

Desarrollo de un equipo de medición de audiencia televisiva

Gabriel P. Gómez Sena, gabriel@inno.com.uy¹

Juan P. Oliver, jpo@fing.edu.uy²

Omar Barreneche, ob@inno.com.uy³

Resumen—Este trabajo describe la arquitectura de un equipo de medición de audiencia televisiva, *Telemeter*TM, diseñado y desarrollado por Inno.Com⁴ en base a una especificación inicial de la empresa Mediciones y Mercado⁵. El equipo consiste de un sintonizador de TV con funciones de OSD (On Screen Display), capacidad de almacenamiento y transporte de datos a través de múltiples tecnologías de telefonía celular. Se describe someramente la arquitectura del equipo centrándose en la complejidad del software desarrollado. Cientos de unidades del equipo se encuentran en funcionamiento ininterrumpido desde hace más dos años demostrando la robustez del diseño e implementación del equipo.

I. INTRODUCCIÓN

El *Telemeter* fue diseñado para utilizarse en un sistema de medición de audiencia televisiva en el cual una muestra de televidentes colabora utilizando el equipo por cierto lapso.

El equipo recoge los datos de audiencia mediante formularios que presenta al televidente sobreimpresos en la pantalla del televisor. El televidente debe responder a dichos formularios indicando cuál o cuáles de los integrantes del núcleo familiar están presentes. Estos datos son almacenados por el equipo.

El *Telemeter* puede ser encuestado por un sistema central de control que recoge los datos almacenados y también puede informar autónomamente los cambios de estado al equipo central. Los sistemas de comunicación desarrollados hasta el momento se basan en sistemas celulares de modo de interferir lo menos posible con el hogar del encuestado. El sistema de comunicación se basa en un protocolo de órdenes y respuestas que permite (además de recoger los datos almacenados) obtener el estado del equipo y alterar su configuración.

¹ G. Gómez Sena es ingeniero en electrónica. Es profesor agregado del Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de la República (Uruguay) y trabaja como consultor y desarrollador en InnoCom.

² J. P. Oliver es ingeniero electricista. Es profesor agregado del Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de la República (Uruguay).

³ O. Barreneche es ingeniero en telecomunicaciones. Es director de InnoCom y Chief Technological Officer de ISN Telcom, Miami, EEUU

⁴ InnoCom es una consultora uruguaya especializada en telecomunicaciones.

⁵ Mediciones y Mercado es una empresa de investigación de mercados y de desarrollo de tecnología específica.

II. FUNCIONAMIENTO DEL SISTEMA

Simplificadamente el *Telemeter* es un sintonizador de TV (también llamado canalera o tuner) con funcionalidades agregadas.

Permite presentar formularios sobreimpresos en la pantalla del televisor y puede registrar los eventos que se desencadenan. Los registros se almacenan en memoria y pueden recogerse mediante un sistema de comunicaciones desde un equipo central.

En la operación normal del equipo, el televidente opera el control remoto para seleccionar el canal de su preferencia como lo realiza normalmente con cualquier televisor o sintonizador común. Cuando el equipo determina que el televidente se ha detenido en un canal, le presenta un formulario al que se llamará “*formulario de audiencia*”. Este formulario presenta la lista de nombres correspondiente a los integrantes del núcleo familiar con un número asociado a cada uno. A modo de ejemplo, el formulario es del tipo del indicado en la Figura 1.

1 _ Televidente 1
2 _ Televidente 2
3 _ Televidente 3
4 _ Televidente 4
5 _ Televidente 5
6 _ Televidente 6
7 _ Televidente 7
0 _ No encuestado
ENTER para finalizar

Figura 1 – Formulario de audiencia

Los nombres presentados en el formulario son los correspondientes al núcleo familiar (en el ejemplo se indican nombres genéricos) que se programan de acuerdo a los integrantes del núcleo familiar.

Cuando se presenta el formulario anterior, el televidente debe accionar con el control remoto el número correspondiente a su nombre, desapareciendo el formulario. El formulario se presentará periódicamente al televidente a efectos de corroborar su permanencia frente al televisor. En caso que no se responda a alguno de los formularios presentados, el equipo registrará que no hay televidentes y volverá a presentar el formulario al cabo de un cierto tiempo.

El comportamiento del equipo se controla por una serie de temporizadores configurables que determinan los tiempos

en que se presentan y se quitan los formularios en caso de no haber respuesta. También un temporizador decide cuándo el televidente se ha detenido en un canal y ha finalizado el “zapping”.

Las presencias detectadas a través de las respuestas a los formularios son almacenadas por el sistema en memoria. Cuando no hay respuesta a los formularios también se generan registros que indican este hecho.

Con los datos registrados se puede determinar qué canales de televisión fueron vistos por cada integrante del núcleo familiar y durante qué lapsos de tiempo. Estos datos son procesados luego por un equipo central para generar los datos de audiencia.

Para extraer los datos almacenados, el equipo posee un módulo de comunicaciones que permite enviar los registros

a un equipo central. El módulo de comunicaciones tiene como premisa la menor interacción con la instalación del hogar del televidente, por lo que se basa en tecnología celular. Se han desarrollado hasta el momento diferentes módulos para la comunicación mediante: CDPD, SMS sobre TDMA, CSD sobre TDMA, CSD sobre GSM y GPRS sobre GSM. Para las comunicaciones celulares se utilizaron módulos y teléfonos convencionales. El diseño modular permite el desarrollo futuro de nuevos módulos de comunicación adicionales.

Además de la funcionalidad principal descrita, el sistema tiene la funcionalidad de presentar un “*formulario de encuesta*” que permite desplegar en la pantalla del televisor del usuario una pregunta utilizable para otros tipos de análisis de mercado.

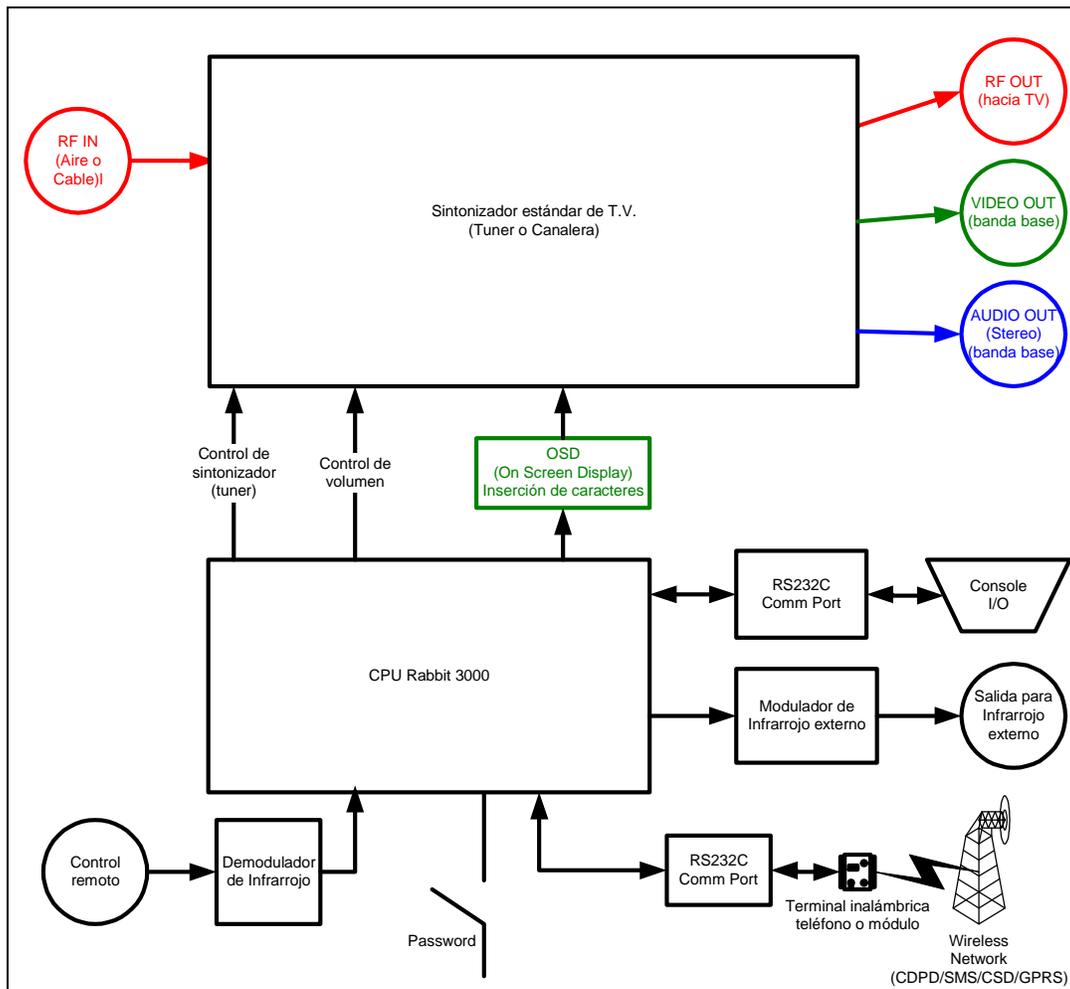


Figura 2 – Bloques funcionales del Telemeter

III. ESTRUCTURA DEL HARDWARE

A. Bloques funcionales

Los bloques funcionales que componen el *Telemeter* pueden apreciarse en la Figura 2.

El *Telemeter* está integrado por un sintonizador de TV estándar o canalera al que se incorpora una placa basada en un microprocesador Rabbit 3000⁶ [1][2][3]

La CPU recibe la señal del control remoto, la decodifica e interpreta cuál tecla del control fue presionada. De acuerdo al estado del sistema, esas órdenes recibidas por el usuario mediante el control remoto generan acciones que pueden

⁶ Rabbit Semiconductor (www.rabbitsemiconductor.com)

ser de cambio de canal (control del sintonizador), control de volumen, despliegue o eliminación de mensajes sobreimpresos en la pantalla del usuario, ajuste fino de sintonía, modificación de la lista de canales favoritos.

Posee adicionalmente un puerto serie de consola que permite tanto comandar el equipo manualmente utilizando un conjunto elemental de comandos así como configurar el equipo completamente mediante un protocolo de órdenes y respuestas cuyos detalles se indican más adelante.

El módulo de comunicaciones permite configurar el equipo desde un centro de operaciones remoto utilizando un canal encriptado. Los mecanismos de comunicación actualmente desarrollados son:

- CDPD (Cellular Digital Packet Data) utilizando un módulo OEM
- CSD (Cellular Switched Data) sobre la red TDMA utilizando un teléfono móvil celular estándar
- SMS (Short Messages Services) sobre la red TDMA utilizando un teléfono móvil celular estándar
- CSD sobre la red GSM utilizando un teléfono móvil celular estándar
- GPRS (General Packet Radio Service) sobre la red GSM utilizando un teléfono móvil celular estándar

El sintonizador del *Telemeter* no posee mecanismos de decodificación de canales para los sistemas codificados, por lo que en caso que el televidente posea un sistema codificado, se dispone de una salida de infrarrojo generada por el *Telemeter* para comandar el sintonizador del decodificador externo.

B. Detalles del diseño físico

El diseño físico del equipo mantiene prácticamente intacto el sintonizador o canalera, habiéndose diseñado un circuito impreso que se conecta como una placa “hija” de la placa de la canalera. Luego de un estudio detallado del circuito de la canalera, se decidió sustituir algunos componentes por conectores, a partir de los que la placa hija desarrollada por InnoCom obtiene los puntos de contacto con el circuito original de la canalera. Particularmente se reemplaza el microprocesador original y de esta forma la placa hija accede a las señales necesarias para comandar los componentes del sistema (sintonizador, control de volumen, recepción de señal infrarroja del control remoto, relay de encendido). Asimismo se utiliza un conector para superponer los caracteres generados por el chip de OSD residente en la placa hija.

Además de la placa hija mencionada, se desarrollaron dos placas auxiliares para los módulos de comunicación basados en teléfonos celulares estándar que resuelven la alimentación permanente de los teléfonos y mecanismos de reset particulares.

C. Uso de funciones del microprocesador

En base a los requerimientos del equipo, se decidió su implementación con un microprocesador Rabbit 3000 ya que el mismo posee todas las funcionalidades necesarias.

Para el desarrollo del *Telemeter* se utilizaron:

- 1 puerto serie RS232 con control de flujo para la comunicación de consola

- 1 puerto serie RS232 con control de flujo para la comunicación remota
- la funcionalidad de “input capture” que permite medir tiempos de duración de pulsos para la recepción de la señal infrarroja proveniente del control remoto
- la funcionalidad de PWM “pulse width modulation” que permite generar pulsos de duración variable para la generación de la señal infrarroja para manejar un eventual sintonizador externo (canalera codificada)
- líneas de control para la implementación de un bus I²C⁷ para comandar el tuner de la canalera (cambio de canales mediante comandos al sintonizador) así como la generación de comandos para el control de volumen
- líneas de control para el comando del chip de OSD (On Screen Display) que permite insertar caracteres en la pantalla del televisor. Este chip se comanda mediante 3 líneas de control con un protocolo propietario del fabricante
- temporizadores (timers) del microprocesador para varias de las funcionalidades requeridas
- líneas de control adicionales para manejar el relay de encendido, los jumpers y salidas adicionales necesarias en el sistema
- interrupciones
- memoria no volátil para almacenar configuración

Con la totalidad de las funciones implementadas se utilizaron prácticamente todas las líneas de entrada y salida disponibles en el microprocesador.

IV. ESTRUCTURA DEL SOFTWARE

Desde el punto de vista del software, el *Telemeter* es un sistema de tiempo real. Está implementado en lenguaje C (con las particularidades del compilador Dynamic CTM de Z-World) [4]. Para la implementación se utiliza el núcleo de tiempo real μ C/OS-II⁸ [6].

El *Telemeter* consta de varias tareas cuya interrelación se puede apreciar en el diagrama de la Figura 3. El intercambio de información entre tareas se realiza utilizando uno de los mecanismos clásicos en los sistemas de tiempo real: las *colas de mensajes*.

Los cometidos específicos de cada tarea se detallan a continuación.

A. MeterTask

Es la tarea central del sistema y la que comanda la operación del mismo. Recibe mensajes de la tarea *IRInTask* correspondientes a las teclas del control remoto accionadas por parte del televidente.

Su estructura es la de una máquina de estados ya que el equipo puede encontrarse operando simplemente como canalera, desplegando un formulario de audiencia o encuesta para el televidente, etc.

Utiliza una serie de temporizadores para determinar el comportamiento del equipo frente en los diferentes estados del mismo (determinación de detención en un canal, tiempo

⁷ Philips

(<http://www.semiconductors.philips.com/markets/mms/protocols/i2c/>)

⁸ Micrium (<http://www.ucos-ii.com/>)

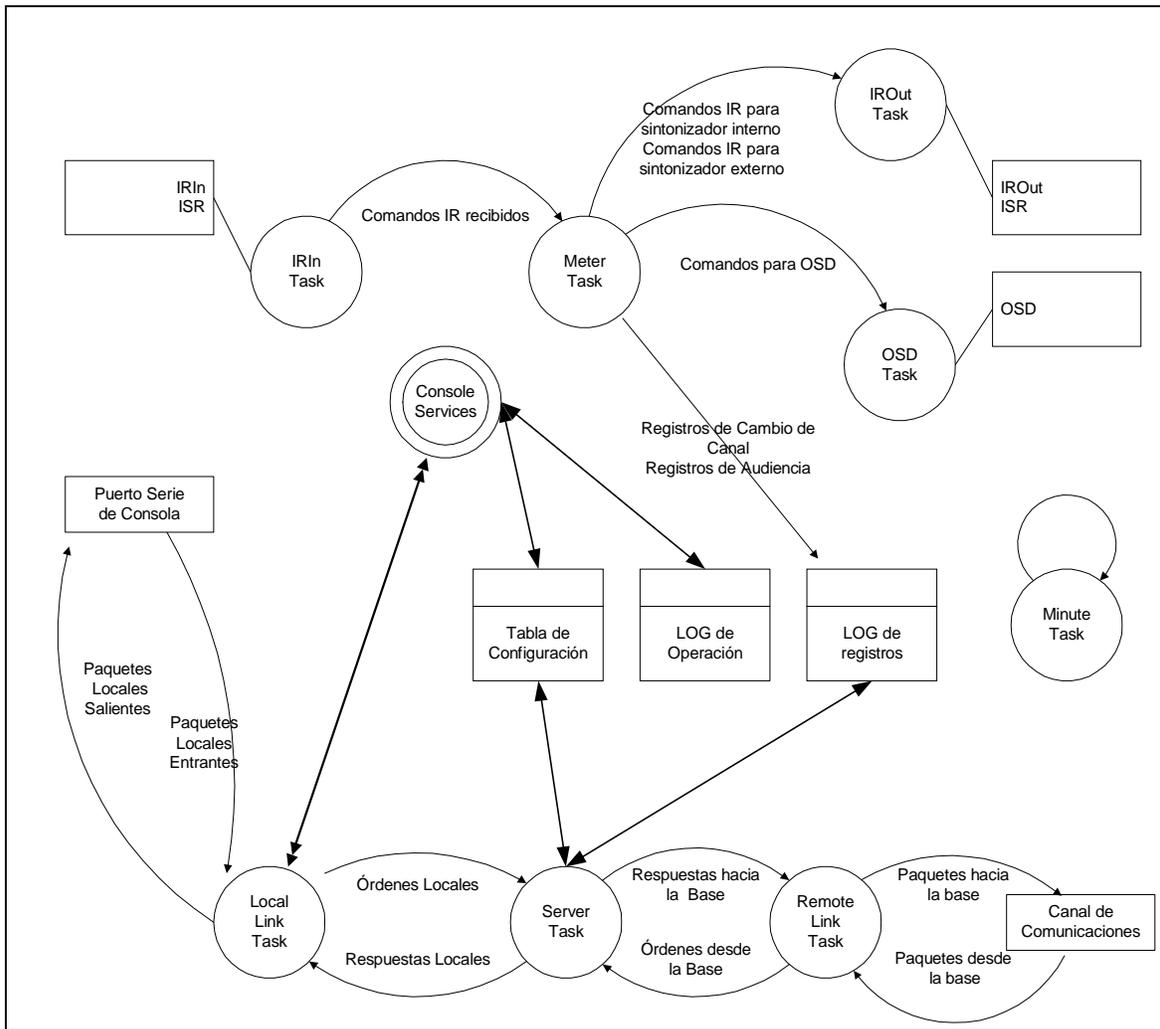


Figura 3 – Diagrama de interrelación de tareas

de espera por respuesta a un formulario, tiempo para borrar mensajes en pantalla, tiempos de repregunta, etc.).

Esta tarea tiene además la importante función de registrar los eventos que se producen en el sistema. Los eventos principales generados por el sistema son los relacionados con la medición de audiencia (registro periódico de qué canal está sintonizado y qué integrante del núcleo familiar está presente frente al televisor). Además de los eventos esenciales para la medición de audiencia, existen otros eventos relacionados con cambios de estado o de configuración del equipo.

Esta tarea dispara además el mecanismo configurable de generación de evento autónomamente que permite la medición de audiencia prácticamente en línea.

B. IRInTask

Esta tarea es la encargada de recibir la señal infrarroja del control remoto y determinar cuál es la tecla accionada por el televidente.

Como se mencionó anteriormente, utiliza la funcionalidad de “input capture” del microprocesador para medir la duración de los pulsos obtenidos por el receptor de infrarrojos. Para esta funcionalidad utiliza una rutina de atención de las interrupciones *IRInISR* que es quien decide

en función de la duración y secuencia de pulsos cuál es el código de la tecla accionada. Con ese código genera un mensaje a *MeterTask*, quien procesará la información recibida.

Para la comunicación entre la rutina de atención a la interrupción y la el cuerpo de la tarea, utiliza un semáforo.

C. IROutTask

Es la tarea encargada de generar la señal infrarroja para un eventual sintonizador externo. Se utiliza cuando el televidente posee un sintonizador externo (televisión satelital, sistemas codificados). En estos casos el usuario opera el sistema mediante el control remoto del *Telemeter* quien sin embargo no utiliza su sintonizador. El sintonizador del *Telemeter* queda en este caso fijo en un canal (normalmente el 3 o 4) y genera una señal infrarroja para comandar el sintonizador externo. Se diseñó un sistema flexible que permite configurar diferentes tipos de sintonizadores externos.

Esta tarea es la encargada de la generación de esta señal infrarroja mediante una salida PWM (Pulse width modulation).

Se utiliza en este caso también una rutina de atención a interrupciones *IROutISR* que se encarga de codificar cada

comando a enviar al sintonizador externo.

Recibe mensajes de *MeterTask* con la indicación de los códigos a enviar.

D. *OSDTask*

Es la tarea encargada de desplegar los mensajes (formularios, canal sintonizado, nivel de volumen, etc.) en la pantalla del televisor. Para esto comanda el chip OSD mediante un protocolo de 3 líneas de control y datos específico del chip utilizado en el diseño.

La tarea recibe de *MeterTask* mensajes indicándole que tipo de orden debe generar para el OSD y los ejecuta.

E. *LocalLinkTask*

Esta tarea tiene por finalidad la atención del puerto serie de consola que como se mencionó anteriormente posee dos funcionalidades.

La tarea está permanentemente monitoreando el puerto serie de consola y si recibe datos los procesa. En caso de tratarse de un comando de línea lo procesa mediante la funcionalidad de “*Servicios de Consola*” y en caso de tratarse de un mensaje del protocolo de comunicaciones definido lo envía a la tarea *ServerTask* quien se encarga de su procesamiento.

F. *RemoteLinkTask*

Esta tarea se encarga de la comunicación remota y está en estrecha vinculación con el módulo de comunicaciones utilizado. Atiende el puerto serie del enlace remoto y comanda el hardware del módulo de comunicaciones utilizado. Utiliza comandos del tipo AT para manejar tanto el módulo CDPD sobre el cual se desarrolló inicialmente el sistema, como los teléfonos TDMA o GSM utilizados en las últimas versiones.

Su implementación se basa en una máquina de estados que refleja las fases de una conexión. Aparece una fase de conexión del equipo a la red, una fase de espera por conexiones provenientes desde el centro de control, una fase de atención de esas conexiones y una fase de intercambio de datos con el centro de control.

Los mensajes del protocolo de comunicaciones definidos son enviados a la tarea *ServerTask* para su procesamiento.

En el caso GPRS se encarga además del registro de IP's dinámicas para posibilitar la conexión desde el servidor central.

G. *ServerTask*

Como se mencionó en la descripción de las tareas *LocalLinkTask* y *RemoteLinkTask*, se definió un protocolo de órdenes y respuestas para interactuar con el *Telemeter*. Este protocolo permite configurar completamente el equipo y obtener información de su estado.

La tarea recibe mensajes de *LocalLinkTask* y *RemoteLinkTask* y los responde para que sean entregados respectivamente por el puerto de consola o por el enlace remoto según de dónde provengan.

Actualmente se han definido alrededor de 50 órdenes. A modo de ejemplo se pueden mencionar algunos grupos de órdenes y respuestas:

- Establecimiento y consulta de: fecha, hora, número de serie del equipo, parámetros de comunicaciones, idioma, nombre de integrantes del núcleo familiar, contraseñas, temporizadores
- Manejo de registros: leer, borrar, buscar
- Obtención de estado: canal actual sintonizado, televidente presente, estado del equipo
- Almacenamiento de configuración en memoria no volátil

H. *MinuteTask*

Es una tarea simple que solamente genera un registro cada minuto para dejar marcas de tiempo en caso de falla del equipo.

I. *Log de operación*

Todas las tareas del sistema generan log de operación. Este log sirve como registro en la fase de desarrollo o para diagnosticar eventuales problemas de funcionamiento. Normalmente se generan mensajes cuando se reciben mensajes inválidos o no esperados, así como cuando pudieran existir valores no admisibles en variables.

Se generan mensajes además de eventos importantes para la determinación de fallas de comunicación o en partes componentes del equipo cuando las respuestas obtenidas no están dentro de las esperadas.

Este log de operación se puede consultar solamente mediante los servicios de consola y está implementado por una cola circular por lo que su tamaño es acotado.

J. *Log de registros*

El log de registros almacena los diferentes tipos de registro existentes en el sistema.

Los registros definidos al momento están en el orden de 20 e indican por ejemplo:

- cuando el equipo es apagado o encendido ya sea con el control remoto o por pérdida de alimentación
- cuando se sintoniza un canal o se abandona el canal sintonizado
- quién o quienes están presentes frente al televisor (respuesta a formulario de audiencia)
- configuración de pregunta de encuesta, respuesta a la misma
- modo de funcionamiento del equipo (normal, juego). Tipo de fuente de señal (aire o cable)
- indicaciones de cambios de configuración al equipo: cambio de fecha y hora, cambio de identificador de hogar, cambio de número de serie
- indicación de registros descargados, indicación de registros borrados

Estos registros permanecen almacenados en memoria RAM del sistema y se pueden consultar mediante servicios de consola o mediante órdenes del protocolo de comunicaciones.

K. Servicios de Consola

No se trata de una tarea desde el punto de vista del sistema de tiempo real. Implementa la interfaz de línea de comandos para la consola.

Permite mediante la conexión de un cable serie el puerto de consola del *Telemeter* dar comandos simples al equipo, principalmente para funciones de prueba.

Se dispone de un conjunto básico de comandos que dan información del estado del equipo y permiten ver el log de operación del mismo o los registros generados.

V. COMUNICACIÓN ENTRE TAREAS

La comunicación entre tareas se realiza utilizando el mecanismo de colas de mensajes provisto por el núcleo de tiempo real μ C/OS-II incluido en la distribución del compilador Dynamic C.

Se definió un formato genérico de mensajes a intercambiar entre las diferentes tareas que consta de un encabezado de tamaño fijo y un área de datos de tamaño variable. En el encabezado se indica el tipo de mensaje, una bandera para indicar cuando se libera el uso de ese mensaje y una referencia a la cola de mensajes de la tarea emisora donde ésta espera la respuesta.

La bandera de liberación de mensaje se utiliza para economizar el uso de memoria en un sistema embebido como el desarrollado. Con este mecanismo, cuando una tarea recibe un mensaje, lo procesa utilizando la misma área de memoria que la tarea que lo emitió. De este modo se evita la necesidad de hacer copias de los mensajes con el consiguiente aumento de uso de memoria. Se corre el riesgo en este encare que la tarea emisora necesite ese espacio para enviar otro nuevo mensaje y éste aún no haya culminado de procesarse por la tarea receptora. Este problema se evita con el uso de la bandera de liberación del área del mensaje y se obliga a que la tarea emisora verifique la disponibilidad del espacio antes de reutilizarlo.

La utilización de un formato uniforme de mensajes simplifica el manejo del intercambio de mensajes y además minimiza los cambios necesarios en caso de requerirse comunicación entre nuevas parejas de tareas.

VI. PROTOCOLO DE COMUNICACIONES

El protocolo de comunicaciones entre la base y los *Telemeters* puede ser utilizado sobre cualquier tipo de red que provea al menos con servicio elemental de transporte de paquetes, como puede ser UDP o SMS.

El protocolo tiene exclusivamente dos tipos de paquetes: órdenes y respuestas. Solamente el centro de control puede generar órdenes, en tanto que las respuestas son generadas exclusivamente por los *Telemeters*. El formato de los paquetes consiste de un encabezado de tamaño fijo, una carga de tamaño variable y una suma de comprobación (CRC). El encabezado contiene los siguientes campos:

- Versión del protocolo de comunicaciones
- Largo total del paquete
- Número de secuencia para control de duplicados
- Identificación de la orden o respuesta.
- Resultado de la operación

Cuando se recibe una respuesta a una determinada orden, se debe verificar que el número de secuencia coincida con el de la orden enviada, descartándose en caso contrario.

Si el CRC no valida también se descarta la orden o respuesta recibida.

El repertorio de órdenes y respuestas permite configurar los modos de operación del equipo, los parámetros de tiempo que determinan su comportamiento y la configuración referente al hogar en el que se va a instalar el equipo.

VII. CONCLUSIONES Y PRINCIPALES PROBLEMAS RESUELTOS

La experiencia de InnoCom en el desarrollo de sistemas de tiempo real, básicamente iniciada con la participación en el desarrollo de los conmutadores de paquetes X.25 *iCPaq* en operación en la red pública de ANTEL⁹ (Urupac), posibilitó un correcto análisis y definición de las tareas necesarias, así como la elección de mecanismos flexibles de intercomunicación entre las mismas.

El sistema se encuentra en producción con algunos cientos de equipos en funcionamiento continuo desde hace más de dos años sin haberse producido fallas de software o hardware, lo que pone de manifiesto la robustez del sistema. Durante el desarrollo del sistema fue necesario resolver una cantidad importante de problemas de diseño e implementación. A modo de ejemplo se mencionan:

- Implementación del bus I2C para comandar el sintonizador y el control de volumen de la canalera.
- Desarrollo de la interfaz eléctrica y de comandos con el sintonizador.
- Desarrollo de la interfaz eléctrica y de comandos del controlador de volumen.
- Desarrollo de la interfaz eléctrica y de comandos con chip de OSD utilizado para la inserción de caracteres (definición de posición, tamaño, fondo de los caracteres, borrar pantalla, selección de norma de TV usada).
- Investigación de las normas de controles remotos para interpretar los códigos generados por el control remoto del equipo y para comandar otros sintonizadores externos. Diseño e implementación de la recepción y generación de códigos infrarrojos con mecanismos basados en interrupciones.
- Implementación de mecanismos de transmisión de datos sobre la red celular utilizando variadas tecnologías y adaptándose a las características de los servicios brindados en el País.
- Investigación de mercado y selección de teléfonos y módulos OEM para la implementación de las interfaces de comunicación diseñadas.
- Desarrollo de un sistema de análisis de performance de la red SMS a efectos de analizar su confiabilidad en relación a mensajes duplicados, perdidos o retardados.
- Diseño e implementación del módulo de comunicaciones GPRS utilizando protocolos estándar PPP y TCP. Se utilizó el stack TCP/IP disponible como módulo del compilador Dynamic C del Rabbit [5]. Al utilizarse un servicio con direcciones IP dinámicas, fue

⁹ Administración Nacional de Telecomunicaciones (Uruguay)

necesario el desarrollo de un mecanismo de registro de los móviles para que informen la IP asignada al servidor central.

- Diseño e implementación del esquema de prioridades para el despacho de las tareas del sistema, aspecto esencial para el correcto funcionamiento de un sistema de tiempo real.
- Optimización del código y las estructuras de datos para adaptarse a las restricciones de memoria del microprocesador utilizado (128K RAM, 256K Flash)

VIII. PARTICIPANTES EN EL PROYECTO

El proyecto fue posible gracias al Ec. Ariel Scarone y el A/P Silvio Barbato directores de Mediciones y Mercado que contrataron con InnoCom el desarrollo de este proyecto y realizaros las especificaciones funcionales del sistema.

La concepción y las especificaciones técnicas iniciales fueron realizadas por el Ing. Omar Barreneche, con participación de los Ings. Juan Pablo Oliver y Gabriel Gómez Sena.

La mayor parte de las decisiones de diseño tanto de hardware como de software fueron tomadas por los Ings. Juan Pablo Oliver y Gabriel Gómez Sena.

El software del equipo fue diseñado e implementado por el Ing. Gabriel Gómez Sena quien además ofició de Jefe de Proyecto durante casi la totalidad del desarrollo del mismo.

El diseño del circuito y de la placa inteligente fue realizado por el Ing. Juan Pablo Oliver, así como el análisis de funcionamiento de la canalera utilizada.

El diseño del circuito y placa correspondiente al módulo de comunicación TDMA fue realizado por el Ing. Leonardo Steinfeld. El diseño del circuito y placa correspondiente al módulo de comunicación GSM fue realizado por el Ing. Andrés Merello de acuerdo a un diseño preliminar del Ing. Leonardo Steinfeld.

El Ing. Víctor González Barbone participó en el desarrollo del servidor de registro de IP's para la implementación de GPRS y en el desarrollo de parte del software para validación del desempeño de la tecnología SMS.

IX. BIBLIOGRAFÍA

- [1] Rabbit 3000 Microprocessor User's Manual, Rabbit Semiconductor.
- [2] Rabbit 3000 Microprocessor Designer's Handbook, Rabbit Semiconductor.
- [3] Rabbit Microprocessor Instruction Reference Manual, Rabbit Semiconductor.
- [4] Dynamic C User's Manual, Rabbit Semiconductor.
- [5] Dynamic C TCP/IP User's Manual, Rabbit Semiconductor.
- [6] MicroC/OS-II, The Real-Time Kernel, Jean Labrosse