el nodo del que llegan las alarmas (\nodo1\InTouch!\$system), o el grupo (\InTouch!grupo1)

Tabla 2.8. Objetos presentes en un Objeto de Alarma

#### 2.2.6.7. Configuración de Alarmas / Eventos.

Se dispone de tres menús para la configuración de las alarmas. A ellos se accede desde el menú *Special ÆConfigure*

- *GENERAL* Configuración de parámetros de las alarmas en pantalla

- *ALARM LOGGING* Configuración del fichero de alarmas
- *ALARM PRINTING* Configuración de la impresión de alarmas

### *Creación de Condiciones de Reconocimiento*

Se puede crear pulsadores de reconocimiento utilizando el campo **.Ack** en un pulsador.

Es también posible reconocimiento por grupo, por tag seleccionado, por display de alarmas, etc.

#### 2.2.6.8. Campos de las Alarmas

Los Campos de las Alarmas disponibles se detallan en la tabla 2.7.

.Ack	Campo discreto de lectura/escritura para reconocimiento de alarmas. P <b>Tagname.Ack=1;</b> <b>.AckNombreGrupo;</b> <b>AckVariableGrupo;</b>
.Alarm	Campo discreto de lectura que se activa cuando se activa

	el tagname o grupo <b>GrupoAlarma.Alarm;</b>
.AlarmClass	Devuelve la clase de la alarma
.AlarmComment	Devuelve el comentario de la alarma
.AlarmDate	Devuelve la fecha de la alarma
.AlarmDevDeadband	Campo analógico de lectura/escritura que controla el porcentaje de desviación mayor y menor de las alarmas. Por ejemplo, para cambiar el porcentaje de desviación al 25%, utilizaremos: <b>Tagname.AlarmDevDeadband=25;</b>
.AlarmEnable	Discreto de lectura/escritura, que activa/desactiva eventos y alarmas de un tagname o grupo. P.e.j. <b>GroupVariable.AlarmEnable=1;</b>
.AlarmGroupSel	Devuelve el grupo al que pertenece a alarma
.AlarmLimit	Devuelve el límite de la alarma
.AlarmName	Devuelve el nombre de la alarma
.AlarmOprNode	Devuelve el nombre del operador del nodo de la alarma
.AlarmOprName	Devuelve el nombre del operador de la alarma
.AlarmPri	Devuelve la prioridad de la alarma
.AlarmProv	Devuelve el proveedor de la alarma
.AlarmState	Devuelve el estado de la alarma
.AlarmTime	Devuelve la hora de la alarma
.AlarmType	Devuelve el tipo de la alarma
.AlarmValDeadband	Analógico de lectura/escritura que controla el deadband de una alarma o grupo. P.ej. <b>Tagname.AlarmValDeadband=10;</b>
.AlarmValue	Devuelve el valor de la alarma

.DevTarget	Campo analógico de lectura/escritura que controla la desviación mayor/menor. P.ej. <b>Tagname.DevTarget=500;</b>

*Tabla 2.9. Campos de Alarmas*

### 2.2.7. Curvas Históricas y Reales

El software **InTouch** permite desplegar curvas y tendencias en pantalla tanto en tiempo real como de valores históricos.

#### 2.2.7.1. Curvas en Tiempo Real

Disponemos de una herramienta en la toolbox que permite crear una curva en tiempo real.

##### 2.2.7.1.1. Creación de una Curva en Tiempo Real

Disponemos de una herramienta en la *toolbox* que permite crear una curva en tiempo real.

#### 2.2.7.1.2. Configuración de la Curva en Tiempo Real.

Dentro de una curva en tiempo real, podemos definir una serie de parámetros, tales como tiempo de actualización de la curva, tamaño de la muestra, colores de los lápices, color del fondo y otros.

Las curvas en tiempo real no almacenan el valor desplegado, sino que simplemente representan un valor o registro real del PLC (o interno de **InTouch**) y lo representan en pantalla en forma de curva.

**Real Time Trend Configuration**

Comment:

**Time**  
Time Span:   
 Sec  Min  Hr

**Sample**  
Interval:   
 Msec  Sec  Min  Hr

**Color**  
Chart Color:   
Border Color:

**Time Divisions**  
Number of Major Div:    
Minor Div/Major Div:    
 Top Labels  Bottom Labels  
Major Div/Time Label:    
HH:MM:SS Display:  HH  MM  SS

**Value Divisions**  
Number of Major Div:    
Minor Div/Major Div:    
 Left Labels  Right Labels  
Major Div/Value Label:    
Min Value:  Max:

Per:	Expression	Color:	Width:
1	<input type="text"/>	<input type="color" value="green"/>	<input type="text" value="1"/>
2	<input type="text"/>	<input type="color" value="orange"/>	<input type="text" value="1"/>
3	<input type="text"/>	<input type="color" value="red"/>	<input type="text" value="1"/>



*Figura 2.20. Configuración de la Curva en Tiempo Real.*

#### 2.2.7.2. Curvas Históricas

***InTouch permite la visualización de históricos distribuidos (gestión de los históricos bajo una estructura cliente/servidor en una red de ordenadores).***

Las curvas históricas permiten visualizar la evolución con respecto al tiempo de un dato en forma de curva o tendencia.

Este dato debe haber sido almacenado previamente, por lo que el tagname visualizado en este tipo de curvas debe haber sido previamente definido como del tipo coged

#### 2.2.7.2.1. Configurar Historical Logging

Para activar el gestor de históricos de **InTouch**, es necesario acudir a SPECIAL -> CONFIGURE  
-> HISTORICAL LOGGING

#### 2.2.7.2.2. Utilización de Wizard en Curva Histórica

Dentro de los WIZARDS disponemos de la herramienta *CURVA HISTORICA*. Esta herramienta es muy completa, ya que nos permite disponer de la mayoría de elementos de una curva sin necesidad de programación.

#### 2.2.8. Comunicaciones I/O SERVERS

*DDE* es un protocolo de comunicaciones para intercambio de datos entre aplicaciones *WINDOWS*, de muy sencillo uso, en el que la comunicación se establece automáticamente entre programas que contemplan la estructura *DDE* (cliente-servidor).

Un programa que puede mandar datos al bus *DDE* es un programa servidor.

Un programa cliente puede recibir datos *DDE*. Ello permite que podamos muy fácilmente crearnos programas con gestiones especiales en *VBASIC*, *EXCEL*, etc., y pasar los datos a **InTouch** sin necesidad de crear un programa de comunicaciones. Así ocurre con los servidores de autómatas que dispone **WONDERWARE**, entre los que se incluyen la práctica totalidad de los PLCs más conocidos del mundo con comunicación tanto punto a punto como en red. Por supuesto, podemos direccionar un servidor *DDE* a un puerto de comunicaciones y otro servidor a otro puerto, con lo que podemos compartir información.

La comunicación *DDE* se basa en una convención con estos tres parámetros:

- Aplicación

- Tópico

- Elemento

Cuando queremos desde otra aplicación (p.e. Excel), enlazar vía DDE un dato InTouch, los parámetros son:

- APLICACIÓN VIEW

- TOPICO TAGNAME

- ELEMENTO {nombre del tag que queremos enlazar}

Para poder enlazar datos vía DDE de otras aplicaciones Windows a **InTouch**, debemos crear un *Access Name*. A cada *Access Name* podemos asociar una Aplicación y un Tópico.

## **CAPITULO III**

# **DISEÑO Y DESARROLLO DEL PROTOTIPO Y DEL SISTEMA**

## ***3.1 Diseño y Desarrollo del Prototipo***

### 3.1.1. Diseño del Prototipo

Para el correcto Diseño del Prototipo se requiere tanto de un cierto proceso para su puesta en marcha, como del correcto funcionamiento de cada una de sus partes.

El elemento principal del Prototipo es un controlador lógico programable, el mismo que adquiere, procesa los datos, y determina las acciones a tomar de acuerdo a las necesidades y requerimientos del usuario.

Primeramente tenemos que determinar los procesos que vamos a automatizar, así como establecer las entradas y salidas disponibles del PLC y la tarea que realizará cada una de ellas.

#### 3.1.1.1. Procesos a Automatizar

Tenemos que definir los procesos a automatizar y la demostración de las siguientes funciones domóticas:

➤ CONTROL DE ALARMAS:

◦ Detección y aviso de incendios.

◦ Detección y aviso de intrusos.

➤ CONTROL DE ACCESOS: Control del Estado de los Accesos, puerta y ventana.

➤ CONTROL DE ILUMINACIÓN: Control de Iluminación por tiempo y por luminosidad de las luces de la vivienda.

➤ CALEFACCIÓN: Control de la temperatura en una de las habitaciones, manteniéndola en un rango definido, mediante un calefactor y un ventilador.

- **SIMULACIÓN DE PRESENCIA:** Medio eficaz para evitar robos y otras agresiones a la propiedad privada. Se utilizan varias luces, riego de jardín y una persiana para realizar dicha función.
  
- **CONTROL DE APERTURA Y CIERRE DE PERSIANAS:** Se ha automatizado la apertura y cierre de una de las ventanas.
  
- **RIEGO DEL JARDÍN:** Control automático.
  
- **CONTROL DE NIVEL DEL TANQUE RESERVORIO:** Control de tanque de nivel bajo y alto del agua.



### 3.1.1.2. Entradas y Salidas

El prototipo dispondrá de 14 entradas:

- 2 entradas para encender/apagar las luces de la casa.
- 1 entrada para determinar el estado encendido/apagado de la luz exterior.
- 1 entrada para activar alarma.
- 1 interruptor para subir/bajar la persiana.
- 2 entradas para nivel alto/bajo de temperatura.
- 2 entradas para control de nivel de líquido de tanque (alto/bajo).
- 1 entrada para detección de incendio

- 1 entrada para detección de presencia exterior.
- 2 entradas para ventana y puerta abierta/cerrada.
  
- 1 entrada para el control manual/automático del sistema

Dispone de 9 salidas para el control de:

- 2 luces.
  
- 2 persiana.
  
- 2 motores para control del nivel del tanque.
  
- 1 calentador.
  
- 1 ventilador.
  
- 1 alarma.

### 3.1.1.3. Diseño de los Circuitos

En el diseño de los circuitos tuvimos que determinar la forma en que el PLC activará cada uno de los circuitos así como la acción que realizará cada uno de acuerdo a nuestros requerimiento, los mismos que se diseñarán en una tarjeta principal (Anexo 3.1).

Para el Diseño de esta tarjeta utilizamos el Software TANGO PCB, la misma que contiene los siguientes circuitos:

1. Circuito para el Control de Temperatura
2. Circuito para el Control del Nivel del Tanque
3. Circuito para el Control de Presencia Exterior
4. Circuito para el Control de Giro del Motor

5. Circuito para la Detección de Humo

6. Circuito para Iluminación Exterior

7. Circuito para Dispositivos que se alimentan con 110 V.

### 3.1.2. Desarrollo del Prototipo

#### 3.1.2.1. Construcción del Prototipo

La construcción de nuestro prototipo lo hemos realizado en base a un plano de una casa a escala 1:12. Ver Anexo 3.2.

La casa consta de 6 habitaciones, que son las siguientes: Sala-Comedor, Cocina, 2 Baños, 2 Dormitorios, así como una zona ajardinada en el exterior donde se encuentra un tanque reservorio.

Una vez definidos los elementos a automatizar y en paralelo con la construcción de la maqueta, se realizará el monitoreo y la ejecución de los dispositivos electrónicos utilizados a través del programa generado con el software STEP 7 MICROWIN V. 3.01. el mismo que enlazaremos con el software INTOUCH 7.1 con el cual desarrollaremos las interfaces amigables para interacción con el usuario.

#### 3.1.2.1.1. Conexiones Eléctricas del Tablero de Control

Todas las conexiones eléctricas tanto de alimentación, entradas y salidas del PLC se las realiza en el Panel en el que está montado el mismo. Para la disposición de los cables se usan regletas plásticas dispuestas en la parte inferior de la maqueta y el cableado se lo realiza con cable #18 AWG.

### 3.1.2.1.2. Construcción de la Tarjeta Electrónica

Luego haber diseñado la tarjeta procedemos a montar sus elementos, (Ver Anexo 3.3) así:

➤ Detección del Nivel de Líquido del Tanque Reservoirio, para esto utilizamos los siguientes elementos:

- 2 Relés de 12 V.
- 2 Transistores
- Resistencias

El circuito detecta los dos niveles del líquido (agua) del tanque reservoirio (alto/bajo), estas dos señales son enviadas a las entradas del PLC cada una de las cuales se activa en el momento que el agua cierra un contacto ya sea alto o bajo, pues el agua es un conductor, para posteriormente

realizar las tareas automáticas de acuerdo al nivel detectado.

➤ Tarjeta para el Control de Giro del Motor, para el diseño de esta tarjeta utilizamos los siguientes elementos:

- 2 Relés de 12 V.
- 2 Transistores PNP.
- 2 Transistores NPN.
- 1 Motor de 5 V.
- Tarjeta universal.

Esta tarjeta se encarga de realizar la inversión de giro de un motor de corriente continua, utilizando un circuito H, para lo cual se utilizó dos salidas del PLC, cada una de las cuales se activa de acuerdo al giro del motor.

➤ Circuito para el Control de Temperatura, para el diseño de esta tarjeta utilizamos los siguientes elementos:

- 1 sensor de Temperatura LM35.
- 1 Conversor Análogo/Digital AC 3161.
- 1 Decodificador a 7 segmentos AC 3162.
- 1 Comparador de Tensión LM 3914
- 2 Potenciómetros
- 2 Reles de 12 V.

Este circuito se encarga de realizar un control automático de temperatura, mediante la comparación de tensión con el



LM3914 y de acuerdo a nuestros requerimientos tomamos la tensión que necesitamos para las temperaturas de 20°C y 30°C, siendo un mv igual a 1° para el LM35, estas señales pasan al decodificador AD CA3161 y de aquí al decodificador a 7 segmentos, para poder visualizar la temperatura en los respectivos displays, este circuito se conecta a dos entradas del PLC.

Si el circuito detecta una temperatura ambiental de 20° se activa una entrada determinada del PLC, la cual activará la salida conectada al calefactor, hasta que este llegue a la señal alta, de esta manera se apaga el calentador y se prende el ventilador, realizándose el mismo proceso cuando llega a la señal baja es decir 30°C.

De esta manera mantendrá la temperatura entre 20° y 30° en forma automática.

➤ Circuito para Detección de Intrusos en el Exterior, para el diseño de esta tarjeta utilizamos los siguientes elementos:

- 1 Fotoresistencia.
- 1 Relé de 12 V
- 1 Puntero Láser
- Resistencias
- Tarjetas universales.

Este circuito se encarga de detección de intrusos en el exterior, mediante la utilización de una fotorresistencia, la cual esta apuntada con un láser, el momento que no incide luz del puntero sobre la fotorresistencia, el circuito envía una señal a la entrada del PLC activándola, para

posteriormente tomar las medidas pertinentes.

➤ Tarjetas para Adaptación de Dispositivos de 110 V., para el diseño de esta tarjeta utilizamos los siguientes elementos:

- Relés de 12V.
- Tarjetas universales.

Existe dispositivos en nuestro prototipo que necesitan alimentación de 110V, para lo cual fue necesario realizar la adaptación de los mismos para que trabajen con el PLC utilizando relés para enviar pulsos y así activar a los dispositivos.

➤ Circuito para Detección de Humo, para el diseño de este circuito utilizamos los siguientes elementos:

- 1 Sensor de humo
  
- 1 Relé.
  
- 1 Amplificador Operacional LM324.
  
- 1 Capacitor.
  
- 1 Transistor PNP

Para el diseño del detector del Humo y para adaptarlo al PLC, utilizamos el Amplificador Operacional LM324, ya que la señal que tomamos del Sensor de Humo es de muy bajo voltaje y su tiempo de activación es demasiado corto, por lo que amplificamos esta señal con el LM324 y también a un capacitor para poder activar el rele y así enviar esta señal a una entrada del PLC y posteriormente trabajar con ella.

### 3.1.2.2. Programación del PLC Simatic S7-200

Previo a la puesta en marcha del sistema se debe descargar el programa correspondiente hacia el PLC, los pasos para este efecto se debe seguir la secuencia que se indica a continuación:

- Iniciar el programa STEP 7 MICROWIN V. 3.01.
  
- Conectar el PLC a la fuente de alimentación (110V/60Hz).
  
- Conectar el cable de comunicación desde el puerto serial PC a la terminal de comunicación del PLC (Figura 3.1.).

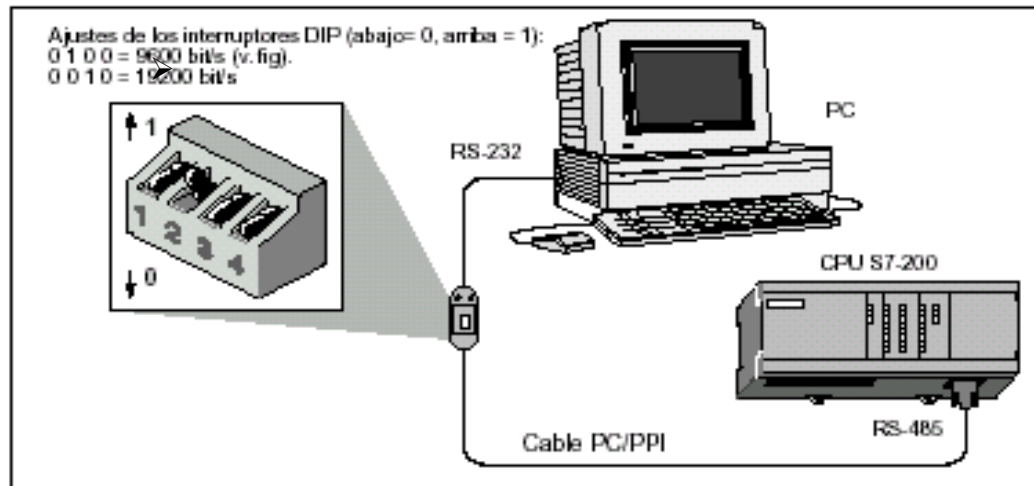


Figura 3.1. Diagrama de conexión del Cable de Comunicación

Los terminales de comunicación RS232 en el PC y el RS485 en el PLC, y la configuración del cable de comunicación se muestra a continuación en la Figura 3.2.

## AJUSTES DE LOS INTERRUPTORES

**0 0 1 0 = 19200 bit/s**

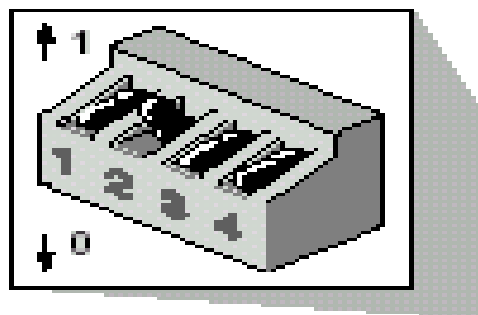
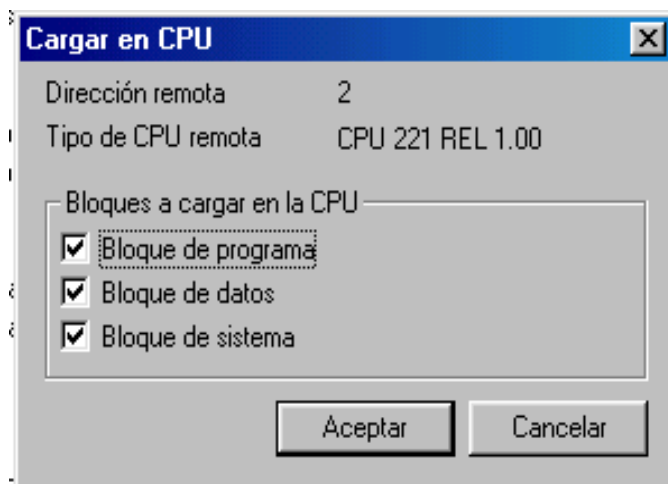


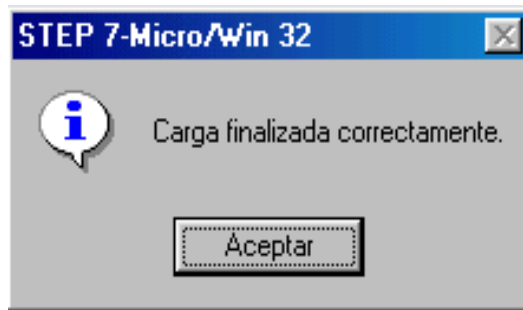
Figura 3.2. Terminales de Comunicación

- Abrir el programa ubicado en el directorio C:\ARCHIVOS DE PROGRAMA\STEP7\PROJECTS\SISTEMA DOMOTICO.PRJ
- Descargar el programa al PLC desde la opción de la barra de menú CARGAR EN CPU. Ver figura 3.3.



*Figura 3.3. Bloques a Cargar en la CPU*

- Una vez finalizada la carga del programa en el CPU del PLC presenta una pantalla de carga exitosa en caso de no existir errores. Ver figura 3.4.



*Figura 3.4. Carga con éxito del programa en el PLC*

#### 3.1.2.2.1. Desarrollo del Programa en el PLC

La Interfaz del Software, necesita para su funcionamiento el desarrollo de un programa en lenguaje STEP 7 con el editor KOP, el mismo que corre en el PLC. Este programa se encarga de controlar la comunicación serial con el computador y realiza la adquisición y filtrado de las señales discretas y analógicas de las tarjetas, trata las entradas digitales, tanto las enviadas por Intouch como las del PLC, lee el reloj de tiempo real, controla los motores para el control del nivel del líquido, realiza la activación de las señales recibidas por tiempos.



Las órdenes son enviadas por el usuario a través del puerto serie al PLC, las cuales las recoge y las deposita en variables intermedias, que son consultadas por el programa de mando, actuando de acuerdo a la lógica de programación almacenada. De esta manera dotamos al sistema de una flexibilidad que el autómata por si solo es incapaz de ofrecer.

El PLC, envía por el puerto serie el estado de todas las variables que intervienen en el proceso hacia el programa del PC, realizado en Intouch, una vez recibidas se depositan en las marcas correspondientes. Los datos manipulados por el PLC son procesados por varias subrutinas.

El programa en STEP 7, esta estructurado de la siguiente manera:

1. Programa Principal

El Programa Principal se encarga de llamar a todas las Subrutinas realizadas.

## 2. Subrutinas

### Subrutina 0

Leemos el reloj del plc y cargamos los minutos y segundos en las variables tipo byte (vb4 y vb5) y los transformamos a entero (vw10 y vw800) para su posterior utilización

### Subrutina 1

Cargamos los valores de minutos inicial y final (vw600,wv650)para la iluminación del dormitorio que viene desde Intocuh y los transformamos a bcd (vw700,wv900),para su posterior comparación.

## Subrutina 2

Cargamos los valores de minutos inicial y final (vw30, vw40) para la iluminación de la sala que viene desde Intocuh y los transformamos a bcd (vw100,vw200) para su posterior comparación

## Subrutina 3

Cargamos los valores de segundo inicial y segundo final (vw600, vw650) para el riego del jardín que viene desde intouch y los transformamos a bcd(vw700, vw900) para su posterior comparación

## Subrutina 4

Cargamos los valores de minuto inicial y minuto final (vw480, vw500)para el control

de persiana que viene desde intouch y los transformamos a bcd (vw490,vw510)para su posterior comparación

#### Subrutina5

Ponemos a 0 las salidas Q06 y Q07 en el primer ciclo, perteneciente al control de las persianas

#### Subrutina 6

Activamos el foco en forma manual y automática por medio de valores de acuerdo al estado de la entrada M13.4 y de acuerdo al valor enviado desde Intouch en la marca M20.1, Q2.0 se activará en forma manual cuando M13.4 esté activo y en forma automática cuando por M13.4 no circule voltaje y se activará en un tiempo

comprendido entre el minuto inicial y final  
enviado desde Intouch

#### Subrutina 7

Activamos el foco en forma manual y automática por medio de valores de acuerdo al estado de la entrada M13.4 y de acuerdo al valor enviado desde Intouch en la marca M20.2, Q0.5 se activará en forma manual cuando M13.4 esté activo y en forma automática cuando por M13.4 no circule voltaje y se activará en un tiempo comprendido entre el minuto inicial y final enviado desde Intouch

#### Subrutina 8

En esta Subrutina enclavamos la salida Q2.3 que es el motor de la cisterna cuando la

entrada I1.4 este inactiva. Enclavamos la salida Q2.3 cuando el nivel del tanque sea alto I1.3, y al mismo tiempo activamos el motor del riego de jardín mediante un temporizador.

Cuando M13.4 este inactivo activamos el riego automático del jardín mediante los valores enviados desde Intouch.

#### Subrutina 9

Cuando la entrada I0.2 está inactiva enclavamos la salida Q2.6 a 1 que es el calefactor, al mismo tiempo desactivamos la salida Q2.4 a 1, cuando la entrada I0.3 enclavamos la salida Q2.4 que es el ventilador y al mismo tiempo desactivamos la salida Q2.6

#### Subrutina 10

Cuando la marca M13.4 está activa, activamos el movimiento a la derecha (Q0.6) o a la izquierda (Q0.7) de la persiana de acuerdo a los temporizadores T37 y T45, cuando M13.4 este inactiva la persiana se abrirá o cerrará en forma automática de acuerdo a los minutos inicial y final enviados desde Intouch.

#### Subrutina 11

Cuando la entrada I0.4 está activa y las entrada I1.1 o I1.2 estén inactivas se activará la salida Q2.5 con un retardo al temporizador T51

#### Subrutina 12

Si la entrada I0.6 (Detector de Humo) activamos la marca M14.3 que será enviada a Intouch

### Subrutina 13

Si la entrada I1.0 (Detector de presencia exterior) activamos la marca M12.2 que será enviada a Intouch

#### 3.1.2.3. Conexión de los Dispositivos con Plc Simatic S7-200

Los Dispositivos que utilizaremos en nuestra tesis serán conectados a las salidas del PLC, el mismo que se alimenta con una fuente de 24 V, los cuales recibirán un pulso del PLC para su correspondiente funcionamiento, la acción (Ver figura 3.5.).





*Figura 3.5. Conexión de Sensores con el PLC*

3.1.2.3.1. Comunicación PLC – INTOUCH

Para el acceso de datos en el PLC se utiliza el driver de comunicación S7200PPI de SIEMENS, el mismo que es un programa de aplicación que trabaja bajo WINDOWS y actúa como un servidor de protocolo de comunicación, que permite el acceso al PLC desde otra aplicación de WINDOWS.

En este caso permite enviar o recibir datos desde el programa de aplicación INTOUCH hacia el PLC ( SIMATIC S7-200). La comunicación interna entre INTOUCH Y S7200PPI se establece por medio del protocolo DDE (Dynamic Data Exchange), protocolo

desarrollado por Microsoft para permitir a aplicaciones que trabajan en ambiente de Windows enviar y recibir datos de una a otra.

El direccionamiento de datos por medio de S7200PPI se lo realiza mediante un convenio que requiere la definición de tres partes denominadas *Application Name*, *Topic Name* y *Item Name* (Tabla3.1).

<i>Application Name</i>	Es el nombre del programa que trabaja bajo Windows y tendrá acceso al PLC, en caso de que los datos sean leídos o escritos en un PLC SIMATIC S7-200, por medio de este driver de comunicación el <i>application name</i> deberá ser S7200PPI.
<i>Topic Name</i>	Es el nombre genérico que se le da a la aplicación para ser identificada en el proceso de comunicación, por ejemplo: PLC.  En el caso de que la comunicación sea con INTOUCH es el mismo nombre que lleva el <i>access name</i> de los I/O <i>tags</i> de la aplicación.
<i>Item Name</i>	Son los nombres de específicos de datos dentro del topic (PLC), por ejemplo un ítem puede ser un contador, un relé, un registro, etc. dentro del PLC.

	Los <i>items names</i> dependen del modo de direccionamiento que utilice la programación del PLC utilizado.
--	---

*Tabla 3.1. Partes del Direccionamiento de Datos*

### 3.1.2.3.2. Configuración del DRIVER de Comunicación

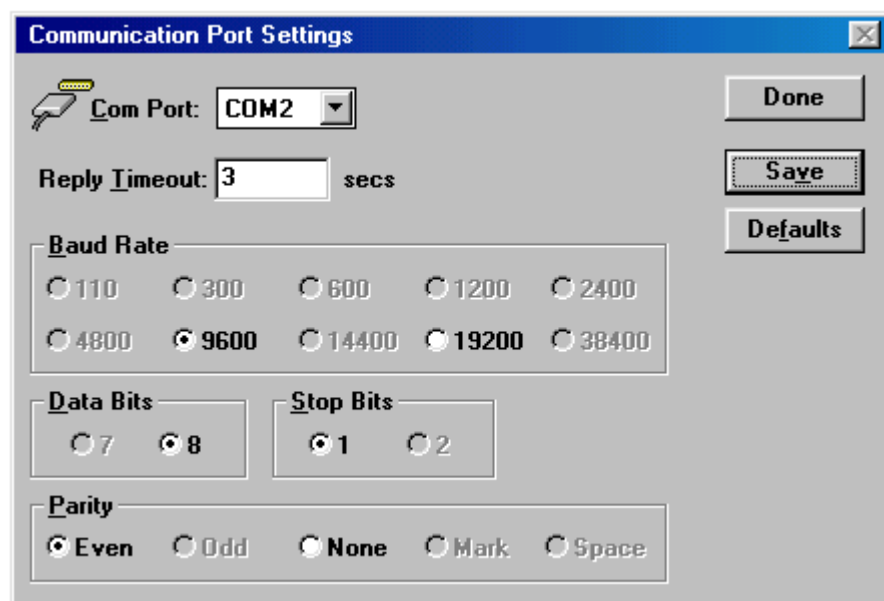
Una vez que el software S7200PPI es instalado en el computador, este requiere una configuración para su correcto funcionamiento. Las opciones de configuración se encuentran en la barra de menú de la ventana de programa en la opción *configure*. Ver figura 3.6.



Figura 3.6. Barra de Menú. Opción Configure

### 3.1.2.3.3. Configuración del Puerto de Comunicaciones

La Figura 3.7. muestra la ventana de configuración del puerto de comunicaciones que es utilizado para la comunicación con el PLC, su acceso se lo realiza desde la barra de menú *configure/com port settings*

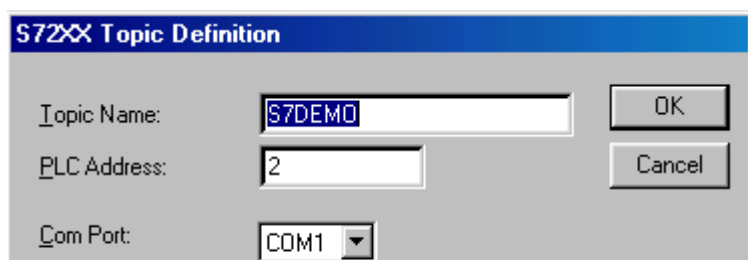


### Figura 3.7. Ventana de configuración del puerto de comunicaciones

*Esta ventana nos permite configurar el puerto de comunicaciones que será utilizado, el modo de transmisión de datos, velocidad, bits de paridad, datos, parada, modo y chequeo de errores de ser el caso.*

#### 3.1.2.3.3.1. Configuración del Topic

Para la configuración del *topic* se debe acceder a la ventana correspondiente desde el menú *Configure/Topic Definition*, en la cual se debe agregar un nuevo *topic* (*New*), como se muestra en la Figura 3.8.



The image shows a software dialog box titled "S7200 Topic Definition". It has a blue header bar. Below the header, there are three input fields: "Topic Name" containing "S7DEMO", "PLC Address" containing "2", and "Com Port" with a dropdown menu set to "COM1". To the right of these fields are two buttons: "OK" and "Cancel".

*Figura 3.8. Ventana de Configuración de Nuevo Topic*

En esta ventana se debe configurar  
y tomar en cuenta lo siguiente  
(Tabla 3.2):

<b>TOPIC NAME</b>	<b>NOMBRE ASIGNADO AL TOPIC.</b>
PLC Ardes	Dirección del PLC
Com Port	Puerto de comunicaciones a utilizarse
Max. Message Size	Es un campo dinámico que muestra un ejemplo del tipo de conexión que debe ser realizada de acuerdo a la opción escogida en el campo <i>Conect Type</i> .
Update Interval	Permite determinar el tipo de direccionamiento de la conexión entre el PLC y el computador, en

	este caso se utilizó la opción <i>direct</i> , que permite una conexión directa vía entre el PC y el PLC.
--	---

*Tabla 3.2. Configuración del Topic*

Finalmente se debe activar el puerto y el tópico desde *Configure/Communication Ports Settings* en la barra de menú inicio del programa.

## ***3.2 Diseño y Desarrollo del Sistema***

### **3.2.1. Visión General del Sistema**

Las computadoras han llegado a ser tan útiles en el campo de la automatización de tareas tediosas o peligrosas que realiza el ser humano por lo que brindan seguridad, confort, y optimización de tiempo.

Para la realización de esto las computadoras se han unido con la electrónica para que esta automatización llegue a dar resultados satisfactorios, así podemos automatizar procesos industriales en

fábricas, empresas, así como también en el mismo hogar dando seguridad a los nuestros y haciendo un poco más fácil la vida.

**Este trabajo nació, en primer lugar,  
con la finalidad de realizar una  
herramienta de seguridad y confort  
para el hogar. Con el propósito de  
evitar el robo simulando presencia en  
viviendas deshabitadas o en estancias  
más largas y que permita monitorizar  
los dispositivos de una vivienda desde  
un PC permitiendo informar en todo  
momento de los eventos que sucedan en  
la vivienda y poder actuar en  
consecuencia.**



### 3.2.2. Objetivos del Sistema

- El Objetivo principal del presente sistema es el diseño y construcción de un Sistema Domótico con el Software InTouch que permita supervisar y controlar la seguridad, la gestión de energía, el confort de una casa a través de un autómata programable en tiempo real.
- Desarrollar un sistema que de la posibilidad de informar y monitorizar en todo momento lo que sucede en la vivienda en cuanto al control de los dispositivos eléctricos ya sea que estos sean manejados por señales analógicas o digitales.
- Desarrollar un documento informativo sobre los principales sensores y actuadores más utilizados en domótica.
- Tener una visualización esquemática del proceso.
- Guardar información y datos de las variables leídas del proceso.

- Tener alarmas sonoras y visuales cuando suceda una condición anormal de funcionamiento o que una variable se encuentre fuera de rango.
  
- Visualizar las variables análogas y digitales del proceso.
  
- Poder configurar, calibrar, setear y guardar los parámetros de proceso.

### 3.2.3. Alcance del Sistema

En la actualidad, con el gran avance de la computación y por la creciente difusión de aplicaciones de la electrónica y la fantástica disminución del precio de los componentes, el nacimiento y el desarrollo de los microprocesadores y, sobretodo, la miniaturización de los circuitos de memoria nos permiten presagiar una magnífica introducción de este tipo de sistemas en una inmensa gama de nuevos campos de aplicación.

Además estos sistemas satisfacen las exigencias tanto de procesos continuos como discontinuos. Regula presiones, temperaturas, niveles

y caudales así como todas las funciones asociadas de temporización, cadencia, conteo y lógica, también puede permitir programar y realizar tareas dentro de un entorno industrial, maquinas o procesos lógicos secuenciales. Es decir nos permite un amplio alcance en muchos campos tecnológicos.

#### 3.2.4. Requerimientos hardware y software

Para ejecutar la versión 7.1 de Intouch se recomienda el siguiente hardware y software:

- Procesador Pentium 100 Mhz o Superior
- 100 MB mínimo libres en disco duro
- 32 MB de RAM mínimo
- Controlador de Video SVGA
- Artículo de punteo (mouse, touch screen, etc.)
- Sistema Operativo: Microsoft Windows 95 (o superior) o Windows NT 4.0 SP4 (o superior).

Una vez que el software es instalado en el computador un archivo .EXE es agregado a la barra del menú inicio (inicio/programas/wonderware factory suite/intouch.exe), el mismo que permite iniciar el programa abriendo primeramente *Intouch-Application Manager*.

#### 3.2.5. Descripción del Sistema

Como se indicó anteriormente el monitoreo y control del proceso se lo puede realizar desde un computador comunicado con el PLC por medio de una interfaz gráfica. Para dicho propósito es necesario determinar el software de desarrollo del HMI, así como los protocolos de comunicación correspondientes.

#### 3.2.6. Análisis del Sistema

Primeramente antes del diseño del mismo tenemos que realizar un minucioso análisis de todos los requerimientos necesarios para el desarrollo, funcionamiento y las tareas que realizaremos en el presente

sistema para lo cual debemos tomar en cuenta aspectos tanto económicos y sociales, así como se debe adoptar una solución para implementar el algoritmo de control de determinada aplicación, se plantea ahora la necesidad de seleccionar, de entre la amplia oferta del mercado, el equipo más adecuado.

Como en otros casos, la decisión debe basarse en análisis sistemático de una serie de factores, pero considerando no solo las características actuales de la tarea de control, sino también las necesidades futuras en función de los objetivos que deseamos cumplir.

Para lo cual se debe tomar en cuenta varios factores, como los siguientes:

#### *Factores cuantitativos*

Se refieren a la capacidad del equipo para soportar todas aquellas especificadas para el sistema de control y se pueden agrupar en las siguientes categorías:

- Entradas/ Salidas ( E/S ): cantidad, tipo, prestaciones, ubicación, etc.
  
- Tipo de control: control de una o varias máquinas, proceso, etc.
  
- Memoria: cantidad, tecnología, expandibilidad, etc.
  
- Software: conjunto de instrucciones, módulos de programa, etc.
  
- Periféricos: equipos de programación, dialogo hombre – maquina, etc.
  
- Físicos y ambientales: características constructivas, temperatura.

*Factores cualitativos.*

Una vez evaluados los factores correspondientes a las características técnicas y constructivas de los componentes y equipos periféricos, de debe tomar en cuenta el numero de equipos posibles para una determinada aplicación.

En muchas ocasiones la decisión se basa en criterios comerciales y en general limitados al aspecto económico de la adquisición, pero hay que tener en cuenta otros aspectos que en definitiva tendrán una mayor influencia a medio plazo. Es el momento de evaluar factores menos tangibles que se ocultan en las mismas características del equipo y en las del fabricante o el suministrador.

Por ultimo hay que considerar las disponibilidad del producto y de recambios. La disponibilidad de recambios en un tiempo mínimo es fundamental ya que, aunque deben existir recambios en el almacén de mantenimiento, no es posible ni rentable cubrir la totalidad de componentes.

### 3.2.7. Diseño de la Aplicación

Para el presente proyecto y con la finalidad de que el operador tenga un ambiente amigable y este posibilitado de controlar y visualizar integramente los procesos de control en el momento de operar el HMI, se crearon ventanas animadas, cuya descripción detallada se muestra a continuación.

### 3.2.7.1. Presentación

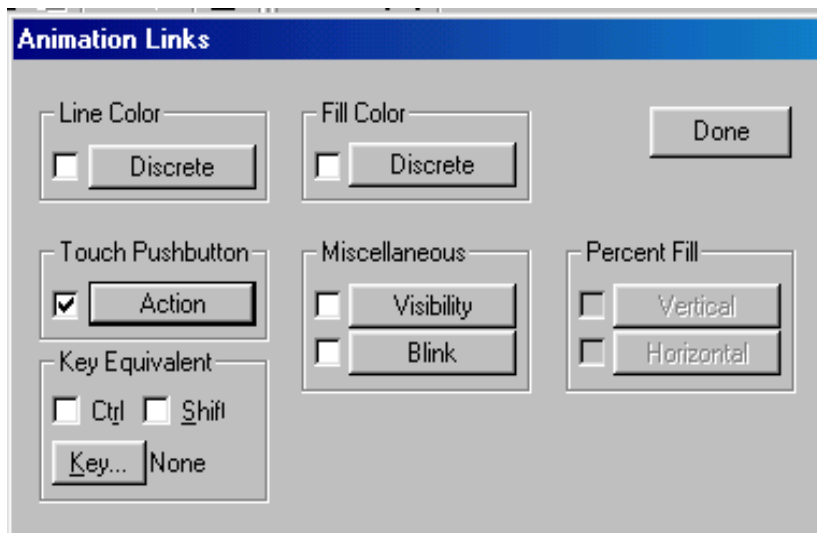
La primera pantalla que aparece en el momento del ingreso al HMI es la ventana “presentación “. Ver figura 3.9. en la que se muestra un texto básico de información acerca del proyecto.



*Figura 3.9. Pantalla de Presentación del Proyecto*



En esta pantalla está ubicado el botón “continuar” al que está vinculado el link Animation Link (*Touch PushButtons/Show Window*) (Ver Figura 3.10.)



*Figura 3.10. Ventana para Configuración del Animation Link*

Esta programación permite mostrar una segunda ventana en el momento que es presionado.

### 3.2.7.2. Botón de Registro Usuario

Para poder establecer seguridad en una aplicación desde *WindowMaker* o *WindowViewer* se debe acceder al comando *Security* en la opción *Special* en la barra de tareas, con lo que el programa permite “log off” (descargar), “log on” (cargar) un usuario, “change password”(cambia la clave), o “configure users” (configurar un nuevo usuario al sistema). Ver figura 3.11.

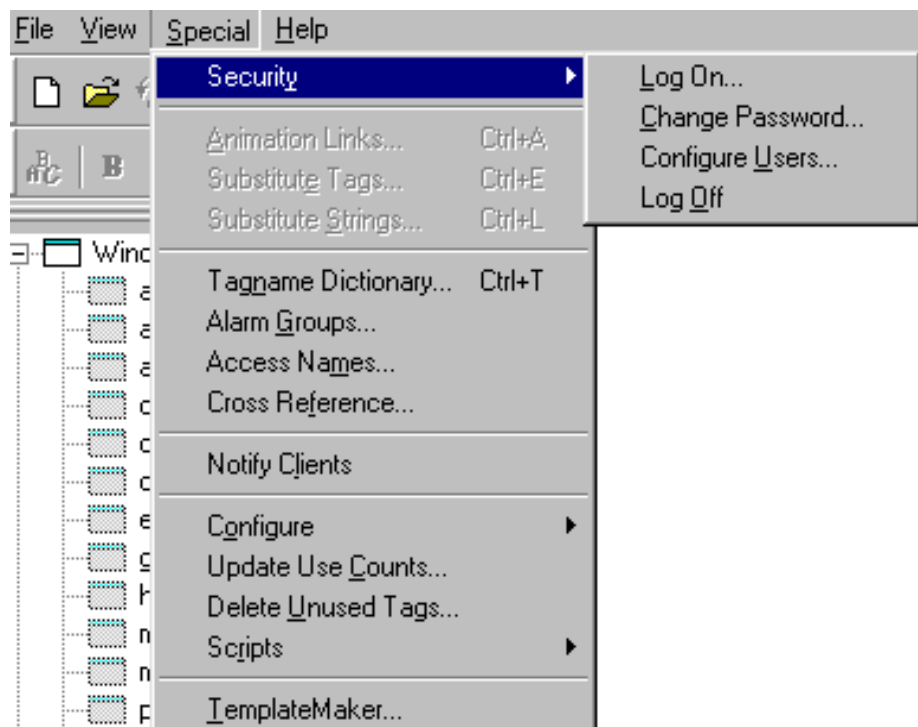


Figura 3.11. Ruta de Acceso a Security

El aspecto seguridad es importante en el HMI por lo que el tipo de seguridad implantado en el Sistema Domótico implementado, trabaja de la siguiente manera: (Tabla 3.3)

<b>NOMBRE</b>	<b>CLAVE</b>	<b>NIVEL DE ACCESO</b>	<b>DESCRIPCION</b>
Administrador	domo	8000-9999	Puede acceder a todos los recursos del sistema, modificarlos y crear usuarios.
Usuario	usuario	5000-7999	Puede operar los procesos.
Invitado	invitado	0-4999	Únicamente puede visualizar los procesos.

*Tabla 3.3. Tabla de Datos de Acceso a Seguridad*

### 3.2.7.3. Menú Principal del Sistema

Una vez que el usuario a sido registrado correctamente se habilita el Menú Principal, el cual tiene los siguientes opciones:

3.2.7.3.1. Estado del Sistema .- Nos da un informe acerca del estado de los dispositivos del sistema, es decir su estado actual, tanto de: La temperatura, presencia exterior, iluminación, alarmas, nivel del tanque, control de persiana, estados de los accesos, aire acondicionado, calefactor, así como del riego del jardín.

ESTADO ACTUAL	CONTROL	CONTROL TIEMPO	ALARMAS	GRAFICAS	SIMULACION	REPORTES		SALIR
ESTADO ACTUAL	PAUL	NIVEL ACCESO	9999	FECHA	01/03/2000	HORA	21:58:39	
<b>ESTADO DEL SISTEMA</b>								
<b>TEMPERATURA</b>  <b>ESTADO</b> ENTRE 20 Y 30 GRA			<b>ALARMAS</b> <b>INCENDIO</b>  NORMAL			<b>ESTADO DE ACCESOS</b> <b>ESTADO VENTANA</b>  ABIERTA		
<b>PRESENCIA EXTERIOR</b>  <b>ESTADO</b> NORMAL			<b>PRESENCIA INTERIOR</b>  DESACTIVADA			<b>AIRE ACONDICIONADO</b>  <b>ESTADO</b> DESACTIVADO		
<b>ILUMINACION</b> <b>SALA</b>  DESACTIVADA			<b>NIVEL TANQUE</b> <b>ESTADO</b>  NIVEL BAJO			<b>CALEFACTOR</b>  <b>ESTADO</b> DESACTIVADO		
<b>DORMITORIO</b>  ACTIVADA			<b>CONTROL PERSIANAS</b>  <b>ESTADO</b> ABIERTA			<b>RIEGO JARDIN</b>  <b>ESTADO</b> DESACTIVADO		
<b>JARDIN</b>  DESACTIVADA								

Figura 3.12. Pantalla Estado del Sistema

3.2.7.3.2. A través de la ventana principal podemos acceder también a la ventana “Módulo de Control” en la cual se muestra un esquema del proceso, es decir los objetos gráficos cambian de apariencia en relación a las acciones reales tomadas (temperatura, iluminación, control de persianas, etc.). Ver figura 3.13



Figura 3.13. Pantalla Módulo de Control

### 3.2.7.3.3. Control Tiempo

Esta ventana contiene gráficas del sistema que van a poder manipularse según la hora y el tiempo que se requiera que se ejecute determinado dispositivo, simulando presencia de personas ya que las acciones que van a ejecutar los dispositivos se las programará de acuerdo a las necesidades (Ver figura 3.14)



*Figura 3.14. Pantalla de Simulación de Presencia*

3.2.4.3.4.1. Botón Activar

Nos permite controlar de acuerdo a un tiempo deseado la iluminación, el riego de jardín y la persiana.

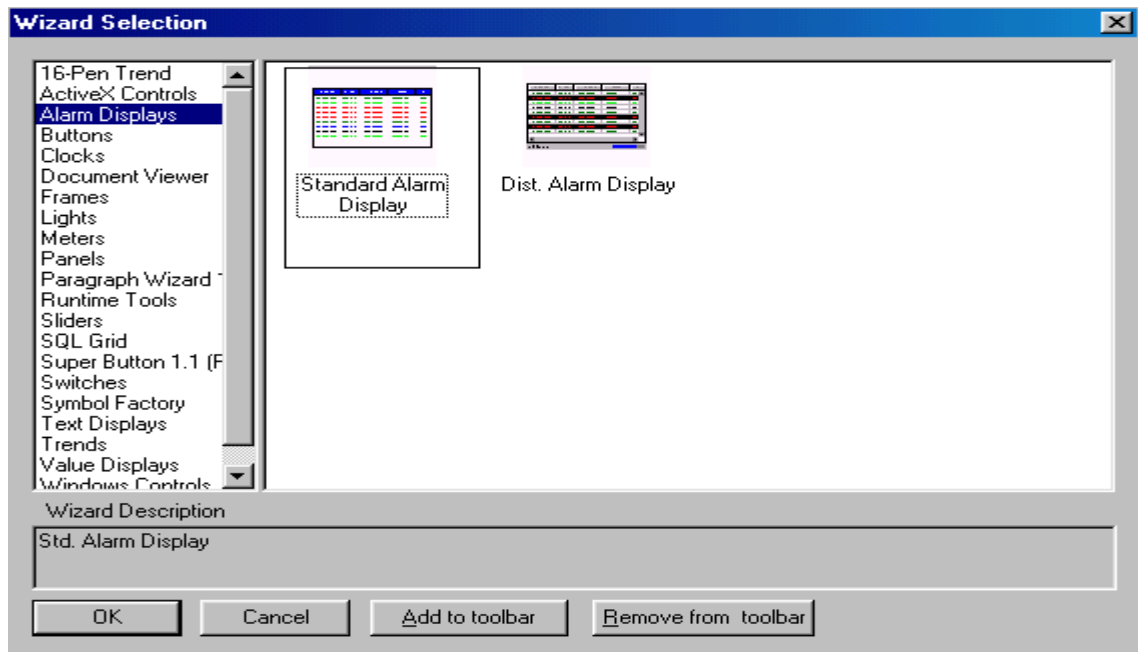
3.2.7.3.4. Modulo Alarmas y Eventos

Finalmente el botón “Alarmas/Eventos” permite abrir la ventana “ALARMAS”, que es un sistema de notificaciones que posee Intouch para informar al operador del proceso y las condiciones del sistema. Las alarmas representan advertencias de las condiciones del proceso mientras que los eventos son mensajes de estado normal del proceso. Intouch permite la muestra, actualización

e impresión de las alarmas y eventos del sistema.(Ver Figura 3.15.)

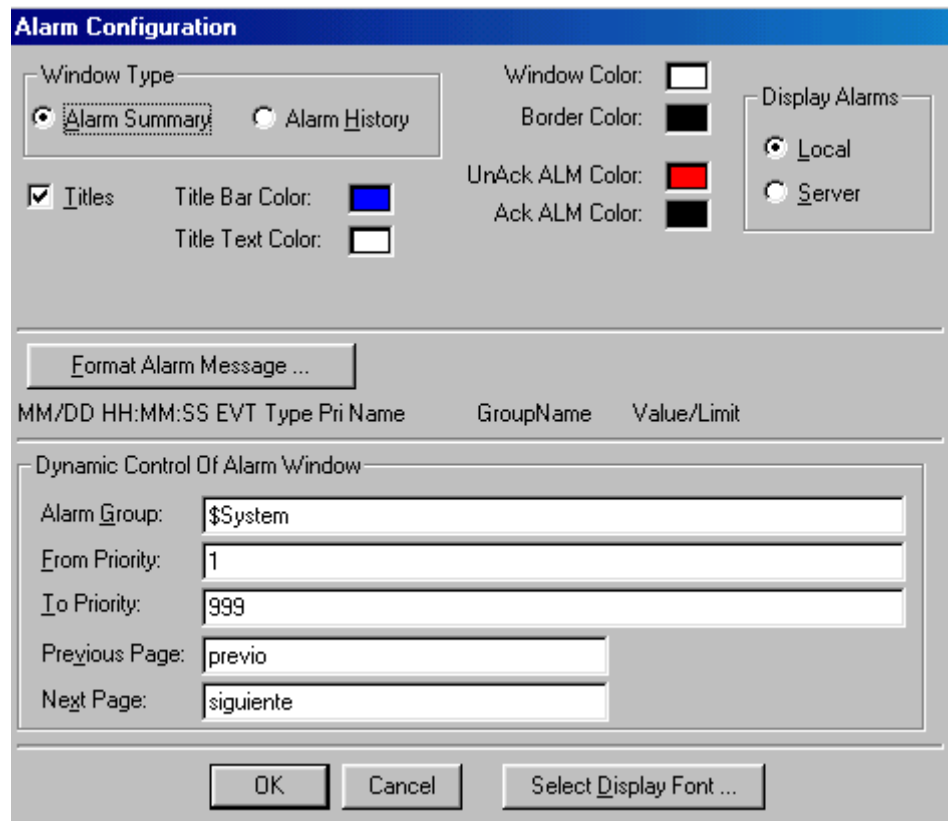






*Figura 3.16. Ventana de selección del Wizard StandarAlarm Display*

Para configurar el wizard se debe hacer clic en el mismo con lo que la siguiente ventana de configuración aparece. Figura 3.17.



*Figura 3.17. Ventana de Configuración del Wizard Standar Alarm Display*

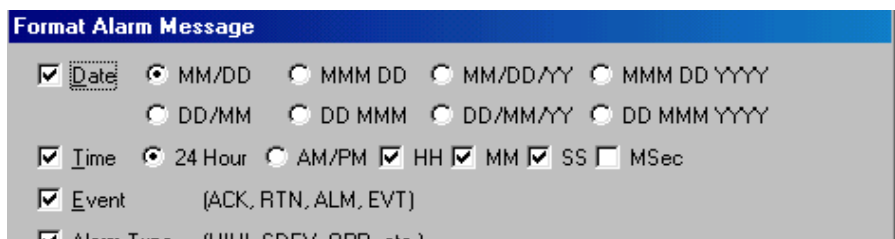
Los aspectos de mayor importancia que deben ser configurados deben son los siguientes (Tabla 3.4.):

Window Type

Alarm Summary	Muestra un resumen de las alarmas
Alarm History	Muestra un resumen de las alarmas y eventos
UnAck ALM Color	Color del mensaje de aviso de alarma desconocida
<b>Alarm Group</b>	Grupo de Alarmas que Wizard está destinado a usar
FromPriority/ToPriority	Rango de prioridad de las alarmas que serán mostradas
PreviousPage/NextPage	Nombre de dos tagnames discretos que pueden ser vinculados con botones, para que en Runtime permitan desplazarse en la ventana de alarmas y eventos.

*Tabla 3.4. Tipos de Window Type*

Cuando se da clic en *Format Alarm Message* se abre una ventana. Ver figura 3.18., que permite configurar el formato con el que el mensaje de alarma o evento va a ser mostrado en el Wizard en Runtime en el momento que este ocurre.



*Figura 3.18 .Ventana de Format Alarm Display*

Esta ventana permite configurar el formato de la fecha y hora en que sucedió la alarma o evento, mostrar el tipo, prioridad, valor, comentario, tagname, estado y más aspectos de la alarma o evento.

Para que un tagname sea mostrado por el *Standar Alarm Display Wizard* se debe previamente definir las condiciones de alarma para dicho tag en el *Tagname Dictionary*, Por ejemplo, la figura muestra la configuración para el tagname *luz* que permite registrar el estado de evento y alarma cuando exista la activación del mismo.

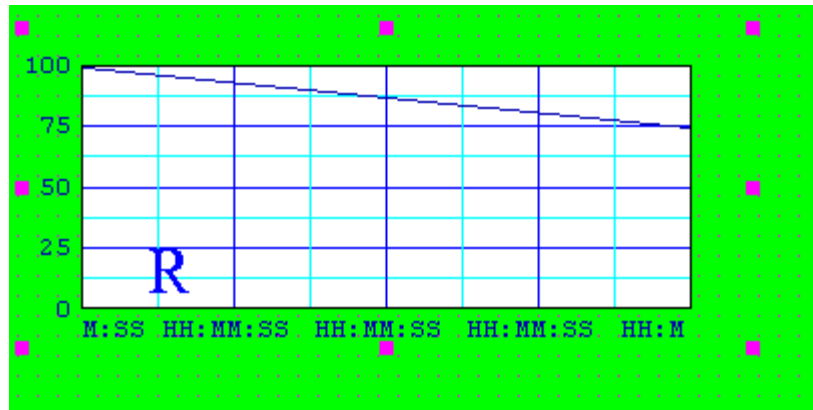
#### 3.2.7.3.5. Ventana Gráficas

La Ventana Gráficas permite abrir la ventana “GRAFICAS” en la que se representan en tiempo real las variables del proceso: Nivel del tanque, Accesos, Iluminación y Temperatura. Ver Figura 3.19.



Figura 3.19. Pantalla de Gráficas en Tiempo Real

Para crear un gráfico de tiempo real se debe seleccionar el icono *Real-time Trend* en *Drawing Toolbar*, a continuación se lo pega en la ventana de la aplicación en donde debe ser configurado. Ver figura 3.20.



*Figura 3.20. Object Real – Time Trend*

Para abrir la ventana de configuración de un *Real-time Trend* se debe hacer doble clic en el objeto.

La siguiente ventana de configuración aparece:

(Figura 3.21)



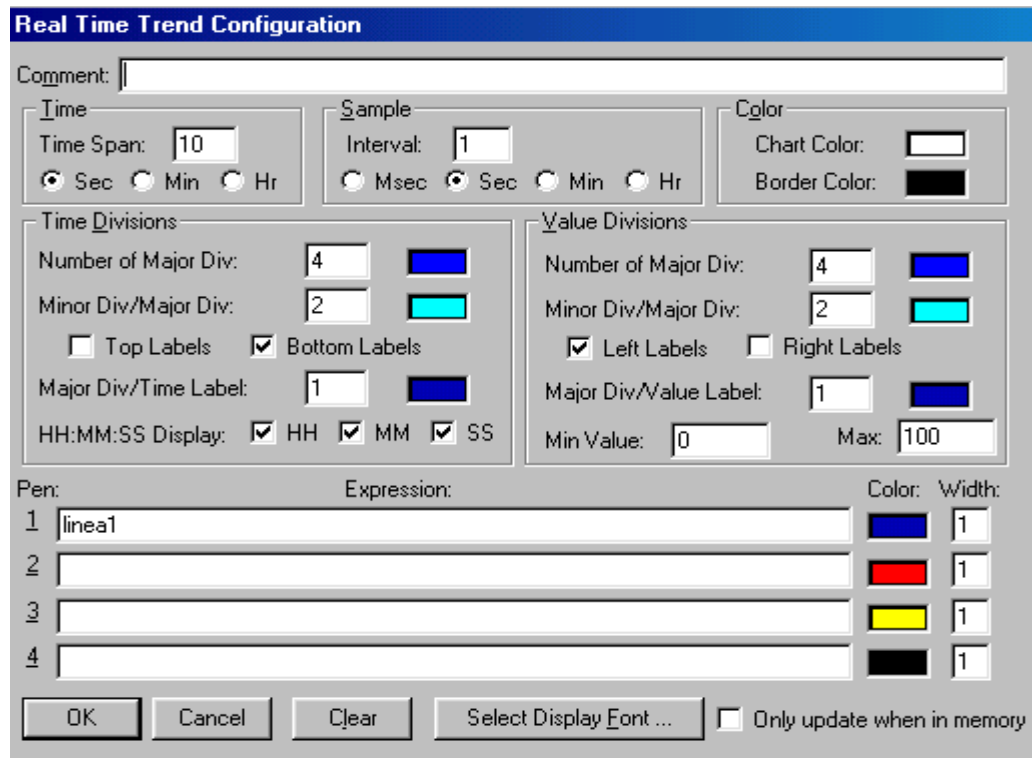


Figura 3.21. Ventana de configuración de un Real-time Trend

Esta ventana permite configurar propiedades que va a tener la gráfica en *WindowViewer*, como tiempo de muestreo, escala máxima de los ejes, divisiones de la cuadrícula en la que es dibujada la gráfica, colores y más.

Para configurar el tagname que va a ser graficado se debe escribir su nombre (*Pen*) en el área de

Expresión. Un total de cuatro tagnames pueden ser graficados en un mismo *Real-time Trend*.

La opción *Only Update when in memory* permite actualizar la gráfica únicamente cuando la ventana es mostrada, si esta opción no es activada, los datos serán actualizados incluso cuando la ventana que contiene el *Real-time Trend* no esté activa, lo que podría ocasionar un desarrollo lento del sistema en general.

Desde la ventana “GRAFICAS ” se puede imprimir la misma, además visualizar el modo de funcionamiento y dependiendo de éste, tomar acciones de control como, encender ventilador, bajar temperatura, etc

#### 3.2.4.3.4.2. Botón Históricos

“Históricos” abre la ventana HISTORICOS que permite obtener

información gráfica acerca de las acciones tomadas en el proceso.

Los históricos del proceso a diferencia de los gráficos en tiempo real proporcionan información pasada y sus datos son actualizados únicamente por medio de scripts o botones con links asociados para hacerlo.

Un histórico puede ser asociado con herramientas como barras de desplazamientos para acceder a fechas específicas, opciones de zoom para mejorar apreciación de los datos, etc. Para crear un histórico se debe seleccionar el icono *Historical Trend* en la *Drawing Toolbar* o bien utilizar uno de los *Historical Trends Wizards* ya existentes y ubicados en la opción *Trends* del *Wizards Selection*. (Figura 3.22)

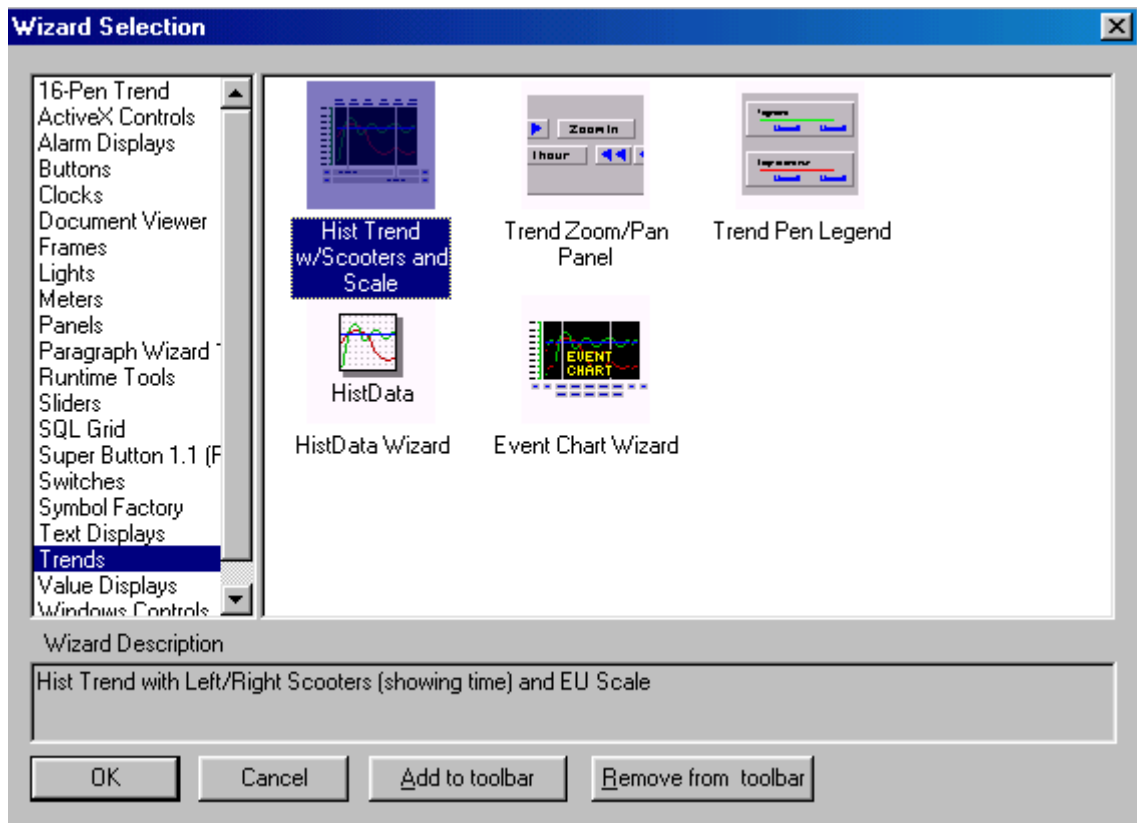


Figura 3.22. Historical Trends Wizards

El histórico desarrollado en el presente trabajo utiliza los siguientes *Historical Trends Wizards* ya existentes:

### Historical Trends w/ Scooters and

Scale.- Este wizard permite mostrar la pantalla e donde se van a visualizar las gráficas de las variables del histórico, el eje “x” con una escala de tiempo y el eje “y” puede cambiar su rango dependiendo del tipo y número de variables que estén configuradas para ser graficadas en el histórico.

Para configurar este Wizard se debe previamente crear un tagname del tipo *Hist Trend* y otro *Memory Integer* luego doble clic en el objeto del wizard con lo que la siguiente ventana aparece (Figura 3.23):

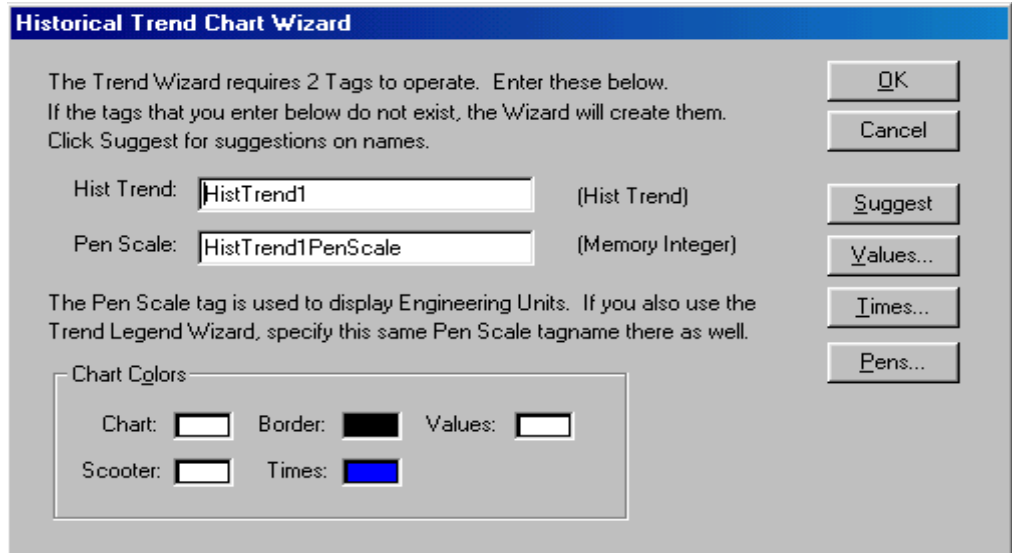


Figura 3.23. Ventana de Configuración del wizard Historical Trend w/Scooters and scale

Los tagnames previamente creados deben ser ingresados en *Hist Trend* y *Pen Scale*.

Además se puede configurar el formato y escalas de los ejes x e y (*Values* y *Times*), y los tagnames que van a ser graficados en el histórico (*Pens*).

Para que un tagname pueda ser graficado y almacenado por u histórico debe previamente ser asignado para el efecto (máximo ocho tagnames por *wizard historical trend*), para lo cual en el *Tagname Dictionary* se debe seleccionar el tag (Select...) y asignarle la opción *Log Data*. (Figura 3.24).

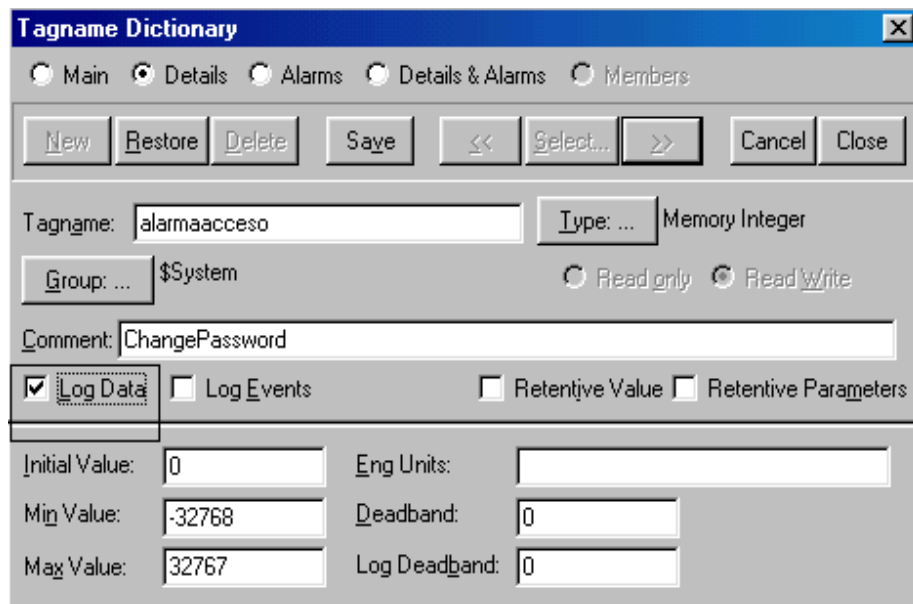


Figura 3.24. Ventana de Asignación del tagname

“alarmaacceso”

Una vez que se han asignado tagnames en la opción (*Pens...*) del *Historical Trend Wizard* se debe habilitar el almacenamiento de datos al histórico desde la opción *Enable Historical Logging* en la ventana de *Historical Logging Properties* que se accede desde la opción *Special/Configure/Historical Logging* (Figura 3.25.)

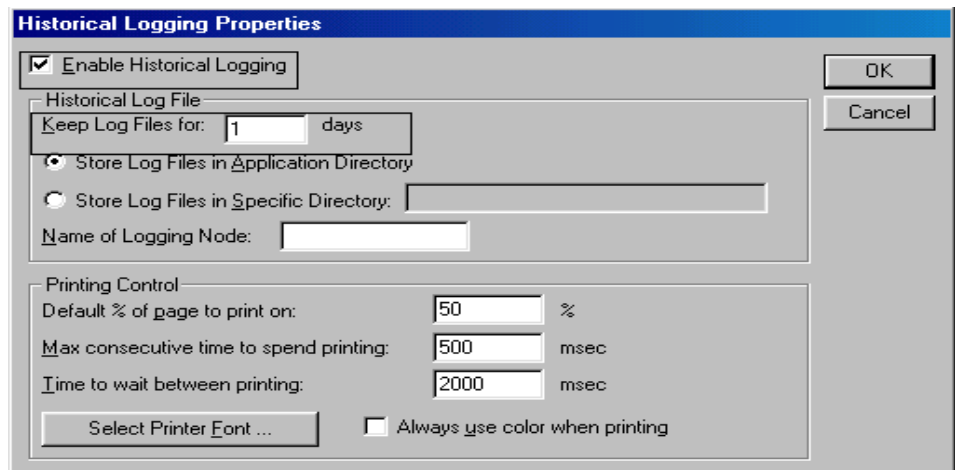


Figura 3.25. Ventana Historical Looging Proprieties



**En esta ventana se puede configurar el número de días que se desea que los archivos de los históricos permanezcan almacenados en el computador (Sep log files for...)y el directorio de destino (Store Log Files in special directory),**

**Intouch crea dos  
archivos cada día  
(.LGH y .IDX), por  
lo que se debe  
considerar el  
espacio libre en el  
disco, si no existe  
suficiente espacio el  
almacenamiento de  
históricos se detiene  
hasta que sea  
liberada memoria  
en el disco duro.**

El formato de los archivos creados por Intouch son los siguientes:

YYMMDD.LGH y  
YYMMDD.IDX, en donde (Ver  
Tabla 3.5.)

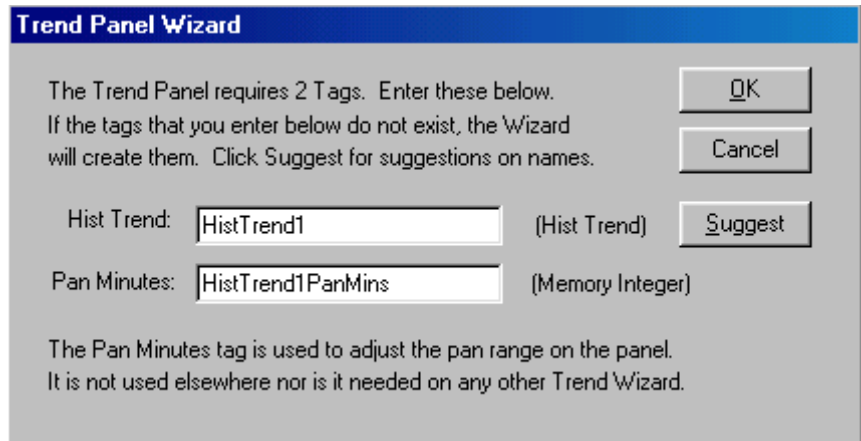
YY	Muestra el año que el archivo fue creado (99-01)
MM	Muestra el mes que el archivo fue creado (01-12)
DD	Muestra el día que el archivo fue creado (01-31)

*Tabla 3.5. Formatos de Archivos de Intouch*

Además en esta ventana se configuran las opciones de impresión cuando se imprimen los archivos históricos.

Trend/Zoom/ Pan Panel.- Este wizard muestra dos barras de botones los que básicamente permiten configurar la escala de

tiempo del *Historical Trends w/Scooters and scale* en *Window Viewer* para poder acceder más rápidamente a históricos en fecha de interés, configurar intervalos de tiempo que van a ser mostrados en la pantalla del *Historical Trends w/Scooters and scale*, tiene también opciones de *Zoom in* y *Zoom out* y un botón para actualizar los históricos hasta la fecha y hora actual. (Figura3.26)



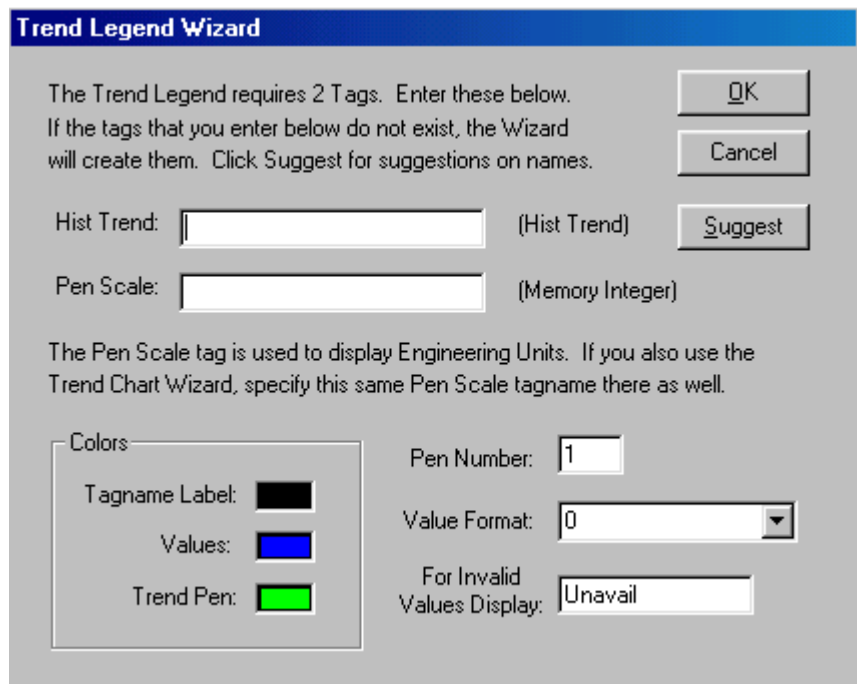
*Figura 3.26. Ventana de Configuración del Wizard Trend Zoom/Pan*

### *Panel*

Al igual que el wizard anterior su configuración necesita de dos tagnames, uno del tipo *Hist Trend* y el otro *Memory Integer*, estos tagnames corresponden a los mismos asignados al wizard anterior.

Trend pen Legend.- Este wizard muestra el nombre del tagname asociado con el *Historical Trends*

w/Scooters and scale, así como el último valor que tuvo esté la última actualización los valores de los históricos, la ventana de configuración se muestra a continuación: (Figura 3.27)



*Figura 3.27. Ventana de Configuración del Wizard Trend pen Legend*

Los tagnames *Hist Trend* y *Memory Integer* deben ser los mismos asignados al wizard *Historical Trends w/Scooters and scale*.

Hist Data Wizards.- Intouch posee un programa interno HIST DATA PROGRAM el que provee un acceso a los archivos de los históricos creados por Intouch, es usado para trasladar datos de los datos de los históricos a un programa de búsqueda como Microsoft Excel lo que ayuda a tener una base de datos de los históricos del proceso.

El *Hist Data Wizard* permite salvar los datos comprendidos en los intervalos de tiempo especificados en el wizard *Trend Zoom/Pan Panel* y guardarlos en un archivo de extensión .csv, cuando este archivo

es abierto en Microsoft Excel los  
datos son automáticamente  
separados en columnas.



### 3.2.7.3.6. Ventana de Históricos

La ventana de históricos implementada por medio de los wizards anteriormente descritos se muestra en la Figura 3.28.



Figura 3.28. Ventana HISTORICOS vista en Windows Viewer

Existen dos botones adicionales cuyos iconos se muestran a continuación:

El botón Base de datos permite abrir en Excel el archivo .csv que contiene la base de datos de los históricos del proceso, mientras que el botón imprimir, imprime el área de la gráfica de la ventana de Históricos, esto se logra vinculando el botón con un link *Touch Pushbutton Action* y la función interna *PrintHT*.

En runtime el objeto correspondiente al wizard *Historical Trends w/Scooters and scale* es configurable y permite determinar la hora y fecha de inicio del eje x en la gráfica el rango de tiempo y los tagnames a ser registrados en el histórico (*Pens...*).

#### 3.2.7.3.7. Ventana Simulación

Existe otra ventana (Figura 3.29) que nos permite manipular los dispositivos a través de un plano, por

ejemplo al dar clic sobre persianas se abrirá una ventana animada, que nos permitirá manipular el estado de la persiana.

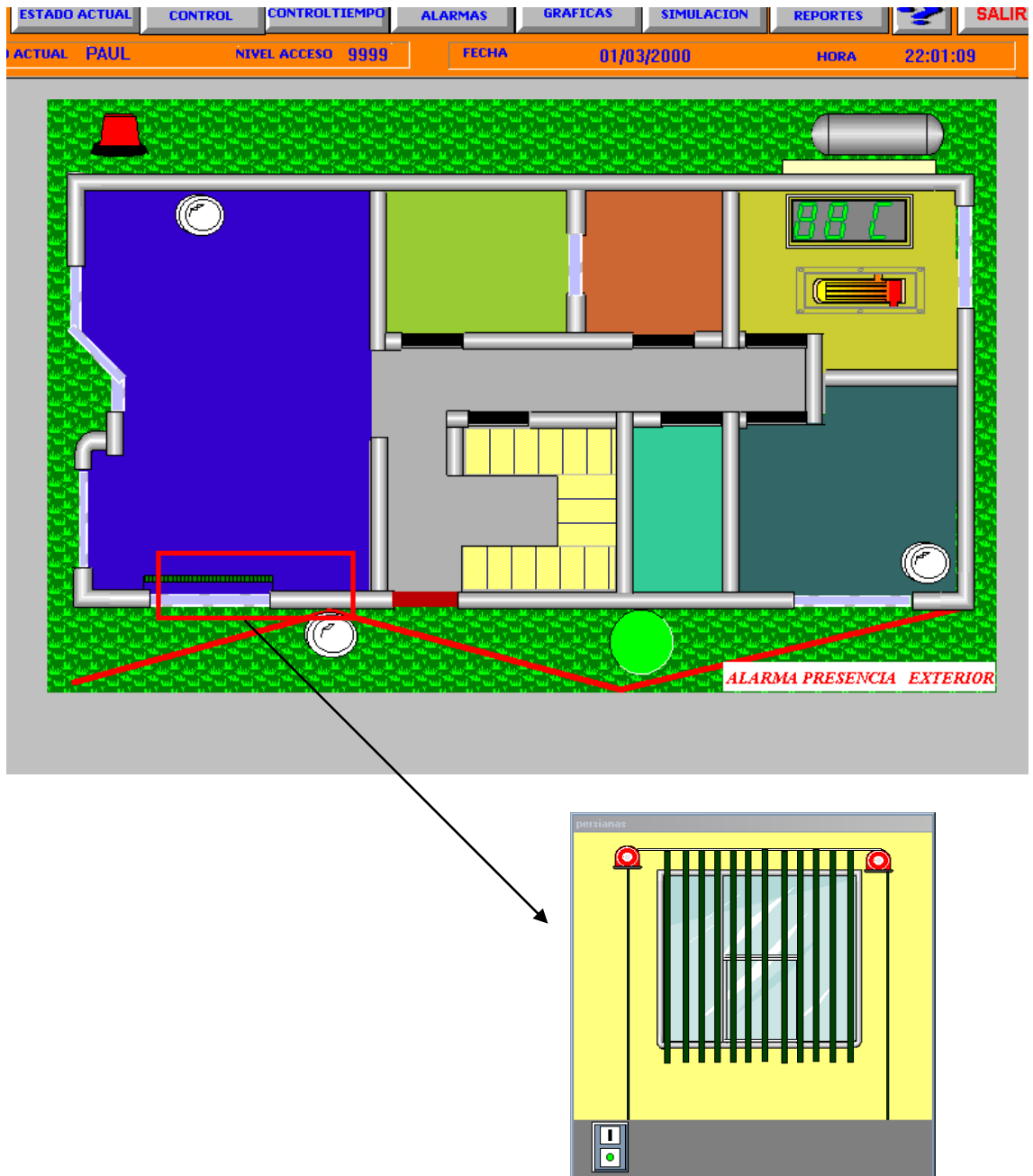


Figura 3.29. Ventana de Manipulación de la Persiana

ESTADO ACTUAL CONTROL CONTROLTIEMPO ALARMAS GRAFICAS SIMULACION REPORTES SALIR

ACTUAL PAUL NIVEL ACCESO 9999 FECHA 01/03/2000 HORA 22:01:09

ALARMA PRESENCIA EXTERIOR

The floor plan shows a building with several rooms: a large blue room on the left, a green room at the top center, a brown room at the top right, a yellow staircase in the center, a teal room at the bottom center, and a dark grey room at the bottom right. A red alarm indicator is located on the exterior wall between the blue and green rooms. A red box highlights this area, and red lines radiate from it. A green circle is also present on the exterior wall between the staircase and the teal room. A black arrow points from the red box to a detailed door view below.

puerta

*ESTADO PUERTA*

The detailed view shows a dark red door with a black handle, set against a yellow background. The door is closed. The background is a yellow rectangle with a green textured border at the bottom.

*Figura 3.30. Ventana de Manipulación del Puerta*

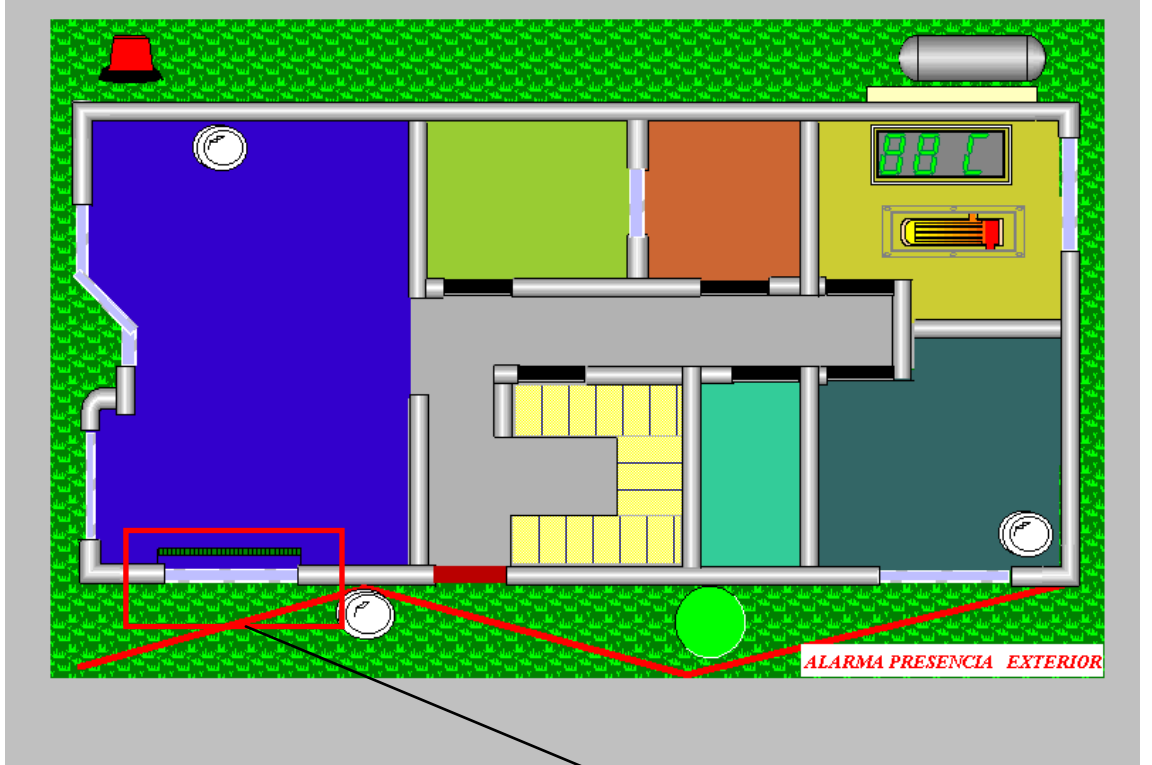
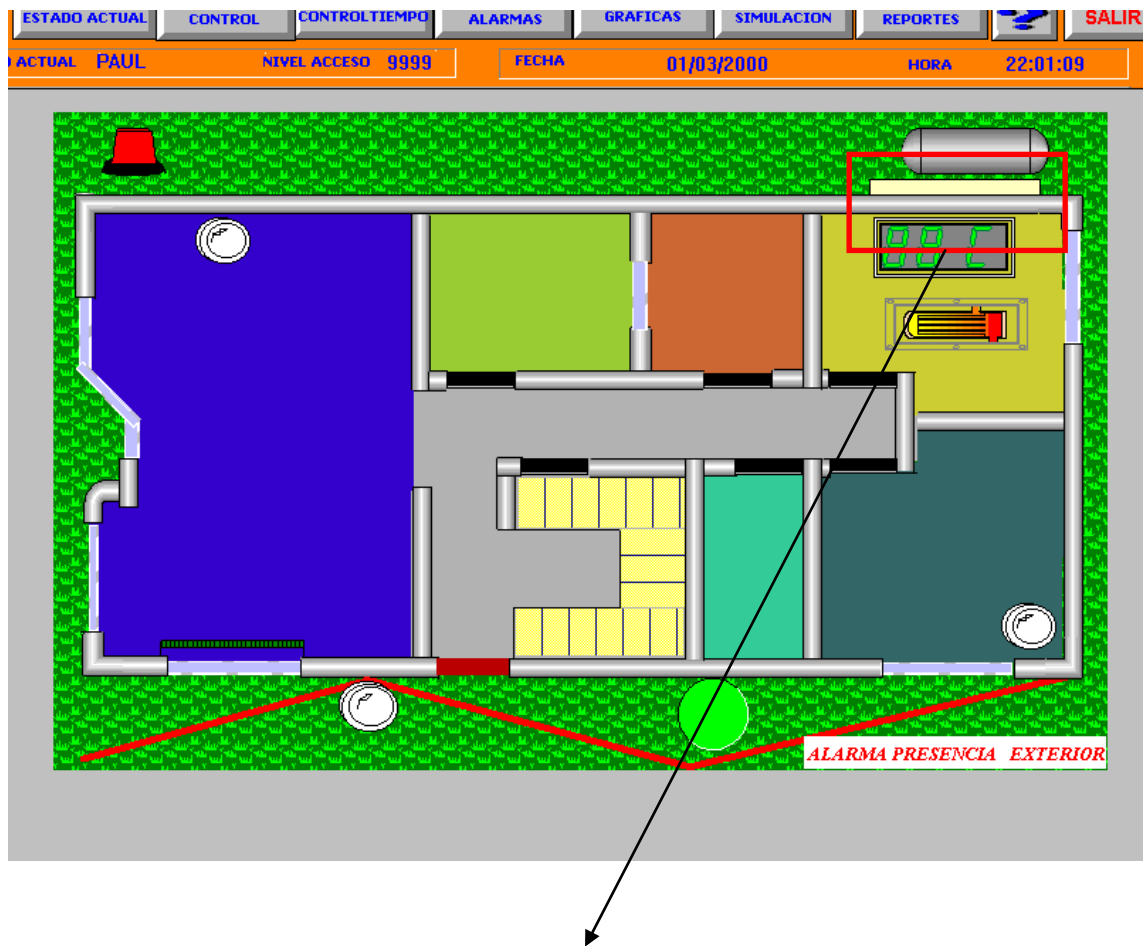


Figura 3.31. Ventana de Manipulación del Ventana



*Figura 3.32. Ventana de Manipulación del Ventilador*



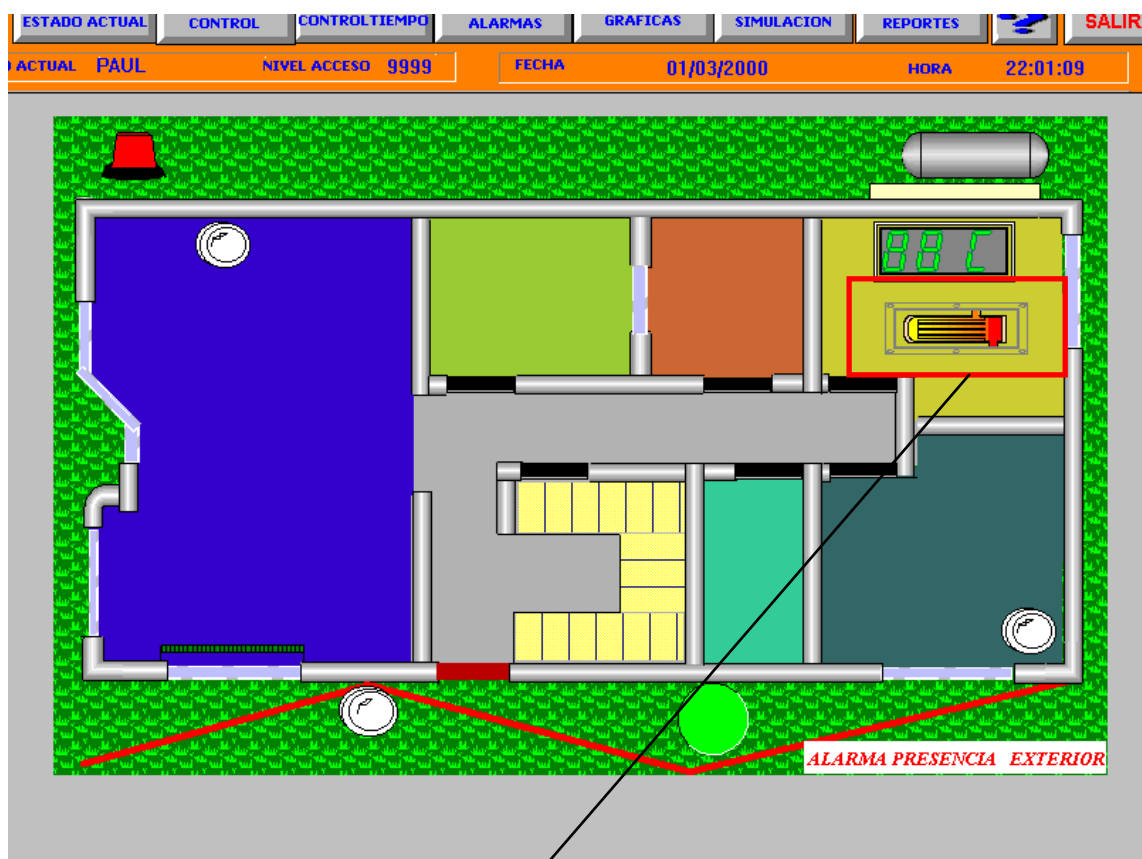


Figura 3.33. Ventana de Manipulación del Calefactor

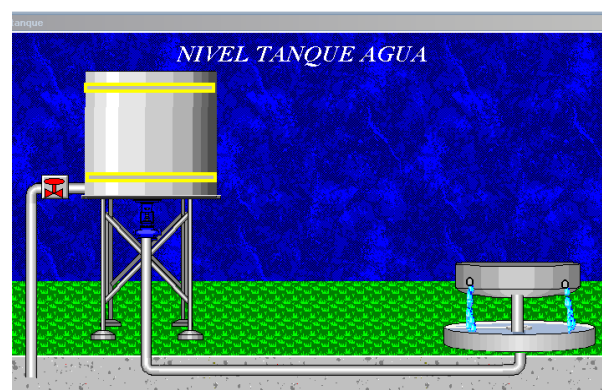
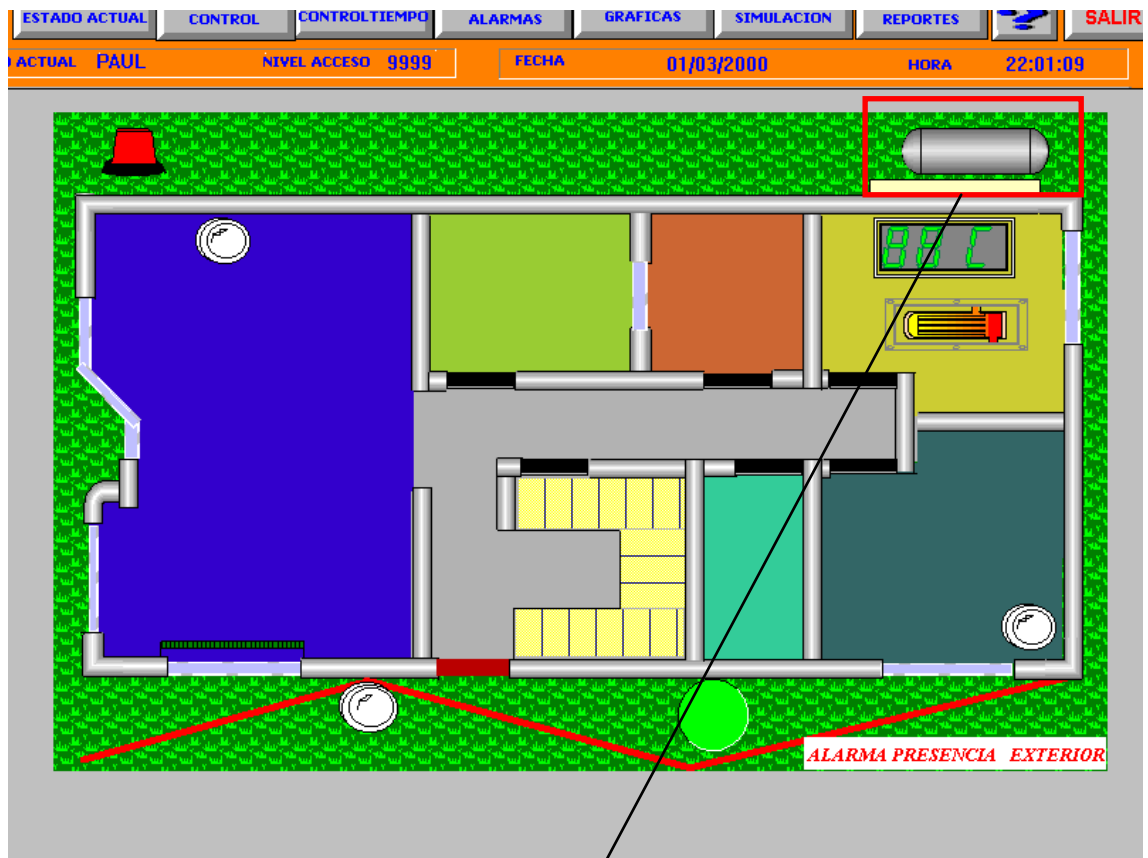


Figura 3.34. Ventana de Manipulación de la Bomba

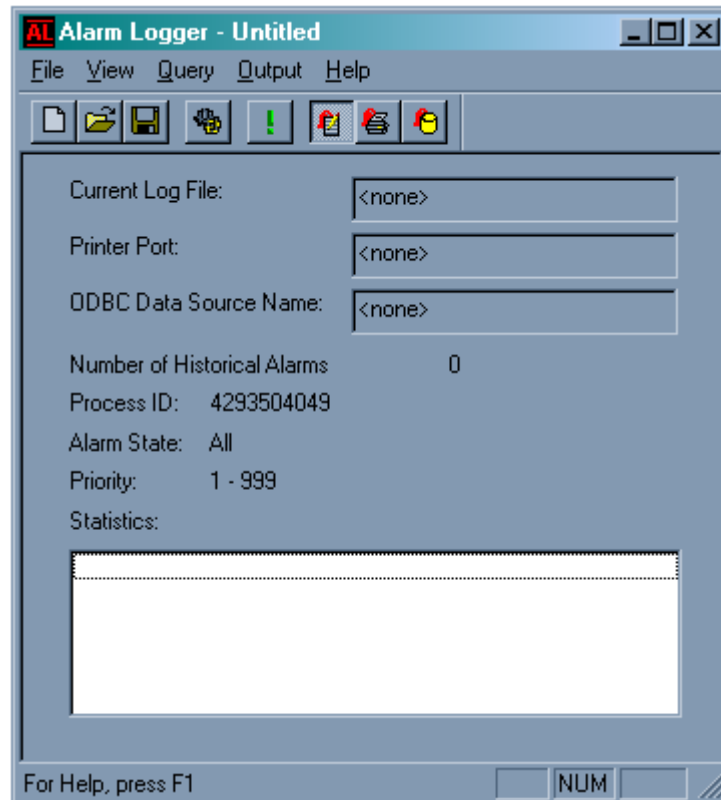
### 3.2.7.3.8. Ventana Reportes

Además de todos los reportes gráficos como los históricos que proporciona Intouch, generamos reportes con el Cristal Report, de acuerdo al operador que esta utilizando el sistema, por tipo de Alarma, por nombre de alarma y de todas las alarmas que se han generado.



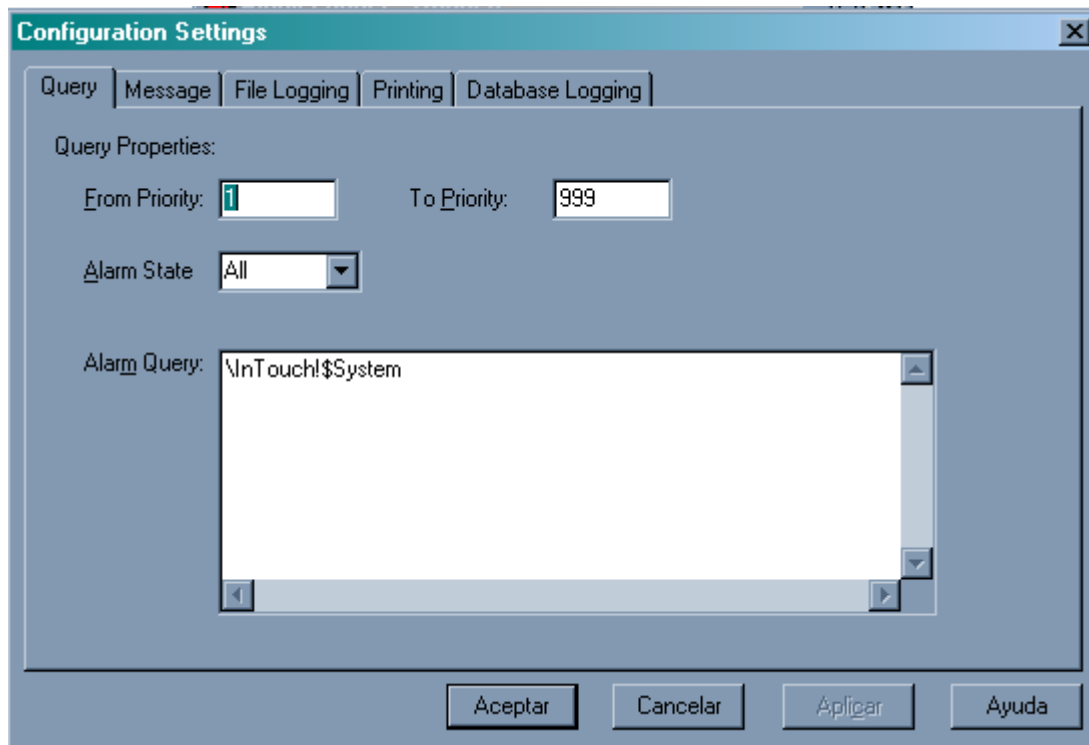
*Figura 3.35. Ventana Reportes*

Para generar estos Reportes, fue necesario primero configurar Application Logger.



*Figura 3.36. Ventana Alarm Logger*

Para configurar una consulta de acuerdo a nuestros requerimientos tenemos que configurar la Ventana Cofiguration Settings (Figura 3.37).

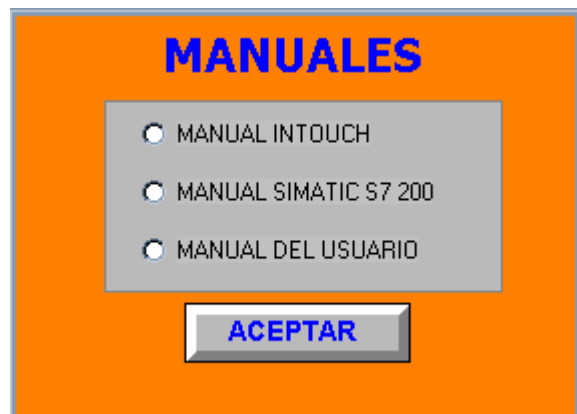


*Figura 3.37. Ventana Configuration Settings*

Para poder exportar los datos que se generen de la consulta en una tabla generada en SQL Server, para lo cual debemos crear un ODBC.

### 3.2.7.3.9. Ventana Ayuda

En esta ventana podremos elegir entre el Manual de Usuario que nos guiará en el uso del Sistema, el Manual del PLC Simatic S7-200 el Manual de Intouch.



*Figura 3.38. Ventana Ayuda*

### 3.2.8. Herramientas a Utilizar

Las herramientas utilizadas para el desarrollo del Sistema son:

- STEP 7 MICROWIN 3.2
  
- Driver de Comunicación S7200 PPI
  
- INTOUCH 7.1
  
- TANGO PCB
  
- SQL SERVER 7.0
  
- CRYSTAL REPORT 7.0

### 3.2.9. Implementación

La implementación la realizamos acoplando tanto el módulo del PLC, la tarjeta electrónica así como todos los dispositivos utilizados en la

Maqueta (tanques, motores, ventilador, calentador, persiana, focos, sensores magnéticos, infrarrojo, sensor de humo) así como el correspondiente cableado de cada uno de ellos, y finalmente el acople de cada una de las marcas que utilizamos en Intouch para comunicarnos con el Programa desarrollo con el STEP7 MICROWIN3.2 y a la vez este activé las respectivas entradas y salidas del PLC.

### ***3.3 Pruebas y resultados del prototipo construido.***

Una vez diseñado y construido el prototipo procedemos a realizar las Pruebas para su correcto funcionamiento, tanto de cada uno de los Circuitos, así como del PLC y su conexión con la PC, comprobando que las señales enviadas y recibidas por cada uno de los circuitos activen las respectivas entradas y salidas asignadas, además de la comprobación de que todas las conexiones del cableado realicen sus funciones específicas.

### ***3.4 Pruebas y resultados del prototipo completo.***



Durante el proceso de pruebas se llevo a cabo una evaluación total de todos los elementos del sistema. La evaluación del prototipo se hizo a lo largo de todo el desarrollo del mismo, no solamente al final, es decir que se realizaron conforme progresaba el trabajo en los módulos del programa acoplado con el Hardware, lo que permitió identificar todos o por lo menos la mayoría de aquellos problemas de comunicación entre el sistema desarrollado (SOFTWARE) y el prototipo construido (HARDWARE), para así depurarlos y garantizar su correcto funcionamiento.

## BIBLIOGRAFÍA

**INSTALACIONES DOMOTICAS** Mateos, F

Ed Genia

México 1998

**MONTAJES DOMÓTICOS**

Tavernier Christian

Paraninfo

España, 1995.

**ANÁLISIS Y DISEÑO DE SISTEMAS** Siemens

1991.

**AUTÓMATAS PROGRAMABLES**

Ings. Alejandro Porras y Antonio Placido

Montanero .

1998.

**DIAGRAMAS PARA CONTROL INDUSTRIAL** Ings. PAULO  
ANGULO,

JORGE MOLINA,

1998

## **DIRECCIONES DE INTERNET**

<http://www.domoticaviva.com>

<http://www.plcs.net>

<http://www.cybernomo.com/scm/PLCs/t100mx.htm>

<http://www.exclbr.com/101/plc.htm>

<http://www.xtronics.com/Tosh/plc.htm>

<http://www.rhrtech.com/html/plc.html>

<http://www.esmii.com/PLC.htm>

# **GLOSARIO**

**Analógico. – El término "Analógico" se refiere a las magnitudes o valores que varían con el tiempo en forma continua como la distancia y la temperatura, la velocidad, que podrían variar muy lento o muy rápido como un sistema de audio. En la vida cotidiana el tiempo se representa en forma analógica por relojes (de agujas), y en forma discreta (digital) por displays digitales. En la tecnología analógica es muy difícil almacenar, manipular, comparar, calcular y recuperar información con exactitud, en cambio en la tecnología digital (computadoras, por ejemplo), se pueden hacer tareas muy rápidamente, muy exactas, muy precisas y sin detenerse.**

**Amplificador Operacional.- Los amplificadores son circuitos que se utilizan para aumentar (amplificar) el valor de la señal de entrada (generalmente muy pequeña) y así obtener una**

**señal a la salida con una amplitud mucho mayor a la señal original.**

**Autómata Programable.- El autómata programable es un aparato electrónico programable por un usuario programador y destinado a gobernar, dentro de un entorno industrial, maquinas o procesos lógicos secuenciales.**

**Código BCD.- BCD son las iniciales de unas palabras inglesas que traducidas vendrían a significar Código Decimal codificado en Binario. Es decir cada cifra decimal se codifica según una serie de bits binarios ¿Cuántos?, como existen diez cifras del 0 al 9 necesitamos 4 bits por cifra. (Con 3 nos quedaríamos cortos ya que como máximo podríamos codificar 8 cifras). Ahora resulta que con 4 bits podríamos codificar hasta 16 cifras, luego vemos que hay 6 combinaciones (de**

**1010 a 1111) que nunca se utilizan en el código BCD; de ahí que este código sea menos compacto que el binario puro.**

**Comparadores de Tensión.-** Son circuitos no lineales que, como su nombre indican sirven para comparar dos señales (una de las cuales es generalmente una tensión de referencia) y determinar cuál de ellas es mayor o menor

**Condensador.- Un condensador o capacitor es un dispositivo electrónico que está formado por dos placas metálicas separadas por un aislante llamado dieléctrico. Un dieléctrico o aislante es un material que evita el paso de la corriente. Es un dispositivo que almacena energía en la forma de un campo eléctrico (es evidente cuando el capacitor funciona con corriente directa) y se llama capacitancia o capacidad a la cantidad de cargas eléctricas que es capaz de almacenar.**

**Corriente continua.- Es el resultado de el flujo de electrones (carga negativa) por un conductor (alambre de cobre casi**

**siempre), que va del terminal negativo al terminal positivo de la batería, pasando por el foco / bombillo.**

**Decodificador.- Un decodificador es un circuito lógico combinacional, que convierte un código de entrada binario de N bits en M líneas de salida (N puede ser cualquier entero y M es un entero menor o igual a  $2^N$ ), tales que cada línea de salida será activada para una sola de las combinaciones posibles de entrada.**

**Digital.- El término "Digital" se refiere a cantidades discretas como la cantidad de personas en un una sala, cantidad de libros en una biblioteca, cantidad de autos en una zona de estacionamiento, etc.**



**Domótica.-** El término Domótica viene del latín “domus” que significa casa y de la palabra “automática”, por lo tanto la domótica se refiere a una casa automática o como se le ha llamado más comúnmente una casa inteligente. En inglés a la domótica se le conoce más como “home networking” o “smart home”. Una casa inteligente es aquella cuyos elementos o dispositivos están integrados y automatizados a través de una red (principalmente Internet) y que a través de otro dispositivo remoto o inclusive interno se pueden modificar sus estados o los mismos dispositivos están diseñados para realizar ciertas acciones cuando han detectado cambios en su propio estado.

**Fotorresistencia LDR (Resistencia dependiente de la luz).-** El LDR es una resistencia que varía su valor dependiendo de la cantidad de luz que la ilumina. Los valores de una fotorresistencia cuando está totalmente iluminada y cuando está totalmente a oscuras varía, puede medir de 50 ohmios a 1000 ohmios (1K) en iluminación total y puede ser de 50K (50,000 Ohms) a varios megaohmios cuando está a oscuras.

**Frecuencia.- Término empleado en física para indicar el número de veces que se repite en un segundo cualquier fenómeno periódico. La frecuencia es muy importante en muchas áreas de la física, como la mecánica o el estudio de las ondas del sonido.**

**InTouch.-** Es un generador de aplicaciones MMI destinadas a la automatización industrial, control de procesos y supervisión, nos ofrece la posibilidad de generar aplicaciones SCADA al más alto nivel, utilizando las herramientas de programación orientadas a objetos, para usuarios no informáticos.

**PLC.- Las siglas PLC significa Programmable Logic Controller, que como su propio nombre indica es un controlador lógico programable.**

**Potencia.- Es la rapidez o velocidad con que se consume la energía se mide en julios/segundo. El trabajo, o transferencia de energía, realizado por unidad de tiempo. El trabajo es**

**igual a la fuerza aplicada para mover un objeto multiplicada por la distancia a la que el objeto se desplaza en la dirección de la fuerza.**

**Relé.- El Relé es un interruptor operado magnéticamente.**

**Este se activa o desactiva (dependiendo de la conexión) cuando el electroimán (que forma parte del Relé) es energizado (le damos el voltaje para que funcione). Esta operación causa que exista conexión o no, entre dos o más terminales del dispositivo (el Relé).**

**Resistencia.- Son componentes electrónicos que tienen la propiedad de oponerse al paso de la corriente eléctrica. La unidad en la que se mide esta característica es el Ohmio y se representa con la letra griega Omega.**

**SCADA.- SCADA viene de las siglas de "Supervisory Control And Data Acquisition", es decir: adquisición de datos y control de supervisión. Se trata de una aplicación software especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controlando el proceso de forma automática desde la pantalla del ordenador.**

**Sensores.- Los sensores o detectores reconocen entradas procedentes del proceso y entorno externo. Son un tipo de transductores que convierten la información física real, como presión, temperatura, posición, etc., en señal de tipo eléctrico, utilizándose éstas señales para supervisar y controlar un proceso. Los sensores suelen clasificarse en función de lo que midan ( temperaturas, presión, velocidad, nivel, etc., que**

serían señales analógicas ) y también detectan estados ON / OFF, ( que serían señales digitales).

**Tagname Dictionary.-** El Tagname Dictionary (Diccionario de nombres de etiquetas), es el “corazón” del Intouch, ya que es aquí donde se definen las variables que serán utilizadas en la aplicación desarrollada.

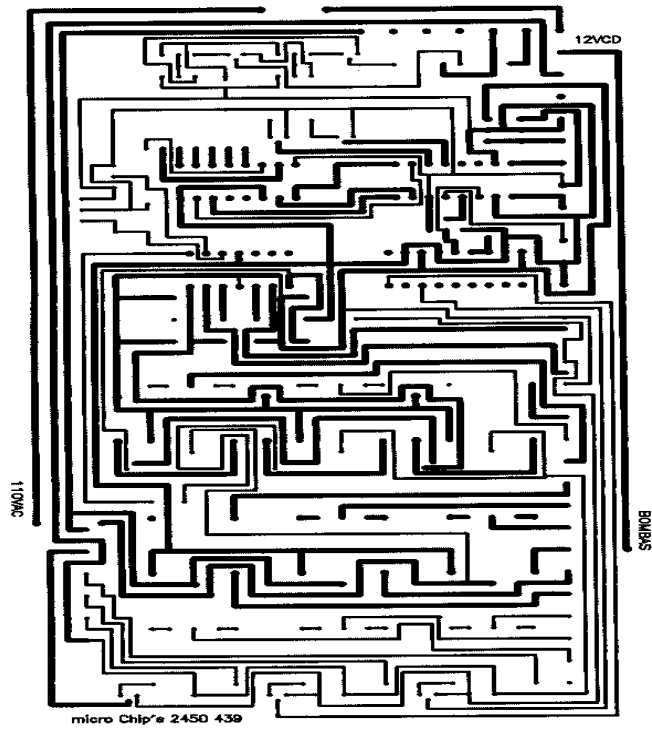
Tarjetas de expansión. – **Como el sistema operativo sólo puede proporcionar respuestas suaves en tiempo real lo más simple es emplear extensiones hardware para las tareas críticas (placas de expansión PC) y soluciones software para el resto de tareas. Esto nos lleva a una compatibilidad con futuros sistemas operativos y una solución totalmente factible actualmente. Estas tarjetas de expansión asumen las tareas críticas en tiempo real que el ordenador (PC) no puede atender, se está hablando de tarjetas que incorporan DSPs (Procesadores de Señales Digitales) o microcontroladores y**

**que aportan una ayuda a la anterior “sobrecarga”  
mencionada para los ordenadores (PC).**

**Tiempo real.** – La capacidad en tiempo real se refiere a la capacidad del ordenador en programas de procesamiento de datos para que siempre esté listo para procesar y proporcionar los resultados dentro de un tiempo especificado

# **ANEXOS**

### Anexo 3.1. Diseño de la Tarjeta



### Anexo 3.2. Maqueta del Prototipo





## Anexo 3.3 Codificador BCD CA3161

August 1997

### BCD to Seven Segment Decoder/Driver

#### Features

- TTL Compatible Input Logic Levels
- 25mA (Typ) Constant Current Segment Outputs
- Eliminates Need for Output Current Limiting Resistors
- Pin Compatible with Other Industry Standard Decoders
- Low Standby Power Dissipation . . . . . 18mW (Typ)

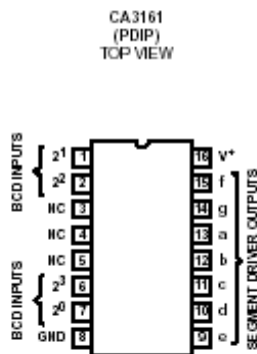
#### Description

The CA3161E is a monolithic integrated circuit that performs the BCD to seven segment decoding function and features constant current segment drivers. When used with the CA3162E A/D Converter the CA3161E provides a complete digital readout system with a minimum number of external parts.

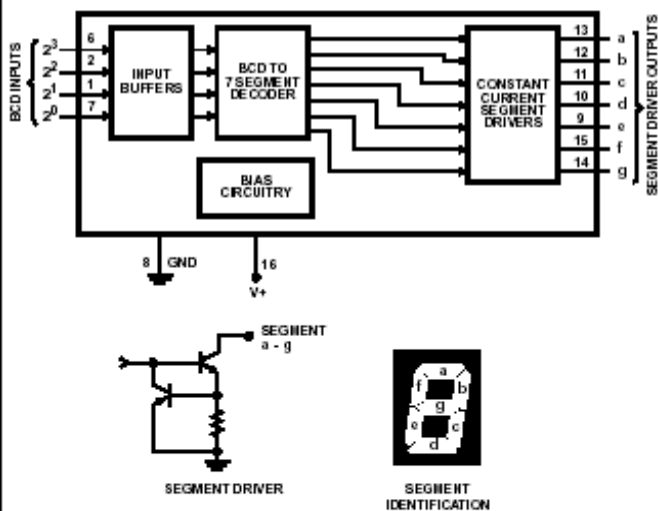
#### Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
CA3161E	0 to 70	16 Ld PDIP	E16.3

#### Pinout



#### Functional Block Diagram





## Anexo 3.4. Conversor Análogo Digital 3162

December 1993

## A/D Converter for 3-Digit Display

### Features

- Dual Slope A/D Conversion
- Multiplexed BCD Display
- Ultra Stable Internal Band Gap Voltage Reference
- Capable of Reading 99mV Below Ground with Single Supply
- Differential Input
- Internal Timing - No External Clock Required
- Choice of Low Speed (4Hz) or High Speed (96Hz) Conversion Rate
- "Hold" Inhibits Conversion but Maintains Delay
- Overrange Indication
  - "EEE" for Reading Greater than +999mV, "-" for Reading More Negative than -99mV When Used With CA3161E
- BCD-to-Seven-Segment Decoder/Driver
- Extended Temperature Range Version Available

### Description

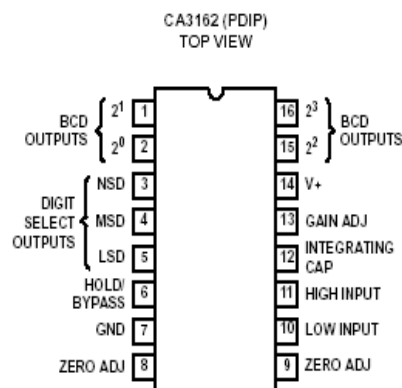
The CA3162E and CA3162AE are  $I^2L$  monolithic A/D converters that provide a 3 digit multiplexed BCD output. They are used with the CA3161E BCD-to-Seven-Segment Decoder/Driver\* and a minimum of external parts to implement a complete 3 digit display. The CA3162AE is identical to the CA3162E except for an extended operating temperature range.

\* The CA3161E is described in Display Drivers section of this data book.

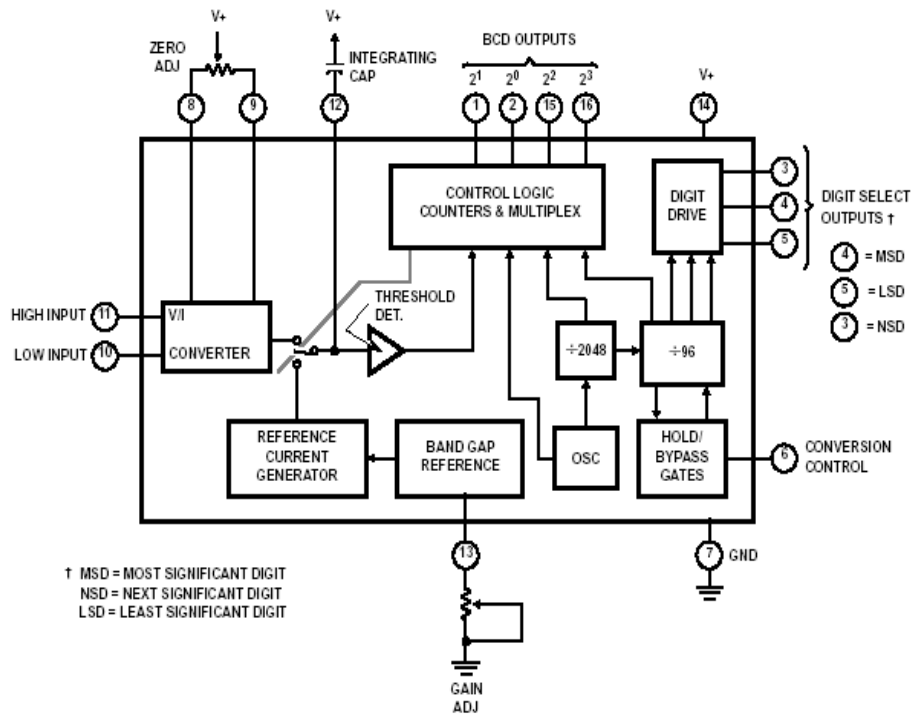
### Ordering Information

PART NUMBER	TEMPERATURE RANGE	PACKAGE
CA3162E	0°C to +70°C	16 Lead Plastic DIP
CA3162AE	-40°C to +85°C	16 Lead Plastic DIP

### Pinout



**Functional Block Diagram**



### Timing Diagram

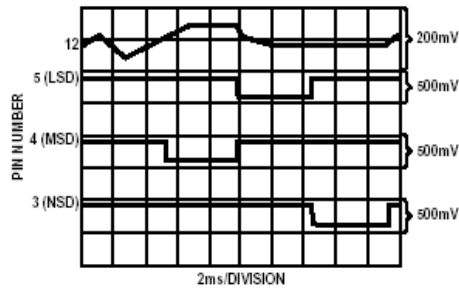


FIGURE 1. HIGH SPEED MODE

### Detailed Description

The Functional Block Diagram of the CA3162E shows the V/I converter and reference current generator, which is the heart of the system. The V/I converter converts the input voltage applied between pins 10 and 11 to a current that charges the integrating capacitor on pin 12 for a predetermined time interval. At the end of the charging interval, the V/I converter is disconnected from the integrating capacitor, and a band gap

reference constant current source of opposite polarity is connected. The number of clock counts that elapse before the charge is restored to its original value is a direct measure of the signal induced current. The restoration is sensed by the comparator, which in turn latches the counter. The count is then multiplexed to the BCD outputs.

The timing for the CA3162E is supplied by a 786Hz ring oscillator, and the input at pin 6 determines the sampling rate. A 5V input provides a high speed sampling rate (96Hz), and grounding or floating pin 6 provides a low speed (4Hz) sampling rate. When pin 6 is fixed at +1.2V (by placing a 12K resistor between pin 6 and the +5V supply) a "hold" feature is available. While the CA3162E is in the hold mode, sampling continues at 4Hz but the display data are latched to the last reading prior to the application of the 1.2V. Removal of the 1.2V restores continuous display changes. Note, however, that the sampling rate remains at 4Hz.

Figure 1 shows the timing of sampling and digit select pulses for the high speed mode. Note that the basic A/D conversion process requires approximately 5ms in both modes.

The "EEE" or "--" displays indicate that the range of the system has been exceeded in the positive or negative direction, respectively. Negative voltages to -99mV are displayed with the minus sign in the MSD. The BCD code is 1010 for a negative overrange (--) and 1011 for a positive overrange (EEE).

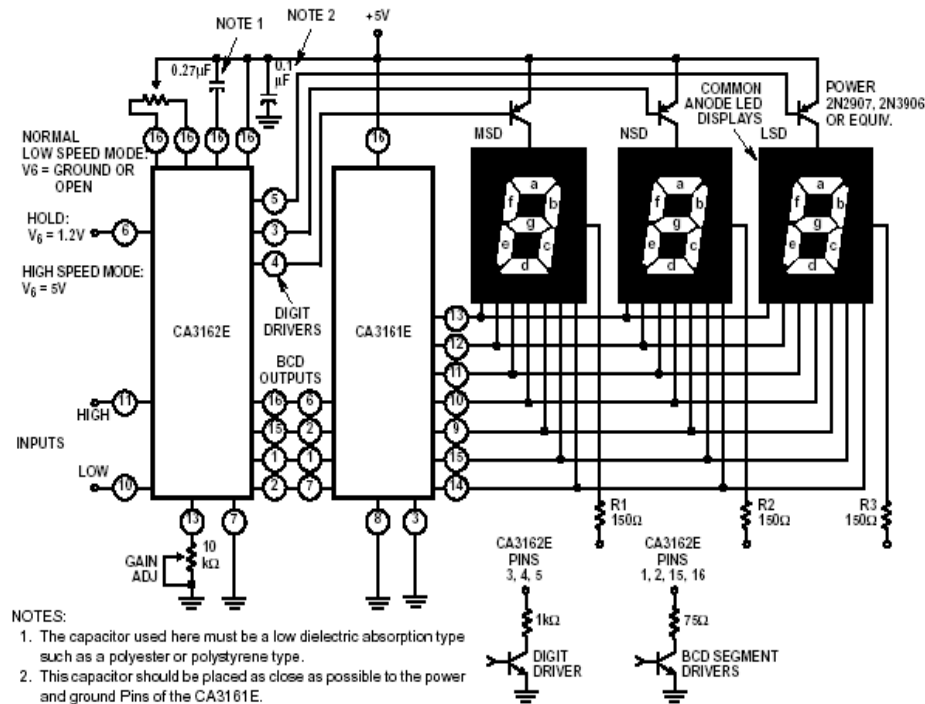


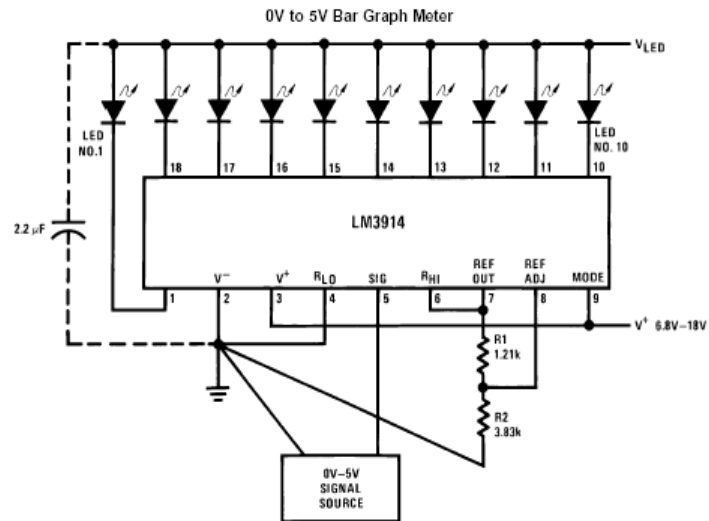
FIGURE 2. BASIC DIGITAL READOUT SYSTEM USING THE CA3162E AND THE CA3161E



## Anexo. Comparador de Tensión 3914

LM3914

### Typical Applications



DS00790-1

$$\text{Ref Out } V = 1.25 \left( 1 + \frac{R_2}{R_1} \right)$$

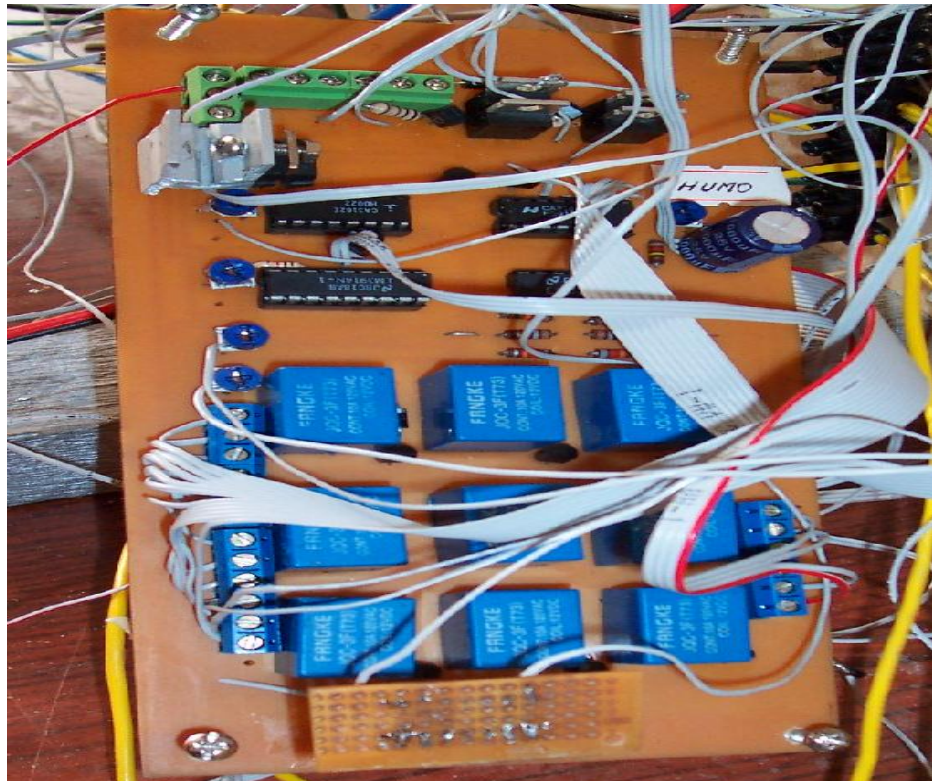
$$I_{\text{LED}} \cong \frac{1.25}{R_1}$$

Note: Grounding method is typical of all uses. The 2.2 µF tantalum or 10 µF aluminum electrolytic capacitor is needed if leads to the LED supply are 6" or longer.





Tarjeta Principal





## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

## ***4.1 Conclusiones***

- La Domótica es un gran avance en la tecnología, sin embargo, la gente todavía no está muy preparada para su implementación en las casas ya que las personas están acostumbradas a realizar manualmente toda actividad casera. Es por esto que este Sistema de Control tratará de dar un panorama de la domótica para que la gente vaya conociendo esta tecnología.
  
- A lo largo de este documento se explicó una breve reseña de la evolución de la domótica y todos los campos que cubre y que nos ayudará a automatizar ciertas tareas de la casa. Los distintos protocolos que existen para la domótica nos dan un criterio para su uso en determinados dispositivos lo que servirá para que en un futuro se pueda implementar en una casa real.
  
- Tanto el software SCADA INTOUCH como el SIMATIC S7-200 han prestado una labor de apoyo al diseño y desarrollo del trabajo que ha sido plenamente satisfactoria, permitiendo garantizar el uso de estas herramientas en proyecto reales.
  
- Con la automatización se ha logrado tener un Sistema Domótico confiable, óptimo y adecuado a las necesidades de control y monitoreo para las viviendas.

- El sistema PC/PLC tiene en la actualidad una gran importancia en la automatización de gran parte de la industria, que va desde aplicaciones sencillas a las más complejas, en el caso de la presente tesis hemos aplicado en la automatización de una vivienda. Para realizarlo hemos utilizado un driver de comunicación que satisface adecuadamente los objetivos planteados al inicio del proyecto. La interfaz desarrollada puede ser aplicada en una vivienda, con el beneficio de realizar control y monitoreo de procesos en tiempo real con un sistema muy confiable y seguro.
- En el presente tesis hemos utilizado el computador como el encargado de realizar el control y monitoreo, y el PLC el encargado de realizar el control, esto hace del sistema PC / PLC una solución muy confiable y segura a la hora de escoger un sistema de automatización que tenga las ventajas de solucionar el control de una manera más fácil, confiable y trabajando en tiempo real.
- La Programación del PLC usado es sencilla y permite enfocarse en la lógica del programa. Si bien es cierto el uso de señales analógicas incrementa las aplicaciones de los del PLCs en sistemas que involucran acciones de control más complejas, se debe tener en cuenta los tiempos de conversión y toma de los datos ya que se podrían tener lecturas incorrectas lo que implica acciones de control no deseadas.

## ***4.2 Recomendaciones***

- El sistema PC/PLC utilizado en la presente investigación es de fácil ampliación debido a la modularidad que ofrece, sea en cuanto a la adquisición de un módulo de señales analógicas o digitales, para controlar un mayor número de tareas .
  
- Se recomienda en futuras investigaciones, realizar la adquisición de un mayor número de parámetros ambientales, con sensores que cada vez tienen mejor resolución, para realizar un manejo interactivo de parámetros entre sí, además de poder comunicarse con estaciones remotas para control de la vivienda desde lugares distantes.
  
- Se sugiere la utilización del sistema PC/PLC para el control y monitoreo de diferentes procesos de automatización ya que se puede procesar grandes cantidades de información en tiempos

relativamente cortos, realizar la comunicación con otros dispositivos,  
disponer de modularidad, etc