

# Estación Meteorológica Web

Martín Federico Pelliza

Instituto Universitario Aeronáutico – Especialidad en Sistemas Embebidos  
fpelliza@hotmail.com

**Abstract.** Este trabajo, enmarcado en la Especialidad en Sistemas Embebido del Instituto Universitario Aeronáutico, tuvo como objetivo realizar una exploración y evaluación de posibilidades existentes para el desarrollo de aplicaciones embebidas que ofrezcan conectividad de red. En ese marco se eligió como ejercicio el desarrollo de un prototipo de una estación meteorológica web, implementando una aplicación embebida en un módulo de desarrollo del fabricante Netburner al cuál se integró el hardware necesario para medición de temperatura y movimientos sísmicos. Como resultado se obtuvo un dispositivo totalmente autónomo que permite a través de una red ethernet el monitoreo de diferentes condiciones ambientales en tiempo real, el acceso a los registros de los datos almacenados internamente en el dispositivo, y la configuración remota de distintos parámetros de su funcionamiento.

## Introducción

Con el objetivo de explorar las posibilidades actuales a la hora de desarrollar dispositivos con capacidades de red, se eligió como ejercicio el desarrollo de un prototipo para la adquisición y registro de diversas condiciones ambientales, que permita el acceso a los mismos en tiempo real a través de una red ethernet.

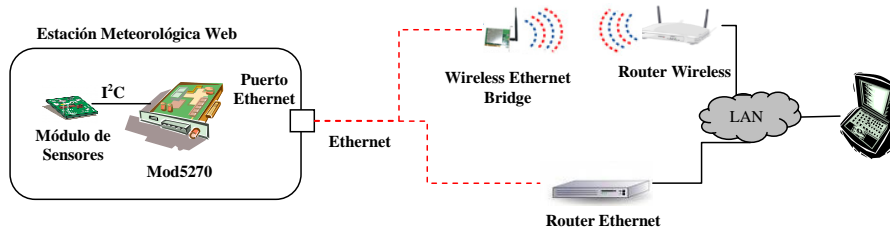
Para la implementación preliminar de una solución, las alternativas que se presentaban eran el desarrollo de un hardware dedicado basado en microprocesador y software embebido que implemente tanto los stacks de protocolos de red y la interfaz con los sensores, o bien la utilización de una plataforma que ya integre la solución de red y acote el trabajo de desarrollo. Se seleccionó esta última opción, ya que se estimó que el esfuerzo de desarrollo requerido para la primera opción sería considerablemente mayor y excedería los límites de tiempo propuestos para este trabajo, y se eligió como plataforma de desarrollo un módulo Mod5270 de Netburner [Ref. 1]

Luego de la selección de la plataforma, se desarrollaron los componentes de Hardware y de Software, actividades que involucraron:

- El diseño y fabricación de un módulo de sensores para la adquisición de datos de aceleración y temperatura, extensible para la incorporación de sensores de presión, humedad, velocidad y dirección de viento, etc.
- El diseño y la implementación en C++ de la aplicación embebida en el Mod5270 para la adquisición y registro de datos del módulo de sensores, y su acceso remoto
- La definición de un protocolo para la comunicación de datos vía UDP.

## 2 Martín Federico Pelliza

- La implementación de un servicio web para el acceso a los datos y configuración del Mod5270, utilizando HTML y JavaScript para el desarrollo de las páginas web. Esta actividad incluyó también la implementación en JAVA de un applet para la visualización en tiempo real y en forma gráfica de los datos de temperatura y movimientos sísmicos.



**Fig. 1.** Contexto de Aplicación

La descripción de las estas tareas y los resultados obtenidos en cada una de ellas se desarrollan en las siguientes secciones.

En **Fig. 1** se muestra el dispositivo desarrollado dentro de un contexto de aplicación.

### Desarrollo de Hardware

#### Selección de la Plataforma

El seleccionó en un módulo Mod5270 de desarrollo de Netburner, que es una plataforma basada en el procesador de 32-bits MFC5270 de ColdFire. El módulo incluye entre sus características principales

- procesador MFC5270
- un puerto 10/100 Ethernet
- interfaz RS-232
- 512K de memoria flash
- 2Mb de SDRAM

Otra característica del Mod5270 es su funcionamiento bajo el sistema operativo  $\mu$ C/OS-II. Este es un sistema operativo en tiempo real multitarea basado en prioridades, cuyas características de pequeño footprint, bajo costo, sencillez y robustez lo hacen ideal para ser utilizado en aplicaciones embebidas.

El sistema operativo está integrado con el sistema de I/O del Mod5270 y ofrece, además de las funciones y rutinas básicas para la creación y administración de tareas, librerías de soporte para diversos protocolos de comunicación de red (entre otros DHCP, TCP, UDP, HTTP) y de interfaz con otros dispositivos (tales como I<sup>2</sup>C y SPI) [**Ref. 2**].

### Desarrollo del Módulo de sensores

Se desarrolló un módulo básico de hardware de adquisición de datos, cuyo diseño se enfocó en la facilidad para la integración de distintos tipos de sensores. Con este criterio se adoptó el bus I<sup>2</sup>C<sup>TM</sup><sup>1</sup> para la comunicación entre los sensores del módulo y la plataforma Mod5270. El bus I<sup>2</sup>C es un bus de bi-direccional, que provee un mecanismo sencillo y eficiente para el intercambio de datos entre dispositivos, con una interconexión mínima entre ellos, ya que es un bus de dos cables.

La flexibilidad del bus permite la incorporación de sensores al módulo de manera directa, con la sola condición de que estos soporten una interfaz I<sup>2</sup>C. De esta forma el módulo de sensores es extensible y permite la medición variadas condiciones ambientales, además de las ofrecidas por el prototipo, como por ejemplo presión, humedad, velocidad viento, etc.

El módulo de adquisición de datos desarrollado para este prototipo cuenta con dos sensores para la medición de temperatura, y un sensor para la medición de movimientos sísmicos.

Para la medición de temperatura se escogieron los sensores digitales de temperatura TMP101 de Texas Instruments [Ref. 3] y STTS75 de STMicroelectronics [Ref. 4], mientras que para la medición de movimientos sísmicos se utilizó el acelerómetro digital de 3 ejes LIS3LV02DL STMicroelectronics [Ref. 5]. Todos los sensores soportan la interfaz I<sup>2</sup>C, están interconectados al mismo bus y se utilizan en modo esclavo, mientras que el Mod5270 es utilizado como maestro.

Los sensores de temperatura poseen características similares: un rango de medición de -55°C a +125°C y resolución configurable entre 9 y 12 bits (0.5°C a 0.0625°C). La utilización de dos sensores de temperatura en el módulo de sensores fue adoptada en un principio con el fin de evaluar el desempeño de cada sensor. Finalmente se decidió configurar cada sensor con una resolución diferente. Aún el sensor configurado con la resolución más baja de temperatura (0,5°C ó 9 bits) cubre los requerimientos a los fines de este prototipo.

Por otro lado se utiliza el sensor LIS3LV02DL para la medición de aceleraciones. Este dispositivo permite rangos de aceleración a fondo de escala de +/-2.0g y +/-6.0g, con un ancho de banda de la conversión variable entre 10Hz y 640Hz, ambos seleccionables por software durante el funcionamiento. Como la aceleración en la mayoría de los terremotos moderados está comprendida entre 0,05 g y 0,35 g (llegando en algunos casos a alcanzar aceleraciones de 0,5 g cuando el movimiento del suelo es medido sobre suelo firme o roca muy cerca de la fuente de ondas), y su frecuencia típica se encuentra entre 0,5Hz y 2Hz (llegando a 4Hz en casos excepcionales) [Ref. 6] y [Ref. 7], este dispositivo es apropiado para la medición de movimientos sísmicos.

Los diagramas esquemáticos de la placa de sensores desarrollada se incluyen como anexo en este trabajo.

---

<sup>1</sup> I<sup>2</sup>C es una marca de NXP Semiconductors

## Desarrollo del Software

Se desarrolló una aplicación embebida<sup>2</sup> en el Mod5270, encargada de obtener los datos del módulo de sensores, mantener un registro de los mismos y permitir el acceso a estos datos en forma remota a través de HTTP, FTP o UDP.

Las principales tareas ejecutadas por la aplicación pueden dividirse en

- Adquisición de datos de temperatura a través del módulo de sensores
- Adquisición de datos de movimientos sísmicos a través del módulo de sensores
- Servicio UDP para la transmisión en tiempo real de datos de mediciones a cualquier cliente mediante un protocolo propio
- Servicio FTP para la transmisión de los registros completos de mediciones de temperatura y movimientos sísmicos.
- Servicio HTTP para el acceso a las mediciones y configuración del Mod5270.
- Sincronización del reloj interno mediante un cliente NTP

El Mod5270 opera bajo un sistema operativo multitarea preemptivo, basado en prioridades. La aplicación ejecuta las rutinas de acceso a la red (inicialización del stack TCP y adquisición de IP dinámica) y luego crea, configura e inicia las tareas que ejecutan las actividades mencionadas en el párrafo anterior en forma independiente.

La aplicación permite almacenar hasta un máximo de 3600 muestras de temperatura y 18000 muestras de aceleración. El número máximo de muestras estuvo limitado por las capacidades de memoria del Mod5270, y se seleccionó asumiendo un tiempo de 1 hora entre cada descarga de los registros completos, con el período de muestreo de temperatura y movimiento más corto permitido.

### Servicio de FTP

Permite obtener archivos con los registros de las últimas muestras de temperatura y movimientos sísmicos, como así también un registro completo de los eventos del sistema. En esta versión de la aplicación en el servidor sólo se implementaron los siguientes comandos definidos por el protocolo FTP: “*ls*” para obtener los nombres de los archivos disponibles, “*get*” para descargar archivos, y “*bye*” para finalizar la conexión.

Los clientes FTP pueden descargar archivos con los diferentes registros en formato de texto. Estos archivos son generados bajo demanda, ya que el Mod5270 no posee un sistema de archivos, y contienen una lista de todas las muestras disponibles.

Las siguientes figuras presentan el formato de los archivos *temperatura.txt* y *movimientos.txt*. Cada entrada del archivo contiene la hora, un índice representando la secuencia y un valor entero que representa la medición, de acuerdo a las tablas de conversión **Table 1** (pág. 13) para la temperatura y **Table 2** (pág. 13) para la aceleración.

---

<sup>2</sup> El código fuente de toda la aplicación se adjunta como anexo en este trabajo

Registro de las últimas 3600 muestras de temperatura [451]			
Fecha	Hora	[idx]	Temperatura
2009-7-1	04:49:17GMT	[0]	5
2009-7-1	04:49:18GMT	[1]	10
2009-7-1	04:49:19GMT	[2]	15
2009-7-1	04:49:20GMT	[3]	20
...			
2009-7-1	04:56:48GMT	[451]	160

**Fig. 2.** Formato del archivo *temperaturas.txt*

El registro de movimiento presenta los valores de aceleración para los tres ejes (Norte-Sur, Este-Oeste y Vertical). La marca de tiempo con la que se registran estas muestras se representa con una precisión de milisegundos.

Registro de las últimas 18000 muestras de aceleración [2216]					
Fecha	Hora	[idx]	Mov.N-S	Mov.E-O	Mov.Vert
2009-7-1	04:49:17.750GMT	[0]	4	3	2
2009-7-1	04:49:17.950GMT	[1]	8	6	4
2009-7-1	04:49:18.150GMT	[2]	12	9	6
2009-7-1	04:49:18.350GMT	[3]	16	12	8
...					
2009-7-1	04:56:40.950GMT	[2216]	108	51	34

**Fig. 3.** Formato del archivo *movimientos.txt*

También está disponible el archivo *eventos.txt* con los registros de los eventos del sistema, cuyo formato se muestra en **Fig. 4**

Registro de Eventos del Sistema		
Fecha	Hora	Evento
1970-1-1	00:00:01GMT	Error al sincronizar con el servidor NTP
2009-7-1	04:49:17GMT	Inicialización completada.
2009-7-1	20:59:38GMT	Error al sincronizar con el servidor NTP
2009-7-1	23:37:54GMT	Modificación de contraseña
2009-7-1	23:38:12GMT	Modificación de configuración

**Fig. 4.** Formato del archivo *eventos.txt*

### Servicio UDP

Esta tarea es la encargada de procesar los paquetes UDP recibidos en el puerto configurado en la aplicación (se utilizó el puerto 7500 por defecto). El servidor recibe mensajes solicitando distintos tipos de muestras (UDP Sample Requests), respondiendo a los mismos con las muestras requeridas en un mensaje de respuesta (UDP Sample Response). Tanto las solicitudes como las respuestas viajan en el campo de datos de los paquetes UDP.

El formato del mensaje UDP Sample Request está resumido en **Fig. 5**



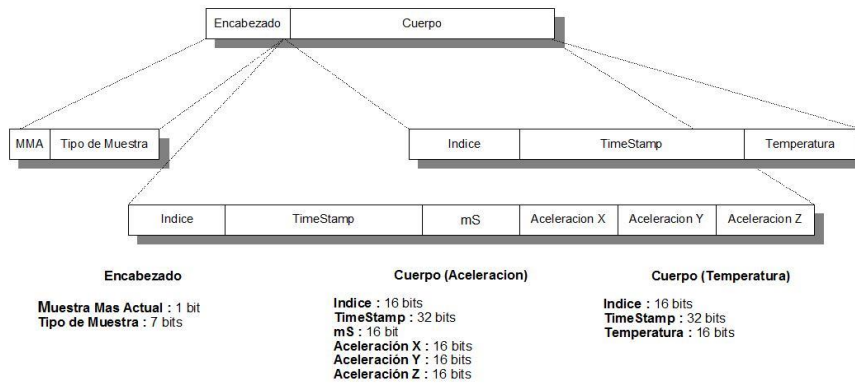
**Fig. 5.** UDP Sample Request

El campo *Tipo de Muestra* contiene un código de 8 bits cuyo valor identifica solicitudes de muestras de temperatura o aceleración.

El campo *Cantidad de Muestras* (entero de 16 bits con signo) es utilizado para solicitar el envío una cantidad fija de muestras.

El campo *Ultima Muestra Recibida* (entero de 16 bits con signo) es utilizado para solicitar el envío de una cantidad variable de muestras.

Un mensaje UDP Sample Response puede contener una o mas muestras, cada una con un formato como el que se detalla en Fig. 6. El encabezado de una muestra indica, en su MSB, si esa muestra es la última muestra disponible o si existen otras muestras más actuales disponibles, y en sus 7 bits restantes el código del tipo de muestra (aceleración o temperatura). Los campos *Indice* y *TimeStamp* son comunes a cualquier tipo de muestra, mientras que los restantes campos se diferencian para las muestras de temperatura y aceleración.



**Fig. 6.** UDP Sample Response

El campo *Indice* (entero con signo de 16 bits) indica el número de secuencia de la muestra. Su valor máximo depende del tipo de muestra, siendo 3600 para muestras de temperatura y 18000 para muestras de aceleración.

El campo *TimeStamp* (entero sin signo de 32 bits) contiene la fecha y hora de la muestra expresada en Tiempo Universal Coordinado (UTC)

El campo *mS* (entero con signo de 16 bits) está presente solo en muestras de aceleración y contiene los milisegundos del TimeStamp

Los campos de *Temperatura* (16 bits) y *Aceleración X-Y-Z* (cada una un valor de 16 bits) contienen el valor de la medición correspondiente, según la codificación representada en las tablas 1 y 2 (los bits 15-14-14-12 son siempre 0).

Se permiten hasta 50 muestras en un mensaje UDP Sample Response (para limitar el tamaño del paquete UDP a 750 bytes).

Este protocolo es el utilizado para transmitir los datos en tiempo real hacia las aplicaciones externas. Cuando un UDP Sample Request contiene en el campo *Ultima Muestra Recibida* el número de secuencia de una muestra, el servidor enviará un mensaje UDP Sample Response con todas las muestras desde el índice indicado, hasta la muestra más actual disponible. El campo *Cantidad de Muestras* es considerado sólo si el campo *Ultima Muestra Disponible* contiene un número negativo.

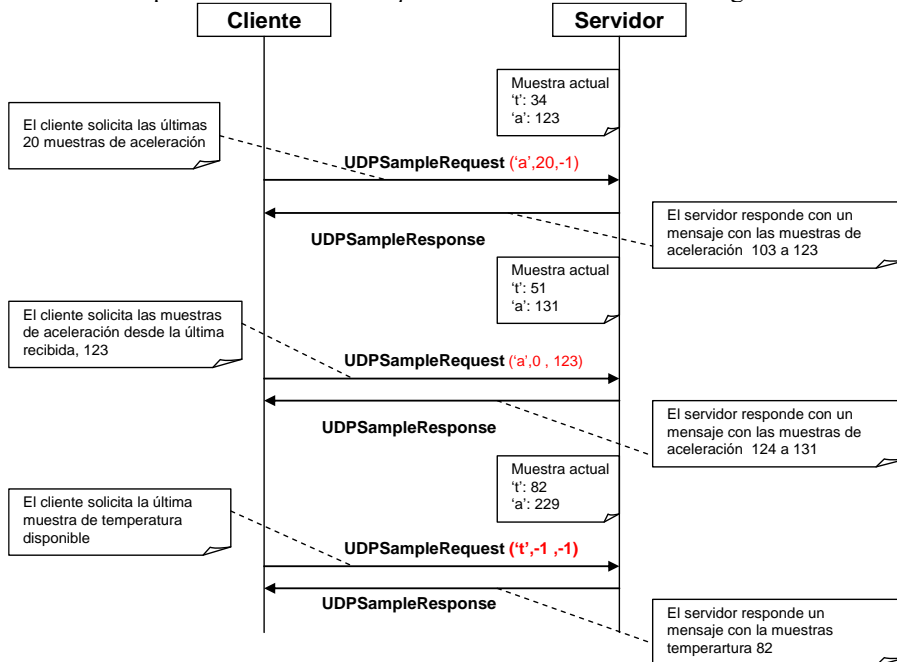


Fig. 7. Funcionamiento del protocolo para la solicitud de muestras

En Fig. 7 se ejemplifica el uso de este protocolo. Además, se adjunta como anexo el código fuente del Applet desarrollado para la visualización de datos en tiempo real desde la página web, que hace uso de este protocolo para la adquisición de las muestras de temperatura y aceleración.

### Servicio HTTP

Esta tarea es la encargada de procesar solicitudes HTTP. El servidor soporta solicitudes GET y POST, respondiendo a las mismas mediante paginas html generadas dinámicamente. Accediendo a este servicio un usuario puede:

## 8 Martín Federico Pelliza

- Configurar distintos parámetros de funcionamiento del Mod5270
- Obtener registros parciales de las muestras de temperatura y movimientos sísmicos
- Obtener registros de eventos del sistema
- Mostrar en tiempo real las muestras temperatura y movimientos sísmicos

### *Página de configuración*

Los parámetros configurables a través la página de configuración son el período de muestreo de temperatura y movimientos sísmicos (aceleración), la dirección IP del servidor NTP con el cual la aplicación se sincroniza, el puerto donde el servicio UDP está disponible, el tipo de filtro aplicado a las muestras de aceleración y la selección del sensor de temperatura activo.

En este prototipo la implementación de HTTP soporta autenticación “BASIC”, y este esquema es utilizado para el acceso a la página de configuración de los parámetros de funcionamiento, para la cuál la provisión de credenciales es obligatoria. Este esquema es utilizado sólo con fines de identificación, ya que no constituye un método seguro de autenticación (los datos son transmitidos como texto claro).

El acceso está restringido para el usuario **admin**. La contraseña (por defecto “admin”) es también configurable a través de esta página. Cada cambio en la configuración de los parámetros de funcionamiento del Mod5270, es registrado como un evento de sistema.

The screenshot shows the configuration interface for the IUA system. The page title is "IUA Especialidad en Sistemas Embebidos". The navigation menu includes "Inicio", "Configuración", "Mediciones", and "Registros". The "Configuración" tab is active, and it contains sub-tabs for "Sistema" and "Contraseña". The configuration form includes the following fields and options:

- NTP Server IP: 164.67.62.194
- UDP Port: 7500
- Muestreo temperatura(Seg): 1
- Muestreo aceleración(mS): 200
- Sensor de Temperatura:  TMP101 (12 bits)  STTS75 (9 bits)

A "Configurar" button is located at the bottom of the form.

**Fig. 8.** Configuración de Sistema

### *Mediciones en tiempo real*

Para la visualización en tiempo real de los datos, el servidor descarga en el navegador del cliente un applet encargado de obtener (vía UDP y adhiriendo al protocolo



descrito en secciones anteriores) y procesar los datos, desplegándolos en forma gráfica. El applet solicita muestras cada 750mS y actualiza los gráficos con las muestras del último paquete UDP recibido. Se despliegan las últimas 50 muestras de temperatura y 300 muestras de aceleración (aprox. 1 minuto a la frecuencia de muestreo más alta).

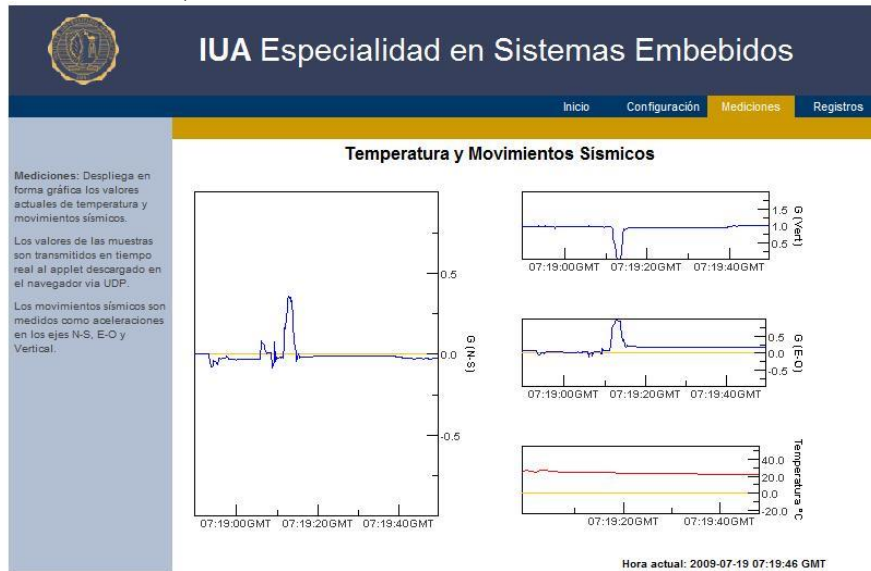


Fig. 9. Mediciones en tiempo real

### Registros

Los registros parciales de las últimas muestras de temperatura y movimientos sísmicos, como así también de los eventos del sistema se obtienen en la pestaña Registros de la página web.



**Fig. 10.** Registros de Aceleración

Los eventos del sistema también son registrados y pueden ser accedidos a través de la página esta misma página.

Es todos los casos, posible seleccionar la visualización de los entre 10 y 100 últimos registros. Los registros completos pueden obtenerse vía FTP.



Fig. 11. Registro de Eventos del Sistema

### Sincronización de la hora del sistema

Esta tarea se ocupa de sincronizar el reloj interno del Mod5270 a través de un cliente NTP. La aplicación inicialmente sincroniza el reloj interno del sistema mediante una solicitud a un servidor NTP, y a partir de ese momento se re-sincroniza cada 4 horas.

De existir algún error durante la sincronización, la aplicación intenta nuevamente restablecer el contacto con el servidor NTP cada 10 minutos, hasta lograr la re-sincronización (inicialmente 00:00:00 GMT 1 Ene 1970 en caso de no poder sincronizarse).

Los errores durante la sincronización de la hora son almacenados en el registro de eventos del sistema.

### Adquisición de datos a través del módulo de sensores

Existen dos tareas encargadas de obtener cíclicamente los datos de temperatura y aceleración, realizando el muestreo de datos con la frecuencia configurada a través de la página de configuración del sistema. El período de muestreo para temperatura puede variar entre 1 y 3600 segundos, mientras que para aceleración el período de muestreo puede variar entre 200 y 1000 milisegundos (en saltos de 50 milisegundos).

La aplicación permite registrar hasta un máximo de 3600 muestras de temperatura y 18000 muestras de aceleración. Este número máximo de muestras se determinó asumiendo una hora de almacenaje, con el período de muestreo de temperatura de 1 segundo y de movimiento de 200mS. Las muestras se almacenan en un buffer circular, por lo que las muestras más antiguas son reemplazadas por nuevas muestras una vez que el límite máximo de cada tipo de muestra es alcanzado.

La adquisición de los datos de temperatura y movimiento del módulo de sensores se realiza a través del protocolo I<sup>2</sup>C. El bus I<sup>2</sup>C es un bus de dos líneas, SDA (Serial DAta) y SCL (Serial Clock). Cada dispositivo conectado al bus posee una dirección única de 7 bits, y puede actuar como transmisor o receptor de datos. El dispositivo que inicia la comunicación es llamado Maestro, mientras los restantes dispositivos cumplen la función de Esclavos. Las transferencias son iniciadas mediante la generación por parte del maestro de una condición de START en el bus (la línea SDA es llevada de alto a bajo mientras SCL está en alto), seguida de 7 bits con la dirección del dispositivo al cuál se quiere acceder más un bit que indica el tipo de operación (lectura o escritura) que se desea realizar con ese dispositivo. El dispositivo seleccionado responde al maestro con una señal de Acknowledge sobre la línea SDA. Dependiendo del tipo de operación, el maestro o el esclavo escribe uno o más bytes en el bus, cuya recepción es confirmada (byte a byte) por el otro dispositivo con una señal de Acknowledge sobre la línea SDA. Por convención, cuando sobre el esclavo se realizan operaciones de escritura, y el dispositivo posee registros internos, los bytes transmitidos por el maestro después de la dirección referencian dichos registros internos. Finalmente, el maestro libera el bus generando una condición de STOP (a línea SDA es llevada de bajo a alto mientras SCL está en alto). La

**Maestro escribiendo un byte a un sub-registro del Esclavo**

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

**Maestro leyendo un byte de un sub-registro del Esclavo**

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

**ST:** Start  
**SP:** Stop  
**SAD + W:** Dirección del Esclavo – Escritura  
**SAD + R:** Dirección del Esclavo – Lectura  
**SUB:** Dirección del sub-registro a leer/escribir (8 bits)  
**DATA:** Dato (8 bits)  
**SAK:** Ack Esclavo  
**MAK:** Ack Maestro

**Fig. 12** describe estas operaciones.

**Maestro escribiendo un byte a un sub-registro del Esclavo**

Master	ST	SAD + W		SUB		DATA		SP
Slave			SAK		SAK		SAK	

**Maestro leyendo un byte de un sub-registro del Esclavo**

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

**ST:** Start  
**SP:** Stop  
**SAD + W:** Dirección del Esclavo – Escritura  
**SAD + R:** Dirección del Esclavo – Lectura  
**SUB:** Dirección del sub-registro a leer/escribir (8 bits)  
**DATA:** Dato (8 bits)  
**SAK:** Ack Esclavo  
**MAK:** Ack Maestro

**Fig. 12.** I<sup>2</sup>C Lectura y escritura de datos

Los sensores de temperatura son configurados inicialmente para una resolución de 12 bits el TMP100 y 9 bits el STTS75, en modo de muestreo continuo (una muestra cada 340ms). El sensor utilizado para la obtención de las muestras es configurado a través de la página de configuración de la aplicación. La tabla de conversión de bits a [°C] se muestra en **Table 1**

Temperatura (°C)	Salida digital (BINARIO)	HEX
128	0111 1111 1111	7FF
100	0110 0100 0000	640
25	0001 1001 0000	190
0.25	0000 0000 0100	004
0.0	0000 0000 0000	000
-0.25	1111 1111 1100	FFC
-25	1110 0111 0000	E70
-55	1100 1001 0000	C90
-128	1000 0000 0000	800

**Table 1.** Conversión de muestras de temperatura

El sensor de aceleración es configurado para obtener una escala de medición de de +/- 2.0g con una resolución de 12 bits (~0,001 g), con una frecuencia de muestreo de 40Hz (25mS). La tabla de conversión de bits a [G] se muestra en **Table 2**

Aceleración (g)	Salida digital (BINARIO)	HEX
2.0000	0111 1111 1111	7FF
1.5625	0110 0100 0000	640
1.0000	0100 0000 0000	400
0.2930	0001 0010 1100	12C
0.0039	0000 0000 0100	004
0.0000	0000 0000 0000	000
-0.0039	1111 1111 1100	FFC
-0.2930	1110 1101 0100	ED4
-1.5625	1001 1100 0000	9C0
-2.0000	1000 0000 0000	800

**Table 2.** Conversión de muestras de aceleración

## Conclusiones

Durante el ejercicio se realizaron actividades de investigación sobre las diferentes alternativas existentes a la hora de diseñar una sistema embebido, lo que involucró el estudio de microprocesadores, plataformas, sensores, e interfaces, la aplicación de diferentes protocolos de red, el diseño y fabricación de un módulo de hardware y la programación, en C++, JAVA, JavaScript, html y sobre una plataforma basada en un sistema operativo en tiempo real, de una aplicación embebida completa.

El resultado demostró que existen en el mercado actual opciones para el desarrollo de sistemas embebidos con conectividad de red que involucran un esfuerzo de desarrollo relativamente acotado; la alternativa que se evaluó no solo incorpora la solución de red, sino que además ofrece herramientas (en la forma de paquete de librerías, entorno

de desarrollo integrado, hardware de base para la prototipación) que facilitan tanto el desarrollo de las aplicaciones embebidas, como así también la integración de los diferentes componentes de hardware necesarios para implementar un sistema totalmente autónomo con capacidades de red.

Si bien el desarrollo de un producto aplicado supone la definición de requerimientos más específicos basados en necesidades concretas, el presente trabajo puede constituir un punto de partida. Posibles evoluciones del mismo podrían incorporar el monitoreo de otras variables climáticas (integrando nuevos sensores al módulo de sensores e implementando nuevas tareas en la aplicación), la definición de una estrategia para el registro por disparo en lugar del registro continuo para las mediciones de movimientos sísmicos o la medición de determinadas variables sólo bajo demanda. Otras líneas de trabajo podrían involucrar la implementación de estrategias para la gestión de fallos o la incorporación de capacidades de seguridad (por ejemplo el uso de SSL para el cifrado de datos intercambiados entre servidor y cliente), ambas opciones que fueron excluidas desde un principio de los alcances del ejercicio.

### **Acrónimos y Referencias**

**DHCP:** Dynamic Host Configuration Protocol (<http://www.ietf.org/rfc/rfc2131.txt>)

**FTP:** File Transfer Protocol (<http://www.ietf.org/rfc/rfc0959.txt>)

**HTTP:** HyperText Transfer Protocol (<http://www.ietf.org/rfc/rfc2616.txt>)

**I<sup>2</sup>C:** Inter-Integrated Circuit

([http://www.nxp.com/acrobat\\_download/literature/9398/39340011.pdf](http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf))

**NTP:** Network Time Protocol (<http://www.ietf.org/rfc/rfc1305.txt>)

**SPI:** Serial Peripheral Interface

([http://www.freescale.com/files/microcontrollers/doc/ref\\_manual/S12SPIV3.pdf](http://www.freescale.com/files/microcontrollers/doc/ref_manual/S12SPIV3.pdf))

**SSL:** Secure Socket Layer (<http://www.ietf.org/rfc/rfc2246.txt>)

**TCP:** Transmission Control Protocol (<http://www.ietf.org/rfc/rfc793.txt>)

**UDP:** User Datagram Protocol (<http://www.ietf.org/rfc/rfc0768.txt>)

**Ref. 1** - Netburner Mod5270 Data Sheet -

[http://www.netburner.com/downloads/mod5270/mod5270\\_datasheet\\_pinout\\_diagram.pdf](http://www.netburner.com/downloads/mod5270/mod5270_datasheet_pinout_diagram.pdf)

**Ref. 2** - Netburner Mod5270 Hardware User's Manual -

<http://csserver.evansville.edu/~richardson/courseware/resources/EE458/nburn/docs/platform/Mod5270.pdf>

**Ref. 3** - Texas Instruments TMP100 Data Sheet -

<http://www.datasheetcatalog.org/datasheet/texasinstruments/tmp101.pdf>

**Ref. 4** - STMicroelectronics STTS75 Data Sheet -

<http://www.stmicroelectronics.fr/stonline/products/literature/ds/13298/stts75.pdf>

**Ref. 5** - STMicroelectronics LIS3LV02DL Data Sheet -

<http://www.stm32circle.com/projects/file/DataSheet/lis3lv02dl.pdf>

**Ref. 6** - *Terremotos* – Bruce A Bolt – Ed. Reverte -

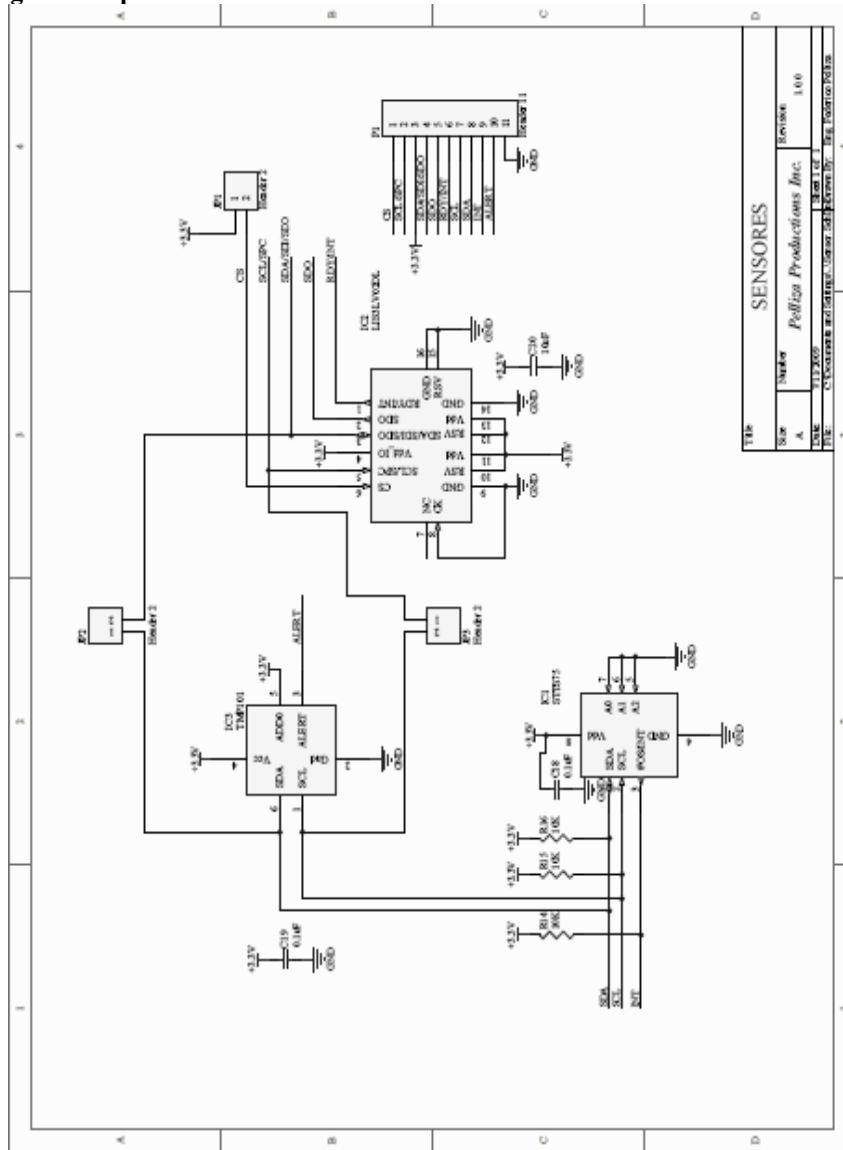
<http://books.google.com.ar/books?id=KmHP0lGeQWQC&lpq=PP1&pg=PP1#v=onepage&q=&f=false>

**Ref. 7** - *Registro y tratamiento de acelerogramas* – Carreño, Suárez, Bravo y Tordesillas -

Física de la tierra, ISSN 0214-4557, N° 11, 1999 (Ejemplar dedicado a: Ingeniería sísmica), pags. 81-111 - <http://revistas.ucm.es/fis/02144557/articulos/FITE9999110081A.PDF>

Anexos

Diagrama esquemático del módulo de sensores



SENSORES			
Rev	Modelo	Revisión	1.0.0
1.0	1.1.1.2009	1.0	1.0.0
1.0	C:\Users\user\Desktop\1.0.0		1.0.0