Diseño e Implementación de una Tarjeta de Monitoreo y Control en forma remota a través de Internet, utilizando la tecnología del microcontrolador PIC18F97J60

Efrén Montenegro Viera
Eduardo Sandoya Tinoco
Ronald Ponguillo Intriago

Egresados, Facultad de Ingeniería en Electricidad y Computación (FIEC), Escuela Superior Politécnica del Litoral (ESPOL)
Director de Tesis, Ingeniero en Electricidad especialización Electrónica, Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km. 30.5 Vía Perimetral, Apartado postal 09-01-5863, Guayaquil, Ecuador
rponguil@espol.edu.ec, emontene@espol.edu.ec, esandoya@espol.edu.ec

Resumen

El trabajo presentado en este artículo fue desarrollado para demostrar la aplicación de los recursos tecnológicos de dispositivos como el Microcontrolador PIC 18F97J60 en la Domótica.

Existen diferentes maneras de implementar la domótica, motivo por el cual, hemos decidido optar por una nueva tecnología como lo es la de los PICs con sistemas embebidos para el desarrollo de una aplicación que reduzca en costo y tamaño lo que actualmente podemos encontrar en el mercado.

Con la ayuda de diferentes herramientas de Microchip, desarrollamos un software que nos permite el monitoreo y control de las luces de una vivienda, recalcando que con las respectivas modificaciones, el mismo podrá ser utilizado para diferentes aplicaciones de control eléctrico. La aplicación consta de una tarjeta de control, en la que está nuestro principal componente, el PIC18F97J60, con módulo Ethernet embebido programado con el software MPLAB IDE utilizando el programador serial ICD2, todos ellos fabricados por la empresa Arizona Microchip Inc.

A más de nuestra tarjeta de control, construimos una tarjeta para el manejo de la etapa de potencia con lo que podemos mostrar completamente el control de dispositivos eléctricos dentro de las viviendas.

Palabras Claves: Ethernet, TCP/IP, Microcontrolador

Abstract

The work presented in this paper was developed for to demonstrate the application of technologies like the Microcontroller PIC18F97J60 in domotic.

There are different ways to implement domotic, so we have decided to choose a new technology like the PICs with embedded systems for the development of an application that reduce the cost and size of products that nowadays we can find in the market

With the *Microchip's* help and its different elements, we developed software that permit us the control of the lights in a house, emphasizing that with some modifications, the same system can be used for different electrical applications. The application has a control board, in this board is our principal component, the PIC18F97J60 with embedded Ethernet module and programmed with MPLAB IDE software and using the ICD2 Serial Programmer, both provided for Microchip.

Also we have made a power board that connected in a subsequent stage by to drive the voltage levels needed to control all devices in a house.

Key Words: Ethernet, TCP/IP, Microcontroller

1. Introducción

El concepto de domótica nace a finales de los años 70 en EEUU, gracias al auge de las telecomunicaciones y una amplia evolución de edificios llamados "inteligentes" debido a la automatización para el control de las actividades del hogar.

Obtenemos como resultado de esta evolución, la integración de los sistemas de gestión de seguridad, comunicaciones, gestión del confort y control energético. (ver Figura 1)

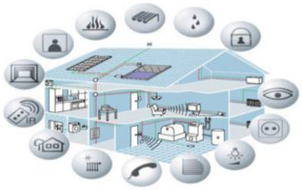


Figura 1. Integración de Sistemas en Vivienda Inteligente

En cuanto a los dispositivos podemos citar que cualquier sistema domótico se basa en un controlador, un actuador, un sensor, un bus y una interface; en nuestro caso, tenemos el microcontrolador PIC18F97J60 como controlador, un sistema de relés como actuador, la lógica del sensor se encontrará en la misma programación del PIC, el bus será una red de datos y como interface tendremos una página web de fácil manejo para el usuario.

El sistema utilizado será un sistema de control punto a punto con estructura de red en estrella, pudiendo nuestra tarjeta principal manejar varias aplicaciones a la vez con ciertas modificaciones que no serán implementadas en este proyecto. El sistema punto a punto genera una arquitectura de red centralizada ya que envía y recibe toda la información necesaria a los objetivos interconectados.

Nuestro medio de transmisión es un cable de red UTP categoría 5, que se conectará a un punto de Internet de una vivienda, mediante el cual controlaremos remotamente las luces de la misma, es decir que utilizamos un cableado ya existente con protocolos ya definidos.

2. Tecnología Ethernet

El protocolo que rige nuestro trabajo es el conjunto de protocolos TCP/IP, que nos define una serie de reglas y primitivas que permiten a una red intercambiar información, sean estas locales o de área extensa.

Este protocolo se puede definir en cinco niveles o capas, que son:

Nivel Físico
Nivel de Enlace
Nivel de Red
Nivel de Transporte
Nivel de Aplicación

Cada uno de los niveles o capas nos permiten en conjunto trocear las tramas de datos de las aplicaciones y encapsularlos en pedacitos de longitud finita, con una dirección origen y una dirección destino.

Si hablamos del protocolo TCP/IP no podemos dejar de hablar de la tecnología Ethernet, que en sí es una familia de tecnologías de redes de computadoras de área local (LAN) del tipo de Acceso Múltiple con Detección de Portadora y Detección de Colisiones CSMA/CD, que cumplen con las especificaciones Ethernet, incluyendo IEEE 802.3; basada en tramas de datos.

Las estaciones en una LAN CSMA/CD pueden acceder a la red en cualquier momento y, antes de enviar los datos, las estaciones CSMA/CD escuchan la red para ver si ya es operativa. Si lo está, la estación que desea transmitir espera. Si la red no está en uso, la estación transmite. Decimos que se produce una colisión cuando dos estaciones que escuchan el tráfico en la red no oyen nada y transmiten simultáneamente. En este caso, ambas transmisiones quedan desbaratadas y las estaciones deben transmitir de nuevo en otro momento. Los algoritmos Backoff son los que determinan cuando deben retransmitir las estaciones que han colisionado. Ethernet está bien adaptada a las aplicaciones en que el soporte de comunicaciones local a menudo tiene que procesar un elevado tráfico con puntas elevadas de intercambio de datos.

Los estándares Ethernet no necesitan especificar todos los aspectos y funciones necesarios en un Sistema Operativo de Red NOS. Como ocurre con otros estándares de red, la especificación Ethernet se refiere solamente a las dos primeras capas del modelo de referencia OSI. Estas son la capa física y la de enlace; que proporcionan direccionamiento local, detección de errores, control del acceso a la capa física, entre otras.

La aparición del modelo OSI en el año 1984 dio la pauta para una mayor compatibilidad e inter-operabilidad entre los distintos tipos de tecnología de red producidos por las empresas a nivel mundial. Este modelo constaba de siete capas, cada una con sus respectivas unidades de datos de protocolo (PDU). (ver Figura 2)

Figura 2. Representación gráfica de las capas del modelo OSI

El modelo de referencia OSI se ha convertido en el modelo principal para las comunicaciones por red.

3. El Microcontrolador PIC 18F97J60

El microcontrolador PIC 18F97J60 que utilizamos en nuestro proyecto fue diseñado especialmente para el uso de Ethernet; en él se integró el controlador y el periférico de Ethernet 10BaseT (IEEE 802.3).

Además nos ofrece una serie de módulos embebidos que nos permitieron desarrollar con mayor facilidad nuestro trabajo.

Como ya indicamos, el módulo principal de nuestro controlador es el módulo Ethernet que incluye implementaciones completas de módulos de Control de Acceso al Medio (MAC) y de la Capa Física (PHY).

A su vez podemos subdividir el módulo Ethernet en cinco bloques de funcionalidad específica que son:

El módulo de Transmisión y Recepción de la Capa Física.

El módulo MAC.

Un almacenador RAM de 8 Kbytes.

Un controlador de acceso al almacenador RAM.

Una interface de registro.

Otro de los módulos importantes de nuestro PIC es el EUSART, que es uno de los dos módulos de entradas y salidas seriales; este puede ser configurado como un sistema full-duplex asincrónico que puede comunicarse con diferentes dispositivos periféricos. También puede ser configurado, al igual que lo hicimos en nuestro proyecto, como un sistema half-duplex sincrónico que así mismo se comunicará con diferentes dispositivos periféricos como EEPROMs, circuitos integrados análogos digitales, etc.

En la programación del microcontrolador hemos utilizado el software MPLAB IDE, proveído por Microchip ya que contiene todos los componentes necesarios para desarrollar y desplegar aplicaciones para sistemas embebidos.

MPLAB tiene componentes llamados compiladores, depuradores y simuladores para todos los microcontroladores Microchip que ayudan a probar el código programado.

El compilador que utilizamos fue el MPLAB C18 que toma el estándar de lenguaje de programación C y posteriormente lo traduce a código binario de máquina. El código es escrito utilizando la notación del estándar ANSI C. El código fuente es compilado en bloques de código y datos que posteriormente son enlazados con otros bloques similares para ser luego ubicados en diferentes regiones de memoria del microcontrolador.

Este proceso es llamado una “construcción”, y es a menudo ejecutado a medida que el código de programa es escrito, probado y depurado.

Una vez revisado el código por MPLAB IDE con sus respectivas herramientas, pasamos a programar el microcontrolador. Para este fin, utilizamos el Programador MPLAB ICD 2 de Microchip que es conectado a la tarjeta con el cable de interfaz modular, que es un conector de seis pines, de los cuales se utilizan únicamente cinco.

El programador además de grabar el código en la memoria del PIC, puede ser utilizado como depurador, previa programación del dispositivo; y se conectará al computador a través de un Bus Serial Universal (USB) que además de transmitir la información hacia el mismo, lo alimentará con el voltaje requerido.

4. Diseño y Desarrollo del Sistema

En el desarrollo del sistema podemos citar cuatro partes importantes que son: el programa del microcontrolador, la tarjeta de control, la tarjeta de potencia y la aplicación web; cada uno de ellos tiene vital importancia en el funcionamiento del proyecto. (ver Figura 3)

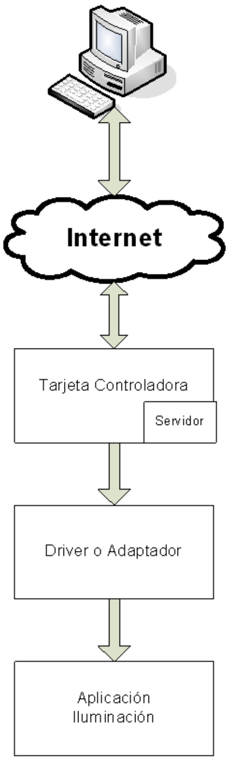


Figura 3. Diagrama de Bloques del Sistema

El programa o software del microcontrolador nos permitirá utilizar y manipular las diferentes características físicas, y lógicas del mismo mediante instrucciones ya definidas por el fabricante. Además de darle el funcionamiento requerido para el cual ha sido diseñado nuestro proyecto.

Para la programación del Microcontrolador nos hemos apoyado en el Stack TCP/IP 4.11, que facilita el fabricante de forma libre. Para desarrollar el programa de una manera eficiente, hemos dividido el código en funciones que serán utilizadas para el adecuado funcionamiento de la aplicación.

Las funciones que permiten ejecutar tareas especificas de la aplicación (Inicializar la tarjeta de control, mostrar el valor IP, actualización del tiempo, etc), son: Inicializar_Tarjeta, Inicializar_AppConfig, Grabar_-

AppConfig, Seteo_Config, Formato_NetBIOS_-Nombre, Mostrar_Valor_IP, UART2TCP-BridgeISR y TickUpdate, etc. Además tenemos otro conjunto de funciones propias del stack TCP/IP(Inicializar el stack, reiniciar la tarjeta de control vía web, telnet, etc), estas funciones son: StackTask, UART2TCPBridgeTask, HTTPServer, DiscoveryTask, NBNSTask, DHCP-ServerTask, GenericTCPServer, TelnetTask, Reboo-Task, SNTPClient, etc.

Con el objetivo de facilitar la programación, hemos visto conveniente el uso de estructuras como una forma de organizar los datos que elementales. Entre las principales estructuras datos tenemos: La PHY para la capa física, así y las estructuras DHCP, HTTP, IP, TCP, TFTP, UDP, utilizadas en los procesos de transmisión y recepción. Además tenemos estructuras para las unidades y denominaciones utilizadas en la transmisión de datos tales como: BOOL, BYTE, WORD, DWORD, QWORD, etc.

Debido a que el programa necesita una serie de funciones que se llaman de manera automática unas a otras, para cumplir con una tarea específica, hemos utilizado cabeceras o archivos '.h'. Las cabeceras contienen única y exclusivamente los prototipos de las funciones y en algunos casos, contienen las estructuras de datos que ya hemos mencionado.

Las principales cabeceras que utilizamos son: ARP.h, Delay.h, DHCP.h, ETH97J60.h, PIC18F97J60.h, FTP.h, funciones.h, GenericTCPServer.h, HTTP2.h, ICMP.h, IP.h, MAC.h, MPFS.h, Random.h, SNTP.h, StackTSk.h, TCP.h, TCPIP.h, TCPPerformance.h, Telnet.h, TPTPc.h, Tick.h, UART2TCPBridge.h, UDP.h.

Con el código realizamos la configuración de los registros del PIC18F97J60 y al stack TCP/IP, permitiéndonos hacer las inicializaciones respectivas en el PIC. Con esto aseguramos un buen desempeño del sistema, tanto en el proceso interno del microcontrolador, como en las diferentes interfaces que se manejan. Las funciones que permiten llamar, y utilizar las distintas funciones, y estructuras del stack, ayudan a que la tarjeta se pueda comunicar con la red externa o con una PC conectada directamente.

La tarjeta de control nos permite dominar la tarjeta de potencia, por medio de instrucciones que da el usuario a través de una interfaz web definida y guardada en el microcontrolador.

Nos permite también la conectividad a la red de Internet, a través del conjunto de protocolos del modelo TCP/IP que estarán manejados por el módulo Ethernet de nuestro Microcontrolador, el cual tiene entradas y salidas específicas para el propósito de transmisión y recepción de datos; estas a su vez pasan por un circuito de transformadores que sirven de acoplamiento entre el Microcontrolador y la red externa.

Cabe recalcar, que este conjunto de transformadores está integrado junto a un conector RJ45 hembra en un solo elemento, que además posee dos leds, uno al lado derecho y otro al lado izquierdo del elemento, que nos indican la actividad y el proceso de enlace respectivamente. Esto muestra si una aplicación de Ethernet está transmitiendo o recibiendo un paquete y si la conexión está activa.

Para la programación de la tarjeta utilizaremos un conector RJ 11 hembra que servirá de interfaz entre el PIC y el programador MPLAB ICD 2, del que ya hablamos en el capítulo anterior.

Hemos implementado un puerto RS232 con un conector DB9, con el fin de facilitar el monitoreo la configuración de parámetros de red como nombre de cliente, dirección IP, máscara de subred, número de puerta de enlace y permitir o no el control de direccionamiento dinámico del PIC.

Nuestro diseño se alimenta directamente de una línea residencial de 120 V. a 60 Hz, por medio de un adaptador a 9 V. DC. obtenemos el potencial necesario para el óptimo funcionamiento de nuestros reguladores, uno a 3.3 V. DC. y el otro a 5 V. DC que nos darán el voltaje necesario que requieren los diferentes elementos que conforman nuestra tarjeta.

Observamos en un LCD la configuración actual de la dirección IP, y algún otro parámetro o frase que implementemos en la programación.

La tarjeta de potencia que utilizamos encenderá las luminarias de una vivienda; recibe las instrucciones de la tarjeta de control a través de tres conectores macho de 16 pines que son unidos por medio de cables planos. Se incluye en nuestro proyecto varios pines de entrada y/o salida para darle escalabilidad.

En la elaboración de la tarjeta de control utilizamos varios elementos que cumplen algunas funciones conforme lo detallamos a continuación.

El Microcontrolador PIC18F97J60: Es la parte principal de nuestro diseño; tiene la particularidad de que incluye un módulo de Ethernet embebido a más de diferentes entradas y salidas. Además cuenta con diferentes módulos utilizados para configurar parámetros de oscilación, memoria, comunicación serial, etc.

La memoria EEPROM 25LC256: Esta nos da la capacidad de almacenamiento al proveernos de 256 Kbits de memoria, la cual utilizaremos para guardar las páginas web de monitoreo y control de nuestra aplicación.

El oscilador: Mediante un cristal de 25 MHz damos sincronismo al PIC, para la transmisión y recepción de datos del módulo Ethernet.

El LCD: Sus dos líneas de 16 caracteres nos mostrará la configuración de la dirección IP y algún otro mensaje que necesitemos el mismo que será ingresado mediante programación del PIC.

Las botoneras: Nos ayudan en diferentes funciones de nuestra aplicación, como es el caso de inicialización del PIC a través de la señal MCLR y habilitar la comunicación serial a través del RS232.

Conector Modular RJ45: Permite la conectividad física con la red, el módulo nos provee de un conector hembra modular, transformadores integrados para acoplar magnéticamente la entrada y eliminar cualquier tipo de corriente inducida; además, posee dos leds que nos indican la actividad y enlace respectivamente.

Conector Modular RJ11: Nos permite la comunicación y programación del PIC a través del MPLAB ICD 2 que utilizaremos para nuestro proyecto.

Puerto Serial: Tenemos un puerto RS232 con un conector DB9 y el respectivo desplazador de nivel MAX3232 que nos permite configurar la dirección IP a través de una comunicación serial estándar.

Reguladores: Dos reguladores, de 3.3v. y 5 v. utilizados para energizar el PIC y los demás elementos de la tarjeta.

Conectores de salida: Hemos utilizado tres conectores de 16 pines que dan escalabilidad a nuestro proyecto, estos nos permiten llevar las salidas del PIC a otras aplicaciones o a su vez importar señales de entradas de las mismas.

Capacitores y Resistencias: Los capacitadores y resistencias utilizadas en nuestro proyecto nos sirven en diferentes casos como protección, filtros, divisores de voltaje, etc.

La tarjeta de potencia permite la adaptación de voltajes entre la tarjeta de control y el circuito de luces, de tal manera que las señales digitales enviadas por el microcontrolador se amplifiquen para obtener el potencial necesario y encender o apagar las luminarias. (ver Figura 4)

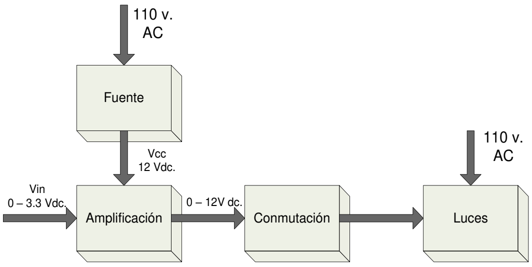


Figura 4. Diagrama de Bloques de Tarjeta de Potencia

Empezamos por la etapa de alimentación que consta de una fuente, la cual será energizada con 120 voltios a 60 Hz por medio de una bornera y su respectivo fusible de protección; mediante un transformador reducimos el voltaje a 12 voltios y lo rectificamos mediante un puente de diodos para convertirlo en DC.

Se utilizan también, dos capacitores y un diodo adicional para eliminar el rizado y proteger el circuito respectivamente.

Este nivel de voltaje de 12 V. DC. es utilizado para polarizar un integrado que hace las veces de amplificador.

En la siguiente etapa de la tarjeta de potencia encontramos el circuito integrado ULN2803 que sirve para amplificar el nivel de 3.3 voltios DC provenientes de la tarjeta controladora a través de un cable plano de 16 pines en 12 voltios DC que activan cuatro relés en la siguiente etapa.

La etapa final de nuestra tarjeta amplificadora consta de cuatro relés que permiten encender o apagar las luces de la aplicación final. Al juntarse los contactos de los relés, estos cerrarán el circuito de cada una de las luces y harán que estas se enciendan o se apaguen según sea el caso.

La aplicación web de fácil manejo, nos permitirá crear una interfaz gráfica entre el usuario y la tarjeta de control. En el cual la persona podrá visualizar el estado de las luminarias y manipularlas según su conveniencia.

El diseño de la página web del Microcontrolador es grabada en una EEPROM, se lo hizo en Dreamweaver versión 8.0 y consta de las siguientes partes:

Identificación clara del Logotipo de la Facultad, de la ESPOL: Se importa las imágenes de las páginas autorizadas de la universidad que se utilizan en dicho propósito y se ubican en la carpeta de imágenes de dónde son importadas para mostrar en la aplicación final.

Nombre y dirección electrónica de los Integrantes: Se incluyen los nombres de los integrantes del Proyecto de Tesis en el diseño web.

Título y Objetivos del Proyecto: Debido a que es una aplicación académica, hemos decidido nombrar los objetivos del proyecto para verificar su cumplimiento y el buen funcionamiento del mismo.

Hipervínculos: Al utilizar dos páginas web, hacemos un hipervínculo entre ellas para pasar de una a otra sin ningún problema.

Estado de Luces: En la primera página mostramos el estado de las luces; en esta sección también tenemos la facilidad de interactuar con ellas encendiéndolas o apagándolas según sea el caso; se han implementado cuatro luces con su respectiva ventana para elegir el estado de la misma.

Para demostrar el funcionamiento de la comunicación por la red de Internet, conectamos la tarjeta de control al punto de red más cercano y esta a su vez está conectada a la tarjeta de potencia que da la energía necesaria para el encendido y apagado de luminarias de nuestra aplicación.

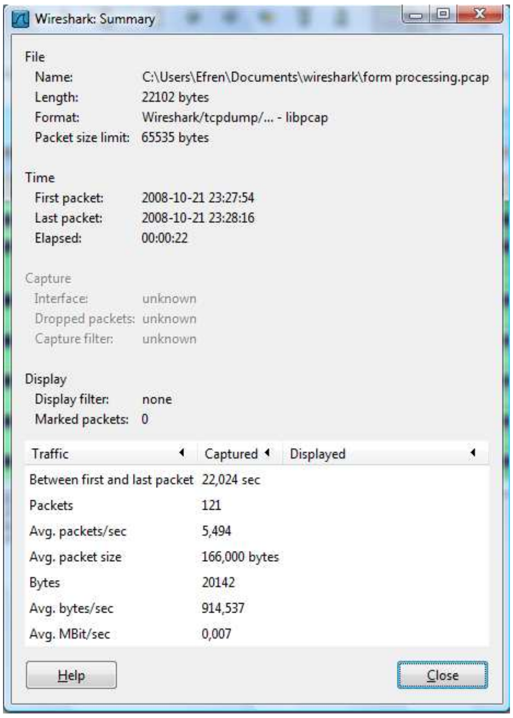
También necesitamos conectar una computadora personal a otro punto de red distinto, mediante la cual accedemos a la aplicación web que nos ayuda a monitorear y controlar remotamente la aplicación final.

5. Análisis de Resultados

Dividimos los resultados en transmisión y recepción, y con la ayuda del Software Wireshark hemos llegado a lo siguiente:

Para la transmisión el tiempo entre el primer y último paquete capturado es 22,02 segundos, el número de paquetes capturados, 121 y su respectivo promedio por segundo es de 5,49 paquetes/seg.

Tenemos también el promedio de tamaño de los paquetes en base a todos los capturados que es de 166 bytes; el número de bytes total transmitidos 20.142 y sus promedios en diferentes unidades, 914,53 bytes/seg y 0,007 MBit/seg.



File
Name: C:\Users\Efren\Documents\wireshark\form processing.pcap
Length: 22102 bytes
Format: Wireshark/tcpdump/... - libpcap
Packet size limit: 65535 bytes

Time
First packet: 2008-10-21 23:27:54
Last packet: 2008-10-21 23:28:16
Elapsed: 00:00:22

Capture
Interface: unknown
Dropped packets: unknown
Capture filter: unknown

Display
Display filter: none
Marked packets: 0

Traffic	Captured	Displayed
Between first and last packet	22,024 sec	
Packets	121	
Avg. packets/sec	5,494	
Avg. packet size	166,000 bytes	
Bytes	20142	
Avg. bytes/sec	914,537	
Avg. MBit/sec	0,007	

Figura 5. Parámetros de Transmisión

Por último, destacamos el gráfico generado por el mismo software en el que podemos observar: en el eje Y el número de paquetes transmitidos y en el eje X: el tiempo en segundos durante el cual fueron transmitidos.

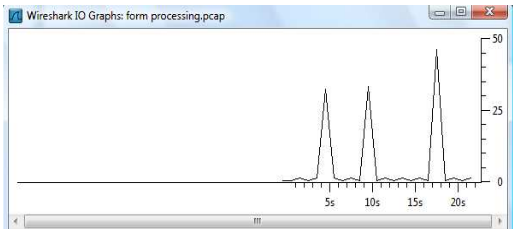
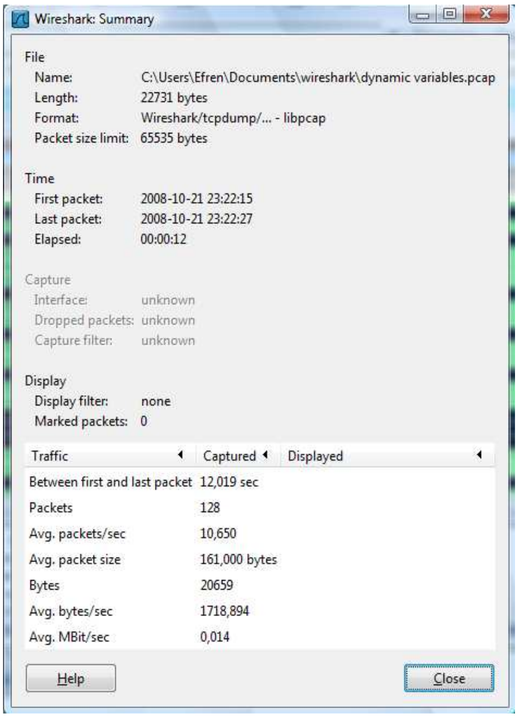


Figura 6. Paquetes vs. Tiempo de Transmisión

Para la recepción el tiempo entre el primer y último paquete capturado es 12,019 segundos, el número de paquetes capturados, 128 y su respectivo promedio por segundo es de 10,650 paquetes/seg.
Tenemos también el promedio de tamaño de los paquetes en base a todos los capturados que es de 161 bytes; el número de bytes total transmitidos, 20.659 y sus promedios en diferentes unidades, 1.718,894 bytes/seg y 0,014 MBit/seg.



File
Name: C:\Users\Efren\Documents\wireshark\dynamic variables.pcap
Length: 22731 bytes
Format: Wireshark/tcpdump/... - libpcap
Packet size limit: 65535 bytes

Time
First packet: 2008-10-21 23:22:15
Last packet: 2008-10-21 23:22:27
Elapsed: 00:00:12

Capture
Interface: unknown
Dropped packets: unknown
Capture filter: unknown

Display
Display filter: none
Marked packets: 0

Traffic	Captured	Displayed
Between first and last packet	12,019 sec	
Packets	128	
Avg. packets/sec	10,650	
Avg. packet size	161,000 bytes	
Bytes	20659	
Avg. bytes/sec	1718,894	
Avg. MBit/sec	0,014	

Figura 7. Parámetros de Recepción

Por último, destacamos el gráfico generado por el mismo software en el que podemos observar lo siguiente: en el eje Y el número de paquetes recibidos y en el eje x el tiempo en segundos durante el cual fueron recibidos.

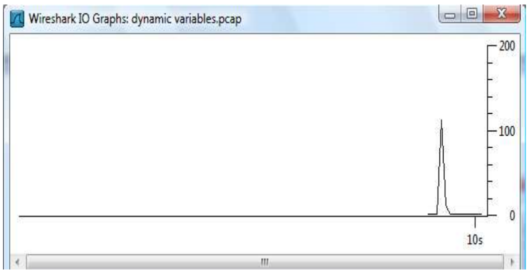
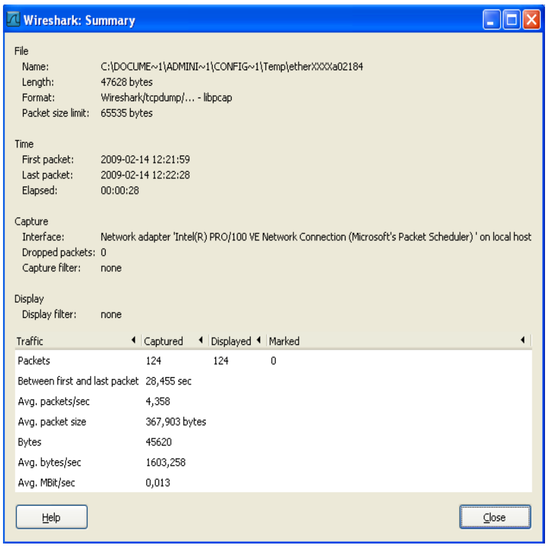


Figura 8. Paquetes vs. Tiempo de Recepción

Si realizamos las mismas mediciones con el mismo software en una tarjeta de red Intel® PRO/100 VE Network Connection, obtenemos la siguiente tabla de resultados:



```
Wireshark: Summary

File
  Name:               C:\DOCUME~1\ADMINI~1\CONFIG~1\Temp\etherXXXXa02184
  Length:             47628 bytes
  Format:             Wireshark/tcpdump/... - libpcap
  Packet size limit:  65535 bytes

Time
  First packet:       2009-02-14 12:21:59
  Last packet:        2009-02-14 12:22:28
  Elapsed:            00:00:28

Capture
  Interface:          Network adapter 'Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler) ' on local host
  Dropped packets:    0
  Capture filter:     none

Display
  Display filter:     none

Traffic                       Captured        Displayed   Marked
Packets                       124             124         0
Between first and last packet 28,455 sec
Avg. packets/sec              4,358
Avg. packet size              367,903 bytes
Bytes                         45620
Avg. bytes/sec                1603,258
Avg. MBit/sec                 0,013
```

Figura 9. Parámetros de Transmisión en Tarjeta de Red Intel® PRO/100 VE Network Connection

Como podemos observar tenemos un promedio de transmisión de 4,358 paquetes/segundo con paquetes con un promedio de 367,903 bytes; lo que finalmente nos da un promedio de 1603,258 bytes/segundo y 0,013 Mbit/segundo; resultados que coinciden con los parámetros de transmisión y recepción de nuestra tarjeta controladora, teniendo en cuenta pequeñas variaciones que dependen del tamaño de página web utilizado en nuestro proyecto con el tamaño de las páginas que encontramos actualmente en la web; además debemos considerar que el microcontrolador utilizado posee una tasa de transferencia de 10 Mbit/seg. y la tarjeta de red utilizada en la computadora es de 10/100 Mbit/seg.

6. Conclusiones

La utilización de los protocolos TCP/IP dan los resultados esperados y son la base en la elaboración del código fuente, especialmente en la etapa de conexión a Internet del módulo embebido Ethernet que posee el microcontrolador que utilizamos; además, se pudo observar el trabajo realizado a través de cada una de las cinco capas del modelo TCP/IP como lo demuestra el código.

El PIC18F97J60 es el indicado para aplicaciones de domótica, ya que su costo no es muy elevado y por medio de una simple lógica de programación nos da la facilidad de desarrollar diferentes tipos de automatizaciones en el hogar o empresa para un mayor control de la misma.

A pesar de que usamos el programa Dreamweaver versión 8.0, es imprescindible indicar que puede usarse incluso un editor de textos tan simple como el notepad de Windows, ya que una página web está construida exclusivamente de texto plano que luego es transmitido vía http utilizando la interface Ethernet del microcontrolador.

Normalmente la interface de fuerza, que es la que maneja los voltajes y corrientes de niveles mucho mayores a los que maneja el microcontrolador suele ser bastante tediosa de resolver debido a que hay que hacer un acoplamiento de niveles. Esto lo hemos manejado con el circuito integrado ULN2803AG provisto por TOSHIBA, con lo que no solo superamos lo engorroso del asunto sino que logramos que la etapa de fuerza sea muy pequeña en comparación con los esquemas tradicionales de manejo de relés hechos a base de transistores y adicional a eso permite disminuir el consumo de potencia.

Cuando hicimos la comparación de velocidades entre la tarjeta que diseñamos con el microcontrolador PIC18F97J60 y la tarjeta de red 10/100 Intel® PRO/100 VE Network Connection, pudimos verificar que esa característica era muy similar en ambas, lo cual nos deja muy satisfechos, ya que nuestro objetivo cuando empezamos este proyecto fue lograr un producto que sirva para el control remoto de actividades en el hogar con características que puedan competir en el mercado.

7. Referencias

[1] Sistemas de Control para Viviendas y Edificios: Domótica, José Mª Quinterito González, Javier Lamas Graziani y Juan D. Sandoval González; Editorial Thomson & Paraninfo, segunda edición, pp. 25 – 37.

[2] Instalaciones automatizadas en viviendas y edificios, Leopoldo Molina y José M. Ruíz, Editorial Mc Grawn Hill, pp. 115 – 121.

[3] Dreamweaver 8, Primeros pasos con Dreamweaver, Macromedia, Primera Edición, Septiembre 2005, pp 59 – 103. Disponible en http://lawebdelprogramador.com/cursos/enlace.php?idp=3024&id=54&texto=Dreamweaver.

[4] Manual de Usuario de MPLAB IDE, Microchip Technology Incorporated, 2005, pp. 75 – 82. Disponible en http://ww1.microchip.com/downloads/en/DeviceDoc/51519a.pdf.

[5] Manual de usuario del Compilador C, MPLAB C18 (v3.00), Microchip Technology Incorporated, 2005, pp. 11 – 50. Disponible en http://www.kevin.org/frc/C18_3.0_users_guide.pdf.

[6] Manual de usuario de Programador ICD 2, Microchip Technology Incorporated, 2007, pp. 26 – 51. Disponible en http://ww1.microchip.com/downloads/en/devicedoc/51331B.pdf.

[7] Hoja de datos del PIC 18F97J60, Microchip Technology Incorporated, 2006, pp. 7 – 447. Disponible en http://melabs.picbasic.com/devicedata/39762a.pdf