

AUTOMATIZACIÓN DISTRIBUIDA INTELIGENTE¹

Mario Isidro De la Fuente Martínez², Raúl Valente Ramírez Velarde³, Luis Ricardo Salgado Garza⁴

El presente documento trata sobre la implantación en un edificio de una nueva técnica de automatización, que además de poderosa es flexible, ya que esta integrada por módulos inteligentes interconectados. La automatización inteligente por medio de una red de microprocesadores neurales, formalmente llamada Automatización Distribuida Inteligente, es aquella en la que cada microcontrolador es capaz de tomar decisiones independientemente por medio de un programa grabado en EPROM. Adicionalmente los microprocesadores utilizan el protocolo de red LonWorks, que cumple con el modelo OSI, para intercambiar la información sobre el estado del sistema a controlar. Los medios de comunicación usados pueden ser par trenzado, línea de corriente de 60 Hz o radio frecuencia. Los "chips" neurales interactúan con el sistema por medio de simples líneas de entrada y salida conectadas a sensores y actuadores. Las aplicaciones son múltiples: control de iluminación, aire acondicionado, alarmas, accesos, riego automático, etc. Este tipo de redes permite alcanzar el doble objetivo de ahorrar en el consumo de recursos y lograr niveles óptimos de confort. Finalmente, en este documento se hace énfasis en la interconectividad con otros sistemas de control ya establecidos que permiten una mayor flexibilidad en la implementación de la red y proporciona ventajas en reducción de costos al aprovecharse inversiones previas. Las técnicas descritas en este documento nos permiten vislumbrar una era de nuevas aplicaciones tecnológicas, con incidencia directa sobre nuestra forma de vida.

Palabras Clave: Automatización Distribuida, Protocolo de Red, Interconectividad, Microprocesadores neurales.

1. Introducción

Desde sus orígenes, debido principalmente a su capacidad de procesamiento y a la flexibilidad de acción que provee la programación, los microprocesadores han sido utilizados para manipular sistemas físicos. Para facilitar la interconexión y para disminuir los costos, los microprocesadores han sido acompañados por una serie de dispositivos especializados en las tareas sensoras y manipuladoras, tomando así el nombre de microcontroladores. Por medio de distintos circuitos digitales los microcontroladores han sido utilizados para una gran gama de aplicaciones como el control de plantas de producción, el control de dispositivos caseros desde hornos de microondas hasta videocaseteras, control de automóviles y aviones, etc. [1,2]

El principal problema que han tenido hasta ahora los microprocesadores en todas estas tareas ha sido el hacer llegar la información de control que obtienen a las estaciones de

¹ Apareció en la XVII Congreso de Investigación y Extensión del Sistema ITESM (1997)

² Ing. Mario Isidro de la Fuente Martínez: Profesor del Departamento de Ciencias Computacionales del ITESM, Campus Monterrey, mario.delafuente@itesm.mx.

³ Ing. Raúl Valente Ramírez Velarde: Profesor del Departamento de Ciencias Computacionales del ITESM, rramirez@itesm.mx.

⁴ Ing. Luis Ricardo Salgado Garza: Profesor del Departamento de Ciencias Computacionales del ITESM, lsalgado@itesm.mx.

Un microcontrolador que sea capaz de utilizar distintos medios de comunicación, distintas topologías y un protocolo de comunicación confiable reduce considerablemente el tiempo de desarrollo de una aplicación de control ya que, si dentro de su microcódigo se incluyen dichas características, el usuario puede enfocar su esfuerzo a resolver su proyecto de automatización, evitándose la labor de desarrollar el protocolo para el enlace de los módulos; lo que es más, si los microcontroladores pueden intercambiar información, es posible formar una red operativa para el control distribuido de un sistema complejo, de aquí el acrónimo LON ("Local Operative Network") aplicado a este tipo de redes de controladores.

El objetivo de la actual investigación es establecer una nueva área de desarrollo que integre la automatización y los sistemas de información basados en redes locales. Como muestra de las técnicas propuestas se mencionarán las tecnologías diseñadas e implantadas en el proyecto de automatización del Centro de Desarrollo Sostenible del ITESM (CEDES), desarrollado por el departamento de Ciencias Computacionales del ITESM-Campus Monterrey, el cual utiliza tecnología LONWorks para integrar un sistema automático de iluminación, riego e información sobre el sistema de alarmas, de forma que pueda ser consultado y programado de manera remota, utilizando la infraestructura de comunicación del Campus Monterrey. Un esquema general del sistema implantado en el edificio, puede ser visualizado en la figura 1.

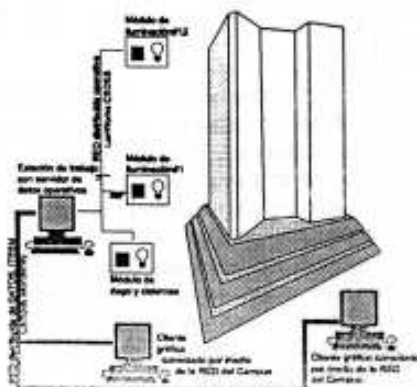


Figura 1. Esquema general de módulos de control para el proyecto de automatización del edificio CEDES.

Este proyecto es único en su género, ya que es el único que utilizando LonWorks basa su administración y control en una plataforma de cómputo que brinde una interfase gráfica amigable y que pueda ser usada en forma remota, mediante una red que utilice el protocolo TCP/IP. Este proyecto contiene las tecnologías que inician un nuevo paradigma en el área de automatización, la cual ya no puede verse desligada de las redes locales ni de las telecomunicaciones.

Mediante las técnicas expuestas en este documento se establece una base de cómo ambas tecnologías son utilizadas como herramientas para consolidar un proyecto de automatización distribuido. Adicionalmente, cabe aclarar que para nuestro caso de estudio (edificio CEDES) se ha contado con la colaboración de alumnos de la carrera de Ingeniero en Sistemas Electrónicos, siendo esta una experiencia integradora de todos los conocimientos adquiridos durante su preparación profesional; de este modo, se puede anticipar como los proyectos de automatización distribuida influirán directamente en la aplicación futura de este tipo de tecnologías para una mejor preparación de nuestros graduados.

2. Metodología.

Para mostrar las etapas y elementos que integran el proceso de la automatización distribuida, procederemos a abordar el caso de estudio del edificio CEDES. Mencionaremos aquí las diferentes etapas del proyecto, analizando específicamente las técnicas de programación y de comunicación entre los nodos diseñadas y empleadas, dejando un tanto al margen los detalles propios de la implantación de cada sistema, tales como los circuitos electrónicos, eléctricos, circuitos integrados utilizados y codificación fuente de los programas desarrollados.

Las etapas del proyecto de automatización abordan aspectos bien delimitados [4] y para nuestro caso contemplaron la especificación de los sistemas operativos que se pretendían automatizar, la identificación de los puntos de control y supervisión, el análisis de sus características, la evaluación de las mejores opciones para interconectar sistemas ambientales a nuestras tarjetas de control, la programación de los algoritmos requeridos, experimentos de laboratorio, pruebas de campo y finalmente la documentación de cada sistema.

Para el desarrollo del proyecto, se decidió utilizar la tecnología LonWorks, de la compañía Echelon, debido a su amplia gama de aplicaciones y versatilidad de adaptación, además de que es el estándar de mayor aceptación para la implementación de sistemas abiertos de control distribuido [5,6]. Ahora, se abordarán los procedimientos seguidos para integrar a nuestra red LonWorks de control los sistemas de iluminación, riego, ruteador de mensajes, interfase con el sistema de alarmas y servidor de datos, interfase gráfica.

El proyecto de automatización del edificio CEDES incorporó a un grupo de profesores y asistentes que integraron equipos de trabajo para cada área mencionada con anterioridad, asimismo en una segunda etapa del proyecto, se invitaron alumnos a participar en el desarrollo e implantación de los sistemas diseñados. Cada equipo, fué responsable de desarrollar un programa que cumpliera con las especificaciones para cada una de sus asignaciones, además de proveer al módulo con un conjunto de variables que, siendo modificadas y consultadas a través de la red, permitieran intercambiar mensajes con los módulos programados por los demás equipos, para ello se utilizaron las variables de red que el protocolo LonWorks provee en cada tarjeta de control (llamadas también Tarjetas Neutrales) y que contienen un microprocesador de propósito general (Neuron Chip), el cual a su vez está formado por tres procesadores en su interior. El primero es el procesador central que se encarga de recibir la información de las líneas de E/S, establecer las estrategias de control y elaborar los mensajes a enviar por la red. A este

procesador se le denomina procesador de aplicación. El segundo procesador recibe los mensajes del procesador de aplicación y se encarga de realizar las conversiones necesarias para encapsular el mensaje en paquetes del protocolo de Red Neural. El tercer procesador se encarga de convertir los paquetes del protocolo en una serie de patrones de bits y de acceder el medio para transmitir dichos patrones[1, 7].

La Red Distribuida Inteligente está constituida de tres componentes: El Hardware, el Software y la Red Neural. Claramente, la Red Neural estará a su vez constituida de hardware y software, no obstante, aquí se menciona aparte debido a que solamente se interconecta, diagnostica y configura la red, pero no se realiza ningún tipo de diseño sobre ella (fuera del diseño topológico). El efecto de la red sobre el sistema neural, sin embargo, es muy importante debido a que afecta directamente al desempeño de la misma y es fuertemente influenciado por la naturaleza del proyecto en cuestión; en el caso del edificio CEDES, la construcción ya había terminado y ya había ocupantes en él, de modo que no se podía instalar una red de comunicación nueva sin afectar la operación normal del edificio. De este modo, se decidió utilizar una red basada en la comunicación por la línea de energía de 110V_{AC}, utilizando las interfases "Power Line" de los módulos de control de la compañía Echelon. Cabe mencionar que la comunicación con el programa residente del servidor de comunicación, implementado en UNIX bajo el sistema NextStep, requirió de enlazar un módulo de interfase tipo Par Trenzado con el resto del sistema, para ello se desarrolló un enrutador que además de filtrar los mensajes los transporta de un medio de comunicación al otro.

Finalmente, se decidió que el hardware de la red neural constara de las Tarjetas Neutrales (microprocesadores y tarjeta madre), interfases de E/S hacia sensores y actuadores (SyA), y los SyA en sí. También, se utilizaron convertidores de señales entre SyA y las interfases de E/S para la tarjeta. Por ejemplo, un nivel de voltaje, que no puede ser leído por una tarjeta neural directamente, por ello se convirtió a un tren de pulsos cuya frecuencia cambiara de acuerdo a la variación del voltaje de entrada.

3. Resultados y Discusión

La red implementada por el departamento de Ciencias Computacionales utilizando tecnología LONWorks se encuentra completamente dentro del modelo ISO [8,9]. La capa de Aplicación se implementa por medio del programa en Neuron C y se graba en EEPROM. Dicho programa es ejecutado por el primer microcontrolador (Microcontrolador de Aplicación). Las capas de presentación hasta Transporte, Red y Enlace se encuentran implementadas por medio de código en "firmware" (figura 2) y es ejecutado por el segundo microcontrolador instalado en la tarjeta Echelon (Microcontrolador de Enlace). Finalmente, el acceso al medio se lleva a cabo por el tercer microcontrolador (Microcontrolador de Acceso al Medio).

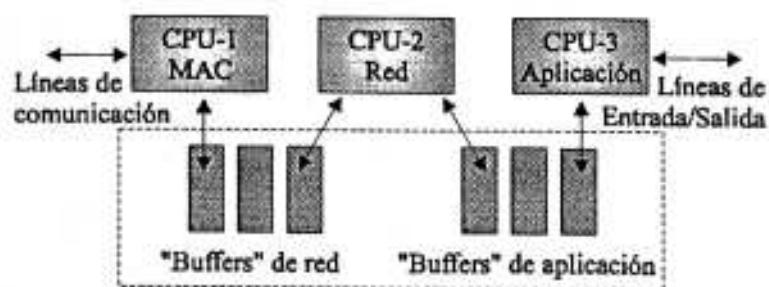


Figura 2. Diagrama lógico de la arquitectura para intercambio de mensajes entre los procesadores del “Neuron Chip”.

Para facilitar la comunicación entre nodos, el protocolo de red neural ha desarrollado una técnica basada en Variables de Red (figura 3). Dichas variables son valores que se propagan a través de la red en forma transparente para el programador, de tal forma que un conjunto de nodos puede realizar labores de control y coordinación por medio ciertas variables que pueden significar tanto valores de medición conocidos a través de todos los nodos involucrados como mensajes de control. Las Variables de Red tienen tipos muy diversos para diferentes aplicaciones y existen varios Tipos de Variable de Red Estándar predefinidos (SNVT) que sirven para comunicar medidas de temperatura, humedad, tiempo, cadenas ASCII, etc.

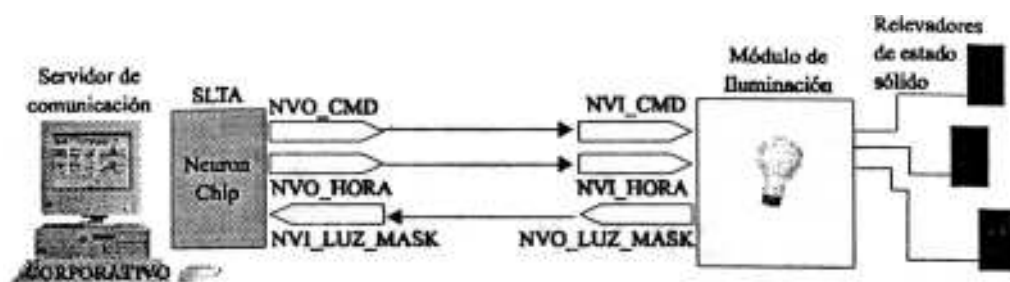


Figura 3. Conexión lógica entre las variables de red de dos módulos en una red LonWorks.

Adicionalmente, para dar mayor flexibilidad a la construcción topológica de la red y reducir el número de mensajes intercambiados, es posible subdividir la red en varios segmentos los cuales se pueden intercomunicar utilizando enrutadores y puentes. Cada segmento puede tener diferente medio de comunicación, lo cual no afecta en lo más mínimo a las aplicaciones (figura 4).

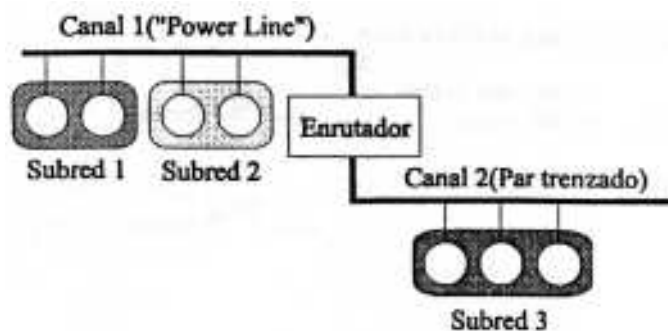


Figura 4. Mapa de la arquitectura de subredes distribuidas en una red LonWorks.

El diagrama anterior de una red LonWorks, muestra un mapa sobre un diseño lógico de una red, pero la interconexión física de los módulos se puede realizar a través de diversos medios de comunicación, tales como par trenzado, línea de energía de 110V_{AC}, radiofrecuencia, y algunos más que posteriormente se mencionará. De esta forma y para el caso de la implementación aquí presentada, se utilizó un enrutador para transportar los datos entre dos medios distintos de comunicación, algo que fue necesario debido a una limitante en las tarjetas de control que resultaron las más apropiadas para este desarrollo.

Inicialmente se programaron los microprocesadores neurales para que intercambian mensajes a través de la red indicando eventos sucedidos en el sistema que supervisan, compartiendo información de estado; para esto se utilizaron variables de red, donde cada una de ellas contiene una especificación de tipo que asigna el significado específico a la información que en ella viaja.

Debido a que la operación de un chip neural es esencialmente llevada por eventos, es necesario modelar el código por medio de un diagrama de estados que especifique todos los posibles estados del módulo y todas las posibles acciones (cambios de estado) de acuerdo a los posibles eventos. Por ejemplo, para un módulo de control de luz, este enciende las luces cuando la hora se encuentra en cierto rango, cuando el nivel de iluminación es bajo o cuando recibe un mensaje de encender inmediatamente. De la misma forma, apaga las luces cuando la hora se encuentra fuera de cierto rango, cuando el nivel de iluminación es alto o cuando recibe un mensaje de apagar la luz. Lo cual parece ser una descripción bastante sencilla; no obstante debemos responder varias preguntas. ¿Cuánto tiempo debe permanecer la luz apagada si se recibe un comando de apagado en medio de la noche? ¿Cuánto tiempo debe permanecer la luz encendida si el sensor de luz indica que existe suficiente luz natural? Analizando los casos obtenemos un diagrama de estados para esta aplicación específica del sistema de iluminación. Finalmente, el diagrama de estados se traduce a una matriz de transiciones-acciones y esta se implementa en el código como una estructura tipo case, tal como puede ser consultado en la figura 5.

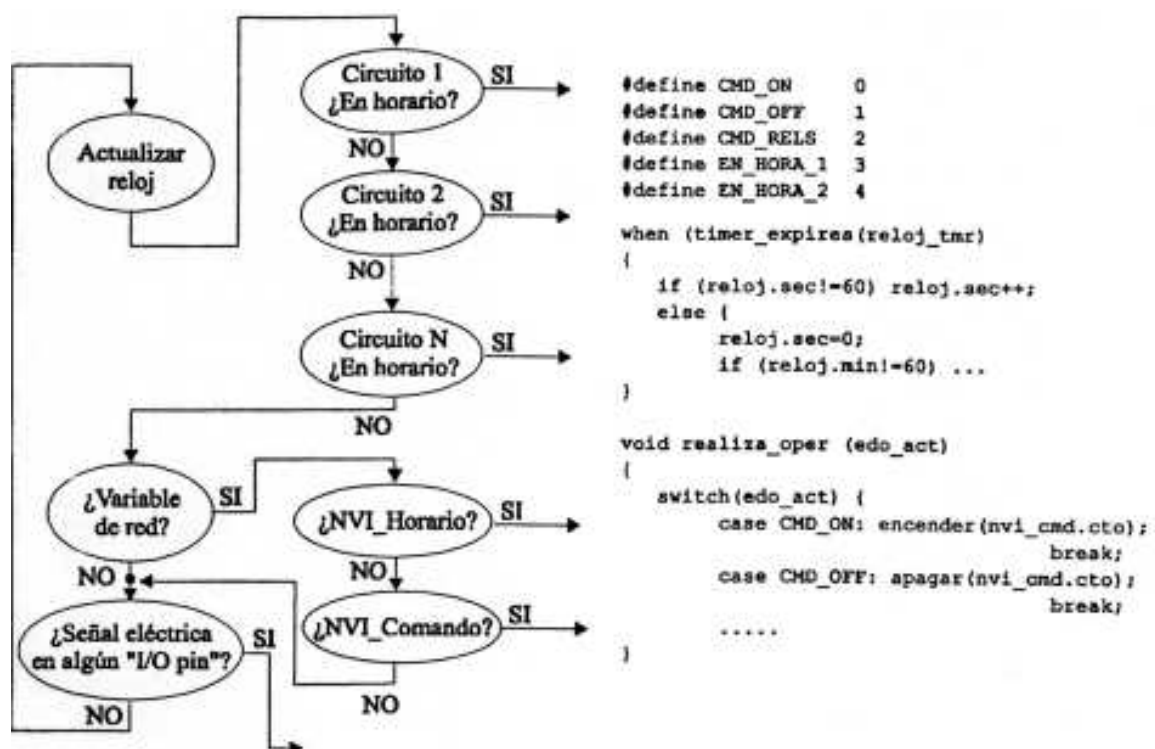


Figura 5. Diagrama de estados parcial y codificación en una estructura tipo "CASE" para el módulo de iluminación.

Se diseñaron diagramas de estados para cada uno de los módulos programados, esta técnica fué seleccionada después de un estudio detallado. En este esquema, cada nodo recorre un diagrama de estados, donde cada uno de ellos involucra la ejecución de una serie determinada

de acciones. Esto significa que las tarjetas neurales pueden realizar tareas con una gran "conciencia" de todo el entorno del sistema. Por ejemplo, cuando se encienden los pisos de un edificio, esto se realiza ya sea porque algún nodo ha "caído" en el estado de encendido por horario, determinando que ya era "hora" de encender las luces y comunicar a los nodos indicados que accionen los relevadores convenientes.

Las interfases de IO de los chips neurales se dividen en dos tipos: directas y temporizadoras. Las interfases directas son: entrada bit, salida bit, entrada byte, salida byte, entrada nibble, salida nibble y detección de nivel. Por otra parte, las interfases temporizadoras son: entrada onetime, entrada periódica, entrada cuadratura, entrada conteo total, entrada conteo de pulso, salida conteo de pulso, salida frecuencia, salida "oneshot", salida amplitud de pulso, salida triac, salida conteo de disparo, salida serial, entrada serial, entrada corrimiento de bits y salida corrimiento de bits.

Algunas de las salidas de bit se utilizaron para activar relevadores y servoválvulas individuales; sin embargo, para los sistemas de iluminación y riego fué necesario utilizar las salidas de corrimiento de bits para controlar el elevado número válvulas de aspersores y de circuitos de iluminación de cada piso, esto debido a la limitante en once salidas digitales del neuron chip.

Para el sistema de riego se utilizaron las entradas de bit para recibir información de tipo encendido/apagado proveniente de algunos sensores; tal como en el caso de la humedad, la cual si ha superado un cierto nivel instruye al sistema de interrumpir el riego. Sin embargo, los procesadores neurales no poseen entradas para medición continua de voltaje, por ello se utilizaron las entradas de conteo, de modo que mediante un convertidor de voltaje a frecuencia se pudiera generar una señal eléctrica compatible con alguna de las entradas predefinidas en el microcontrolador. En este caso, los múltiples y distintos tipos de entrada y salida pueden mezclarse utilizando el multiplexor interno del chip neural como se muestra en la figura 6.

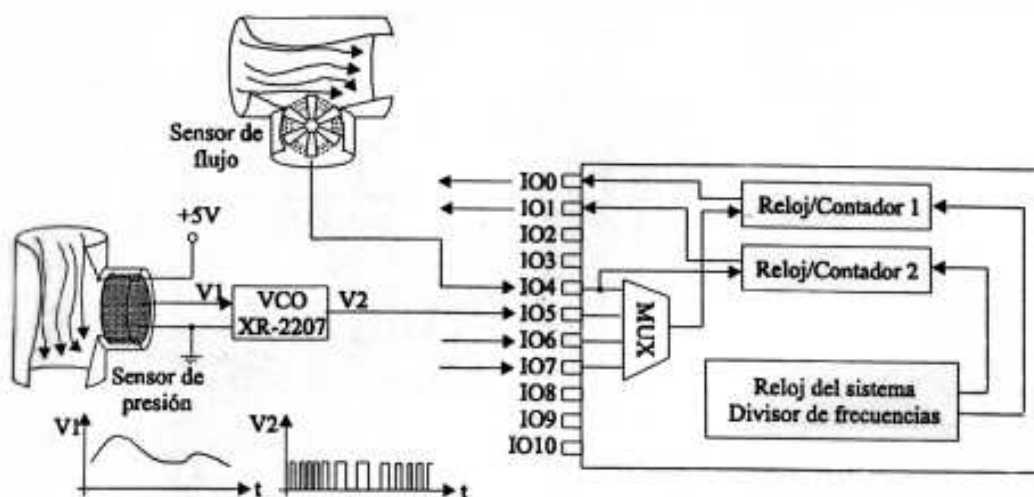


Figura 6. Diagrama explicativo sobre la medición de señales analógicas y su conversión a señales digitales de frecuencia variable.

Los dispositivos de entrada y salida se referencian en la aplicación por medio de variables de entrada/salida (E/S) del lenguaje Neuron C. Las acciones de lectura y escritura sobre dichas interfaces se llevan a cabo por primitivas especiales del lenguaje llamadas IOin y IOout.

El diseño de sensores y actuadores es en sí un tema aparte. En este artículo solo se mencionará que fue importante determinar el tipo de sensor más adecuado para medir la característica deseada, ya sea nivel de profundidad, presión del aire, humedad del aire, humedad en el suelo, nivel de voltaje, nivel de iluminación, consumo de corriente, detección de presencia, contacto entre superficies y otros. Una vez que se hubo determinado el sensor más adecuado, fue necesario examinar la salida que dicho sensor producía para determinar la forma en que debía ser convertida a alguna señal de las que se mencionó en la sección anterior para que pudiera ser interpretada por el chip neural.[10]

También es posible implementar sistemas de riego por humedad, dónde un sensor de detección de humedad activaría el riego de un jardín o de todo un campo sembrado. Este tipo de riego utiliza solo la cantidad de agua necesaria. Si llueve, el riego no se activa. Si existe una sequía, las plantas son regadas sólo con la cantidad de agua necesaria. No obstante, una vez que se activa el riego, la humedad se satura por lo que el sistema tendería a encender el riego y luego apagarlo inmediatamente. Una cuestión importante aquí es identificar la cantidad mínima necesaria de riego continuo que se requiere dado un conjunto de plantas, algo que por el momento está fuera del alcance de este documento.

Una de las técnicas más útiles de entrada/salida diseñadas es la comunicación serial. Este puerto serial sin "buffer" ni interrupciones puede, sin embargo, permitir la interconexión entre diversos sistemas de control. La gran mayoría de los sistemas de control comerciales tienen salidas de impresión de sus computadoras de control. Esto significa que al menos, la red neural puede estar supervisando constantemente el estado de los sistemas que no puede controlar directamente y reaccionar en concordancia. Por ejemplo, para este caso de estudio, se deseaba generar una ayuda adicional para el sistema de alarmas y aprovechando que el sistema ya instalado tiene salida a impresora por medio de un puerto serial, se desarrolló un nodo que permite generar un mapa en la interfase con el usuario, de modo que rápidamente le permita ubicar gráficamente el lugar del problema reportado. El sistema de supervisión inteligente puede interceptar estas salidas (simulando ser una impresora), sin alterar el funcionamiento independiente del sistema de alarmas, e incluso mantener un control constante de la operación del sistema de aire acondicionado; por ejemplo, si el sistema de alarmas detecta un siniestro, el sistema inteligente podría identificar la zona, iluminar los corredores de salida, pedir al sistema de aire acondicionado que interrumpa la circulación de aire en la zona siniestrada (para no alimentar con oxígeno el incendio), presurizar las escaleras con aire (para evitar que el incendio se propague a estas y prevenir intoxicación de las personas que ahí se encuentran), etc. El sistema inteligente podría también comunicarse por medio del puerto serial al equipo de multimedia y suministrarle una serie de mensajes con instrucciones pertinentes para la eventualidad.

Ya se mencionó la importancia del puerto serial para comunicarse con otros sistemas de control. La desventaja principal de utilizar un procesador neural, de la marca Echelon, con puerto serial es la cantidad de variables de red sobre las que se puede mantener control, número que se encuentra limitado a 16. Existe un segundo dispositivo denominado "Serial Lon Talk Adapter" (SLTA) que por medio de un puerto serial de alta velocidad se puede conectar a cualquier ambiente de cómputo. Además de una mejor comunicación serial, el SLTA permite guardar registro de hasta 1024 variables de red; cantidad suficiente para llevar una supervisión de la gran mayoría de las variables importantes en el sistema.

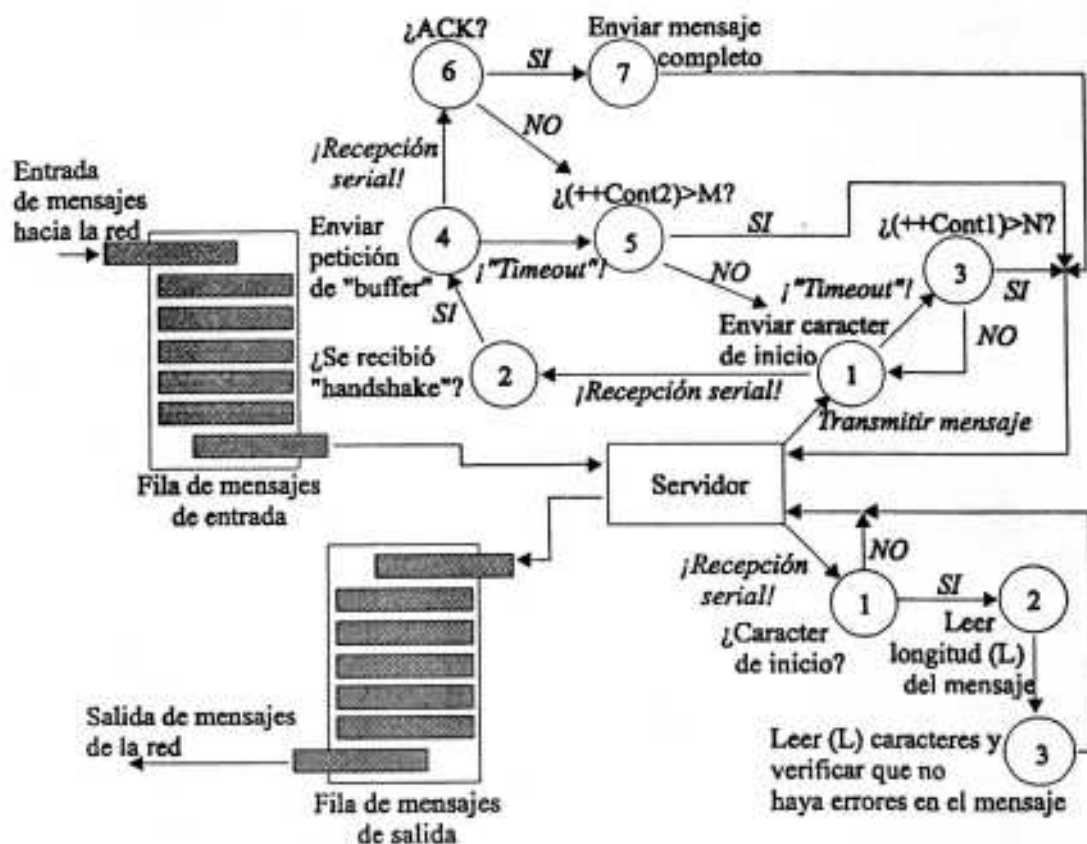


Figura 7. Diagrama de estados principal para el programa servidor de datos del edificio CEDES.

El sistema inteligente es por definición distribuido. Es decir, que cada módulo individual es capaz de supervisar y controlar una serie de dispositivos a los que se encuentra físicamente conectado tomando en cuenta el estado global de la red inteligente. No obstante, el sistema diseñado por el departamento de Ciencias Computacionales contempló la implantación de un Servidor Central en una plataforma de cómputo estándar como NeXTStep. La principal tarea de dicho servidor consiste en llevar un control de todos los eventos que ocurren en la red y mantener una base de datos de dichos eventos con fines de diagnóstico y revisión. Por medio de una técnica avanzada de TCP/IP, denominada Objetos Distribuidos bajo el sistema NeXTStep, el servidor puede recibir consultas sobre la base de datos desde clientes remotos. A través de los mismos paquetes de TCP/IP, pasando por el SLTA, el servidor puede transmitir comandos de los usuarios autorizados que se den de alta en los clientes, a los nodos dentro de la red neural. Los clientes son percibidos por los usuarios por una interfase gráfica de usuario en plataforma NextStep donde el usuario se da de alta utilizando una cuenta y una clave. De esta forma, un operador local o remoto puede utilizar la interfase gráfica para comunicarle al servidor que desea apagar o encender luces, regar jardines, enviar mensajes por medio del sistema de multimedia, en forma manual, etc. Para lograr esto, el servidor actúa como un nodo más dentro de la red, y envía mensajes a los diversos nodos (el servidor central "conoce" a todos los nodos de la red). Se establece entonces un protocolo de peticiones que diversos nodos pueden cumplir.

Algo relevante sobre el servidor es que en realidad es un programa residente en UNIX, basado en plataforma NextStep que a través del SLTA obtiene y registra los valores históricos de todas las variables importantes en el sistema y los almacena en una base de datos, la cual

puede ser consultada para extraer información instantánea sobre la operación del edificio. La figura 7 muestra el diagrama de estados principal para la implantación del servidor.

Hasta el momento, hemos mencionado términos muy sugestivos, como red inteligente y microprocesadores neurales. No obstante, en nuestra discusión, no se ha hablado nada relacionado con la inteligencia artificial, como redes neurales clásicas, algoritmos genéticos, etc. Y no lo haremos. No obstante, mantenemos que los edificios automatizados por medio de una Red Operativa Distribuida tienen un comportamiento inteligente.

Aún no se comprende el funcionamiento exacto del cerebro ni dónde reside la inteligencia. No obstante, sabemos que el cerebro humano está compuesto por millones de células especializadas llamadas neuronas. Ninguna de estas células muestra un comportamiento que individualmente hablando se denomine inteligente. Lo único que sabemos es que a diferencia de otras células del cuerpo humano, está conectadas con cientos de células vecinas e intercambian intensamente mensajes entre ellos. Eso es justamente lo que sucede en un edificio inteligente. Cada nodo neural es individualmente bastante tonto, aunque altamente especializado. No obstante, la inteligencia parece provenir de la colectividad. Cada nodo sabe su papel exacto dentro del concierto y está "conciente" de lo que ocurre a su alrededor. Incluso, es posible incluir paralelos interesantísimos con enfermedades patológicas humanas. La paranoia resulta de la ausencia de un neurotransmisor en ciertas regiones del cerebro. Un sensor defectuoso puede convertir a todo un edificio en paranoico con constantes avisos de intrusos. Un edificio puede quedar ciego. Incluso, los patrones caóticos que causan la epilepsia pueden repetirse con ciertas situaciones eléctricas en las líneas de comunicación. Los paralelos no terminan ahí. Igual que el cerebro, un edificio inteligente tiene áreas de especialización, puede recibir un daño permanente o temporal dependiendo de la naturaleza del daño, y el código de las tarjetas neurales puede modificarse, o pueden agregarse nuevas tarjetas, de tal forma que el edificio inteligente puede "aprender". También puede perder la memoria e incluso morir. Una prueba de Turing modificada para comunicación no verbal sería un ejercicio muy interesante para probar la inteligencia de los edificios inteligentes. Tal vez lleguemos incluso a encontrar temperamentos.

Al menos en la instalación del sistema inteligente del edificio CEDES del ITESM, debido a la presencia de una gran cantidad de personas dentro del edificio utilizando radios de comunicación, y a variaciones constantes en el voltaje de la línea de corriente por las pruebas que se llevaban a cabo en los equipos justo antes de la inauguración, el sistema inteligente tendía a "distraerse" y muchos mensajes importantes se perdían. Sin embargo, estas fallas solo se presentaron durante los primeros minutos de instalación de la red, después de ello los índices de errores de recepción detectados en los módulos de control permanecieron en cero, todo ello gracias a que el protocolo utilizado en las redes LonWorks incorpora técnicas para la recuperación de mensajes en los que se haya detectado errores, así como un algoritmo para el ajuste dinámico de las bandas de frecuencia en los sistemas de transmisión y recepción de datos.

4. Conclusiones

Un Sistema Operador Distribuido es un conjunto de microprocesadores (o tarjetas neurales) cuyo principal fin es la supervisión y el control de un edificio, planta, vehículo, o cualquier sistema físico. Se dice distribuido porque cada microprocesador es capaz de controlar su entorno inmediato, puede supervisar y actuar sobre su parte del sistema de forma autónoma. No obstante, cada tarjeta neural se intercomunica con el resto por medio de una red llamarla Red Neural y un protocolo de comunicaciones. La existencia de dicha red es de crucial importancia porque permite a la tarjeta neural tener un conocimiento de lo que sucede en todo el sistema. De

este conocimiento, de la naturaleza distribuirla de entorno, y de las funciones que realiza el sistema operador, emerge un comportamiento que aparenta cierta inteligencia, aunque ningún componente individual sea inteligente por sí mismo.

La flexibilidad de los microcontroladores neurales y la gran cantidad de interfases de E/S permite una gran variedad de aplicaciones: como control de iluminación, control de riego, control de accesos, supervisión de consumo de corriente, alarmas y aire acondicionado, entre otras. [5, 6]

La naturaleza abierta de la red operadora, permite la interconexión con otros sistemas, con los consecuentes beneficios de poder aumentar la funcionalidad de un sistema ya instalado, o realizar la instalación de un sistema inteligente escogiendo la mejor opción para el tipo de control.

La utilización de un servidor interconectado por medio de un SLTA, con capacidad de guardar registros en una base de datos, y de recibir consultas de SQL por medio de "sockets" de TCP/IP permite mantener un sistema de monitoreo gráfico y amigable remoto. Los clientes de TCP/IP permiten mantener un sistema de seguridad al mismo tiempo que da flexibilidad en la operación del edificio.

Finalmente, la tecnología de edificios inteligentes y toda la gran variedad de aplicaciones, permite prever que en un futuro muy cercano, la sociedad en México, será testigo de una revolución tecnológica sin precedentes que cambiará, para bien, nuestras vidas.

5. Bibliografía

- [1] Reji, R.S; "Smart networks for control", IEEE Spectrum 18:49-55 (1994).
- [2] Health, S.; "Echelon networking control", IEEE Review: 363-367 (1992).
- [3] Reizenman, M.J. y Dooling, D.; "The Twin Towers of Kuala Lumpur", IEEE Spectrum 18:44-47 (1995).
- [4] Vandoren, V.; "The Anatomy of An Automation Project", Control Engineering:6-7 (1996).
- [5] Hoske, M.; "LonWorks Expands, Positions as Universal Network Solution", Control Engineering:77- 78 (1996).
- [6] Blomseth, R.; "Developing Home Automation Devices with LonWorks", Home automation & building control:51-58 (1995).
- [7] Echelon Corp.; Neuron C: User's Manual; Echelon Corp., 1st Edition (1992).
- [8] Stallings, W.; Data and Computer Communications; Prentice Hall, 4th Edition(1995).
- [9] Tanenbaum, A.; Computer Networks; Prentice Hall, 2nd Edition (1992).
- [10] Doebelin, E.; Measurement Systems; McGraw Hill, 1st Edition (19.i6).