ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

Capturador de Paquetes en Hardware

Simón Vinicio Castro González
Melany Carla Salvador Antón
Ignacio Marin-Garcia.
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral
Campus Gustavo Galindo Km 30.5, vía Perimetral, Apartado 09-01-5863, Guayaquil, Ecuador
scastro@fiec.espol.edu.ec, msalvador@espol.edu.ec, imaringa@fiec.espol.edu.ec

Resumen

__ *El objetivo de este proyecto es la realización de un prototipo(tipo hardware) capturador de paquetes, permitiéndonos guardar la información que esta transportándose por una red; dado que para su funcionamiento es necesario estar directamente conectados a la red donde se requiera realizar la captura. Por este motivo el dispositivo deberá estar enlazado al distribuidor de red, para ello se asignará ciertos parámetro como dirección IP, mascara de red, un Gateway y finalmente un DNS. Una vez que tenga estos parámetros configurados el dispositivo o prototipo estará en óptimas condiciones para comenzar su captura, la cual será archivada en un micro memoria SD. Su capacidad de captura será, capturar paquetes UDP y TCP dirigidos a su dirección, así como paquetes que no sean dirigidos a su dirección bajo la condición que sean del protocolo internet.*

Palabras claves: *IP, Mascara, Gateway, Hardware, DNS, UDP, TCP.*

Abstract

__ *This project was created to do a hardware prototype that capture network packets that allow us the option of saving the captured information. In order to let prototype do its job is necessary to be directly connected to network. Because of that, the prototype must be linked to the switch of the network, has to be assigned a few parameters as IP address, netmask, gateway and a DNS. When all this parameters were configured the prototype is able to capture packet that will be saved in the SD memory. This prototype can capture UDP and TCP packets routed or not routed to its IP address with the condition that packets belong to internet protocol.*

Keywords: *IP, Netmask, Gateway, Hardware, DNS, UDP, TCP.*

1. Descripción general del proyecto.

Nuestro proyecto consiste en un dispositivo capturador de paquetes de datos. Se debe conectar el equipo al dispositivo de distribución de red, una vez ya realizado esto el dispositivo se encuentra en estado de espera, en donde un administrador le dará la orden de comenzar con la captura de paquetes.

En la etapa de captura existen tres fases, la primera es donde recibiremos las tramas Ethernet en el módulo ENC28J60 con la ayuda de configuración que le otorga el PIC18F4520, al recibir éste sus correspondientes configuraciones está listo para realizar la recepción de paquetes Ethernet.

En la segunda etapa nuestro PIC18F4520 que anteriormente otorgó las correspondientes configuraciones para el módulo ENC28J60, será el encargado de recibir los paquetes (información) que nosotros hayamos solicitado para ser capturados.

Finalmente al ya haber obtenido nuestros paquetes estamos listos para poder guardarlos en una micro memoria SD y así poder tener un archivo de los mismos.

2. Herramientas utilizadas.

Utilizamos el módulo ENC28J60 debido a que es un controlador Ethernet y está diseñado para servir como una interfaz de red Ethernet para cualquier controlador equipado con SPI.

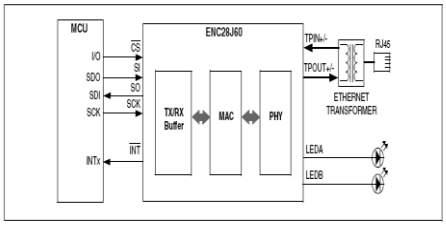

Figura 1. Descripción del Módulo

El ENC28J60 cumple con todas las especificaciones IEEE 802.3. Incorpora una serie de filtros de paquetes planes para limitar los paquetes entrantes. También proporciona un módulo interno DMA para transferencia de datos rápida y hardware cálculo de suma de comprobación asistida, que se utiliza en varios protocolos de red. La comunicación con el controlador de host se implementa a través de un pin de interrupción y el SPI, con velocidades de reloj de hasta 20 MHz Dos dedicados pines se utilizan para LED de conexión y actividad de la red indicación.

Otra herramienta que utilizamos es El PIC18F4520 que permite instrucciones de palabras de 16 bits, bajo consumo de energía, rendimiento de 10 Millones de instrucciones por segundo (MIPS), posibilidad de comunicarse con periféricos y protocolos USB, SPI y TCP/IP.

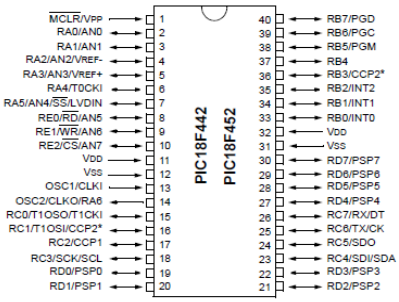

Figura 2. Configuración de Pines del 18f4520

El PIC18F4520 es un microcontrolador tipo CMOS FLASH de 8 bits de arquitectura RISC capaz de operar con frecuencias de reloj hasta los 40 MHz. Posee internamente un oscilador de 4 MHz y un circuito de Power-On Reset que eliminan la necesidad de componentes externos y expanden a 16 el número de pines que pueden ser utilizados como líneas I/O (Input/Output) de propósito general. Proporciona una memoria de datos EEPROM (128 Bytes), una memoria de programa FLASH (2K con 14 bits por localidad), una memoria de datos RAM de propósito general, dos módulos CCP (captura/comparación/ PWM) de 10-bit, un USART y tres comparadores análogos.

La memoria utilizada en la realización de nuestro prototipo está basada en una tarjeta de memoria micro SD con interfase de comunicación de 8 pines. Es compatible con memorias SD con la utilización de un adaptador.

Figura 3. Tarjeta de Memoria micro SD y Adaptador

La tecnología SD permite portabilidad, almacenamiento de datos, lo cual es apropiado para nuestro prototipo. La capacidad de almacenamiento escogida fue de 4GB.

Mikrobasic es una herramienta poderosa, rica en función del desarrollo para microcontroladores. Está diseñado para proporcionar al cliente la solución más fácil posible para el desarrollo de aplicaciones para sistemas embebidos, sin comprometer el rendimiento o el control.

Utilizamos Proteus debido a que es un paquete de software para el diseño de circuitos electrónicos que incluye captura de los esquemas, simulación analógica y digital combinada, además posee una herramienta ARES que se utiliza para el diseño de circuitos impresos. Esta herramienta nos permite simular nuestro proyecto antes de ensamblarlo y así evitar gastos innecesarios de hardware.

3. Diseño

En la figura 4, podemos observar claramente las tres fases de las que consta el dispositivo; partiendo de una fase de recepción de tramas provenientes de la red, una vez ya obtenidos se realiza el procesamiento de cada uno de ellos y así pasar a la tercera etapa que es la de archivo de paquetes.

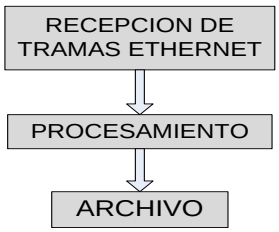

Figura 4. Diagrama de bloques.

En la figura 5, se ve el algoritmo del programa principal se especifican los comandos para leer, procesar y archivar las tramas. Primero se seteo o inicializo los módulos, luego se configuró cada uno de los parámetros necesarios para estar conectados a la red, una vez cumplido esto se procece a capturar las tramas y procesar cada una de ellas, clasificándolas en los diferentes protocolo para ser archivadas, la etapa o fase de captura, procesamiento y archivo se lo realiza en un lazo infinito.

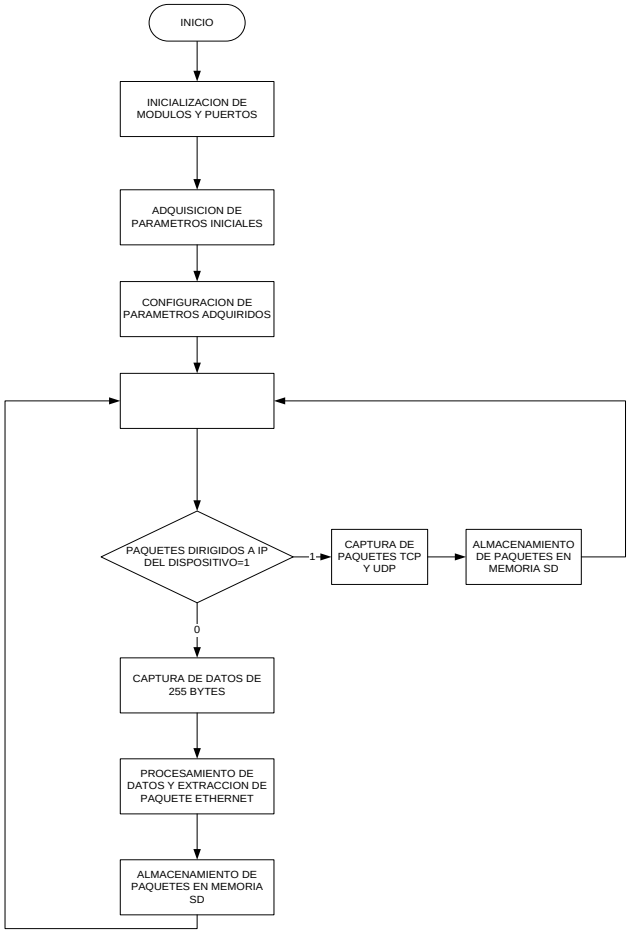

Figura 5. Algoritmo principal

Para comenzar el programa principal es necesario mencionar a las librerias que hemos de usar. A continuación en la figura 6 se detallan las librerias usadas para el desarrollo de la conversión.

```
SPI Ethernet Library       ; diseñada para concectarse a la red
Multi Media Card Library ; diseñada para manejar las mini SD
eth_enc28j60_utils         ; librerías correspondiente al modulo ENC28J60
eth_enc28j60
```

Figura 6. Librerías usadas en el programa principal

Ya realizado la captura de paquetes, entramos en la fase de proceamiento en donde se selecciona si el paquete esta dirigido a nuestro prototipo, si el paquete es no dirigido, su análisis se basa en los protocolos de transporte como son el protocolo UDP y TCP.

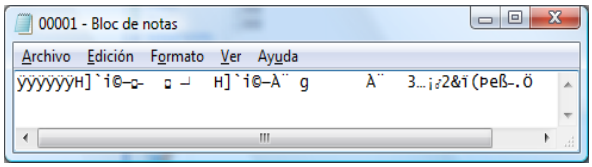

Figura 7. Paquete Capturado.

4. Resultados Experimentales.

Hemos utilizado el software Wireshark para el conteo, captura de paquetes Ethernet y realización de gráficos estadísticos en base al número de paquetes y la clase de los mismos; y adicionalmente se utilizó el software Labview para realizar en envío de paquetes hacia el prototipo.

Al realizar la captura de paquetes UDP y TCP dirigidos pudimos obtener los siguientes resultados: en cuanto al conteo de paquetes.

En primer lugar, enviamos paquetes UDP y TCP dirigidos para nuestro dispositivo, esto lo hicimos con la ayuda de un software creado en Labview.

Para la captura de paquetes UDP se hizo un muestreo, esto partió con el envío de paquetes desde el software y con ello se determinamos la eficiencia del equipo.

En la siguiente tabla se halla detallado el muestreo que realizó,

PAQUETES UDP DIRIJIDOS			
PAQUETES ENVIADOS	PAQUETES RECIBIDOS	PERDIDOS	EFICIENCIA
26	24	2	92,31%
27	25	2	92,59%
34	31	3	91,18%
33	29	4	87,88%
			90,99%

Tabla 1. Resultados UDP.

En las pruebas realizadas se envió paquetes desde el software y estos fueron recibidos por nuestro dispositivo. En la tabla 1 observamos la eficiencia del prototipo, tenemos un 90,99%, una de las causas de no obtener un 100% es tiempo de espera que debe existir entre paquete y paquete para el dispositivo los procese correctamente y como consecuencia tenemos una conectividad de 10Mbits.

La siguiente tabla nos muestra el tiempo de espera promedio entre paquete y paquete para que sean capturados correctamente por el dispositivo,

TIEMPO		
MINIMO (ms)	MAX(ms)	MEDIA (ms)
7	625	229
90	998	350
10	1037	233
32	595	233
		261,25

Tabla 2. Tiempo de Espera en UDP.

Para la captura de paquetes TCP se hizo un muestreo, esto partió con el envió de paquetes desde el software y con ello se determinamos la eficiencia del equipo.

En la siguiente tabla se halla detallado el muestreo que realizó,

PAQUETES TCP DIRIGIDOS			
PAQUETES ENVIADOS	PAQUETES RECIBIDOS	PERDIDOS	EFICIENCIA
34	29	5	85,29
28	24	4	85,71
25	23	2	92,00
23	20	3	86,96
			87,49

Tabla 3. Resultados TCP.

La tabla 3 muestra los datos capturados por nuestro prototipo, y la eficiencia del mismo. Tenemos un 87,49% de eficiencia y al igual que el caso anterior por los tiempos de espera, no logramos llegar a un 100%.

La siguiente tabla nos muestra el tiempo de espera promedio entre paquete y paquete para que sean capturados correctamente por el dispositivo,

TIEMPO		
MINIMO (ms)	MAX(ms)	MEDIA (ms)
895	1678	1286,5
651	1969	1310
670	1435	1052,5
473	2411	1442
		1272,75

Tabla 4. Tiempo de Espera en TCP.

Como en los casos anteriores utilizamos Wireshark, para utilizarlo como referencia para encontrar la eficiencia de nuestro dispositivo.

Desde un ordenador comenzamos a navegar en internet para obtener un flujo de red y al mismo tiempo se realizó la captura con el software y con el prototipo.

Para la captura de paquetes se hizo un muestreo, esto partió la generación de tráfico de red, navegando en internet. Con este ensayo obtuvimos los siguientes resultados.

PAQUETES NO DIRIGIDOS		
GENERADOS	PAQUETES RECIBIDOS	EFICIENCIA %
61	10	16,4
101	16	15,85
78	14	17,95
98	17	17,35
		16,9

Tabla 5. Resultados de Paquetes No Dirigidos.

Se realizaron varias pruebas en la cuales se genero tráfico de red navegando por algunos sitios web, estos paquetes fueron recibidos por capturados por el dispositivo, con estos datos realizamos la tabla 5 donde podemos evaluar la eficiencia, tenemos un 16,9%, igual que en los casos ya revisados los tiempos de espera son una limitante para obtener el 100%.

5. Conclusiones

El conocer las direcciones IP visitadas por empleados y el tener control sobre los mismos, lo podemos realizar con la captura de paquetes de datos de la red empresarial previamente configurada en nuestro prototipo para su posterior análisis.

Debido a la construcción del prototipo con componentes de electrónica básica puede ser considerado de bajo costo y fácil ensamblaje, si se lo fabricase a gran escala se reducirían los costos un cincuenta por ciento (50%) adicional, convirtiéndolo de bajo costos.

6. Recomendaciones

Este prototipo se lo puede mejorar y aumentar la longitud de los paquetes pero para realizar esto se debe de cambiar el módulo, debido a las limitaciones del mismo.

7. Referencias

[1] Jimmy Wales y Larry Sanger; Enciclopedia Libre; http://es.wikipedia.org/wiki/; 23 Febrero 2011.

[2] Centro de publicaciones y documentos en la red; http://www.monografias.com

[3] Héctor Delgado Ureña Poirier y Juan Francisco Rodríguez Martín; Montaje y Configuración de una LAN: Ethernet; http://www.gobiernodecanarias.org/educacion/conocernos_mejor/paginas/ethernet.htm; Septiembre 2010

[4] Sergio Vélez; Blog Modelo OSI'; https://velezconde.wordpress.com/modelo-os