

## SISTEMA DE MEDICIÓN “SISMED/C – GLENCROSS” PARA SITIO AISLADO

I. Lescano<sup>1</sup>, M. López<sup>2</sup>, R. Oliva

Área Energías Alternativas - Universidad Nacional de la Patagonia Austral

Lisandro de la Torre 1070 - 9400 Río Gallegos - Argentina

TE +54 2966 442317/19 - FAX +54 2966 442620 E-mail: micro-en@unpa.edu.ar

**RESUMEN:** En el presente trabajo se analiza la implementación de un sistema embebido de medición de elevada robustez para sitios aislados, utilizando un sistema operativo (S.O.) Linux. El equipo descrito tiene por objeto monitorear módulos de energía renovable en una Escuela Rural (Paraje Glencross / Santa Cruz) y es una evolución de los equipos de medición para aerogeneradores implementados por el grupo desde 1999. Si bien se mantiene la tecnología del bus PC/104, la implementación discutida utiliza procesadores ARM de bajo consumo y almacenamiento en memorias Flash. Como contraste con los antiguos desarrollos bajo un S.O. DOS, se enumeran las ventajas y desventajas de emplear Linux, un S.O. de disponibilidad libre y código abierto.

**Palabras clave:** Linux, Sistema Embebido, Sistema Operativo, Registro de Datos, Sitio aislado, Supervisión Remota.

### INTRODUCCIÓN

La Universidad Nacional de la Patagonia Austral (UNPA) se encuentra desarrollando, en convenio con el Consejo Provincial de Educación de Santa Cruz, las etapas finales del Proyecto Federal de Innovación Productiva SC-03/2004 para la construcción de un sistema de suministro energético con energía eólica y solar para Escuela Rural N° 25 Casimiro Biguá (Paraje Glencross, Santa Cruz). Dicho proyecto implementa un sistema de medición de tipo embebido que permite la evaluación continua de la producción de los equipos y facilita su mantenimiento, y se basa en trabajos anteriores desarrollados por el Área Energías Alternativas (Oliva et al., 2000)

El equipamiento a instalar en dicha escuela consistirá de un subsistema eléctrico-eólico y otro de tipo solar térmico. El primero consta de 2 aerogeneradores de imán permanente de 1kW de potencia nominal eléctrica sobre 48V y de origen nacional. Se incluye además un banco de baterías de 660Ah en 48V y un inversor capaz de entregar 3.6kW de potencia. La existencia de un grupo motogenerador diesel en la escuela, hará necesaria la instalación de una llave conmutadora. El segundo subsistema de tipo solar térmico se conectará en serie con el termostanque para el precalentamiento de agua. Se incluyen como parte del equipamiento sensores de efecto Hall para la medición de corriente (también para tensión en el caso de corriente alterna) que permiten medir y registrar las magnitudes de operación de los módulos a través del sistema de adquisición de datos “embebido” denominado SISMED/C -Glencross.

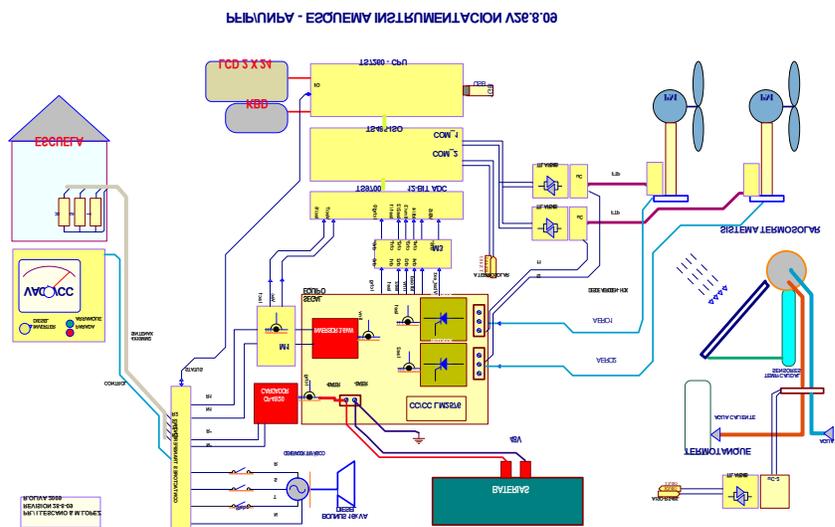


Figura 1 - Diagrama general de la instrumentación SISMED-C / Escuela Rural Glencross

Un sistema embebido es un dispositivo diseñado para un fin específico, generalmente de costo, tamaño y consumo energético dependiente de la aplicación para la cual se lo diseña. Se basa en un microcontrolador o microprocesador como unidad central de procesamiento (CPU) y en un conjunto de subsistemas o periféricos adicionales comunicados por distintos buses

<sup>1</sup> Estudiante de Ing. Electrónica ([mmaquina@gmail.com](mailto:mmaquina@gmail.com)).

<sup>2</sup> Estudiante de Ing. Electrónica ([martinlopez83@hotmail.com](mailto:martinlopez83@hotmail.com)).

<sup>1,2</sup> Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario.

para lograr la funcionalidad requerida. En contraste, una computadora personal (PC) no se diseña para un uso específico, y de hecho es un dispositivo de propósito general. Las primeras unidades de SISMED se diseñaron en torno a la arquitectura x86 de 16bits, preponderante en las PCs, aunque en un formato “embebido” o industrial muy compacto conocido como PC/104 y utilizando una versión DOS de 16 bits complementada con el kernel de tiempo real uC/OS-ii (Labrosse, 1999). La nueva unidad SISMED/C se diseña utilizando el mismo formato PC/104 pero con la arquitectura de procesadores ARM de 32 bits, de menor costo y consumo que los x86, y adaptada específicamente a los S.O. multitarea como el Linux, al contar por ejemplo con una unidad de manejo de memoria (MMU) en hardware. La arquitectura ARM mencionada tiene la ventaja adicional de que ha sido licenciada a múltiples fabricantes, lo cual contribuye a una reducción de precios tanto en los componentes como en las herramientas de desarrollo.

## CARACTERISTICAS GENERALES Y APLICACIÓN DE SISTEMAS OPERATIVOS EMBEBIDOS

Por una parte el constante aumento en la capacidad de memoria y procesamiento de los microcontroladores disponibles en el mercado (a precios razonables) y por otra parte la creciente complejidad de las funciones que se exigen a los sistemas embebidos han impulsado un aumento en la cantidad de aplicaciones embebidas corriendo sobre un S.O. (Wang et al, 2001). La utilización de un sistema operativo como Linux (Japenga, 2009) en esta aplicación tiene como ventajas (además de su gratuidad), lo siguiente: a) Es robusto y viene en sucesivas mejoras y revisiones desde 1991. La enorme cantidad de usuarios hace que el sistema esté en constante revisión y el hecho de que el código sea abierto facilita la corrección de errores (bugs). b) Existen gran cantidad de aplicaciones disponibles. Esto reduce drásticamente el tiempo de desarrollo de cualquier proyecto. c) Proporciona un soporte seguro para redes. d) Está disponible para una gran cantidad de plataformas y cuenta con una extensa base de desarrolladores. e) Provee soporte para una gran cantidad de periféricos. f) Es modular, lo cual significa que sólo se carga lo necesario, pudiendo lograrse una excelente relación recursos requeridos (memoria volátil y no volátil) contra funcionalidad obtenida.

En contraposición, podrían señalarse como *desventajas* las siguientes: 1) una mayor demanda de recursos que el desarrollo equivalente sin S.O. 2) El desarrollador debe poseer o incorporar conocimientos de Linux además de los requeridos para hacer funcionar el hardware sin S.O. Si bien para una aplicación relativamente compleja (ej. mostrar una página web) el tiempo de aprendizaje empleado por el diseñador en Linux es ostensiblemente menor al que consumiría construir el andamiaje necesario para lograr la misma función sobre el hardware “directo”, algunas funciones que resultaban triviales en ausencia de S.O. (ej. poner en uno o en cero una salida digital) pueden no ser tan sencillas bajo Linux. 3) El hecho de que esté en constante revisión resulta un arma de doble filo, pudiendo implicar que la existencia de una gran cantidad de versiones (no necesariamente compatibles entre sí) de las herramientas disponibles dificulte la compilación y funcionalidad del programa final. 4) No es un sistema operativo estrictamente de tiempo real (de los conocidos como “hard” Real Time Operating Systems), aunque existen variantes (Real-Time Linux) en desarrollo y aún no demasiado estables: para algunas aplicaciones de control embebido esta característica puede ser imprescindible.

En este caso particular, se consideró más favorable la utilización de Linux como S.O., aún existiendo alternativas intermedias como el **FreeRTOS** (FreeRTOS Project, 2009), un S.O. de tiempo real “minimalista” en el sentido de que provee tan sólo las funciones básicas características de un S.O. multitarea: tareas, mutexes, semáforos y colas (*queues*) entre otras.

## ESTRUCTURA DEL SISTEMA

### a) Hardware

El hardware base utilizado es la placa TS-7260 de formato PC/104 (de Technologic Systems, TS), que cuenta con un microcontrolador Cirrus Logic EP9302 (basado en el “core” ARM920T), implementa el bus PC/104 de 16 bits, y cuenta con un puerto Ethernet, dos puertos seriales (UARTs) y dos puertos USB (Figura 2). Asimismo se adquirieron para ser conectadas a la CPU través del bus PC/104 las placas TS-485 de comunicación (dos puertos RS485 optoaislados) y TS-9700 de conversión A/D (12bits, 8 canales) y D/A. Como interfaz con el usuario se dispone de un display LCD de 2x24 y un teclado que se conectan a la placa TS-7260 mediante puertos de entrada/salida específicos. La empresa TS provee placas de expansión que utilizan el bus PC/104 con una amplia gama de funcionalidades (Technologic S.,2009) que permiten la escalabilidad del sistema o la eventual reutilización del código para otros proyectos.

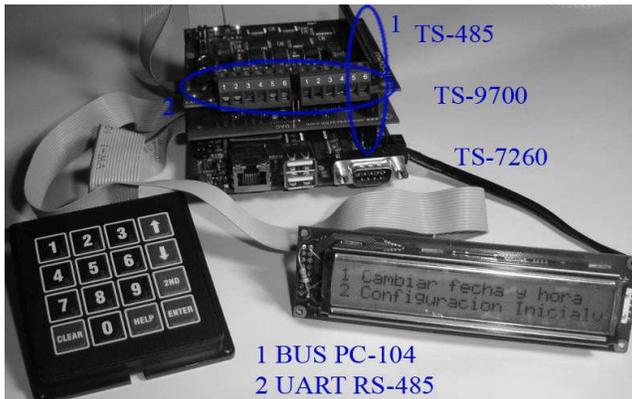


Figura 2 - Foto de los componentes principales SISMED-C

### b) Funcionalidad

El sistema adquiere, utilizando controladores muy elementales enlazados en dos buses RS485, señales de temperatura de entrada y salida del agua del calentador solar, su caudal, y las RPM de ambos aerogeneradores. A través de la placa TS-9700 mencionada se toman señales de corriente y tensión, tanto de continua como de alterna, a efectos de conocer los flujos de potencia entre los distintos componentes. Los datos registrados, después de un procesamiento estadístico se almacenan en una memoria Flash USB estándar, lo cual permite hasta 2GB de capacidad de almacenamiento. El display LCD y el teclado permiten la visualización y modificación de parámetros de configuración sin necesidad de conectarlo a una computadora. Aunque aún no está resuelta la conectividad de la Escuela a Internet (pero probablemente sí en

el futuro próximo), el SISMED/C implementa un servidor web, para acceder a una página con información del funcionamiento del sistema y eventualmente configurar el mismo de manera remota.

### c) Software

El programa, escrito en lenguaje C y compilado bajo gcc, consta principalmente de cuatro tareas que corren en paralelo:

- **HMI:** Controla las entradas/salidas utilizadas para lectura del teclado y envía información al display LCD. Contiene la lógica del menú, mediante la cual el usuario tiene la posibilidad de visualizar algunos datos y cambiar distintos aspectos de configuración (intervalos de registro o *logging* y muestreo, dirección IP estática o dinámica, Memoria USB Flash por defecto, hora y fecha, retirar *pen-drive* con seguridad, borrar memoria interna, recuperar configuración por defecto).

- **IO:** Realiza las funciones de entrada salida de datos, tanto analógicas y digitales como de comunicación. Con un período configurable, este proceso realiza un muestreo de los datos de campo y los guarda en memoria compartida.

- **DATA LOGGER:** Básicamente, a intervalos también configurables, toma las muestras almacenadas por la tarea anterior, las procesa estadísticamente para obtener medias, desvío estándar, máximos y mínimos, y las guarda en la memoria USB-Flash con una *estampa de tiempo* y un número de orden. Antes de guardar el registro, controla si está conectado el dispositivo USB seleccionado como “de almacenamiento”, y si no lo está guarda los datos en la memoria flash interna, pero al momento de reconectar el USB vuelca automáticamente todos los datos y vacía la memoria interna.

- **WEB INTERFACE:** Toma los datos de memoria compartida y los adapta para que el servidor Web pueda interpretarlos. A través de una tarea complementaria llamada *Inadyn* (de código abierto), se logra el acceso remoto vía un browser a través de la asignación dinámica de una dirección IP. Además, se implementa el comando *telnet* usual para reemplazar los accesos vía teclado en caso de necesidad.

El esquema siguiente muestra la relación entre los diferentes componentes. Además de las tareas mencionadas se observa la interacción entre ellas, con las secciones de memoria y con los componentes hardware. Como se puede notar, todas las tareas

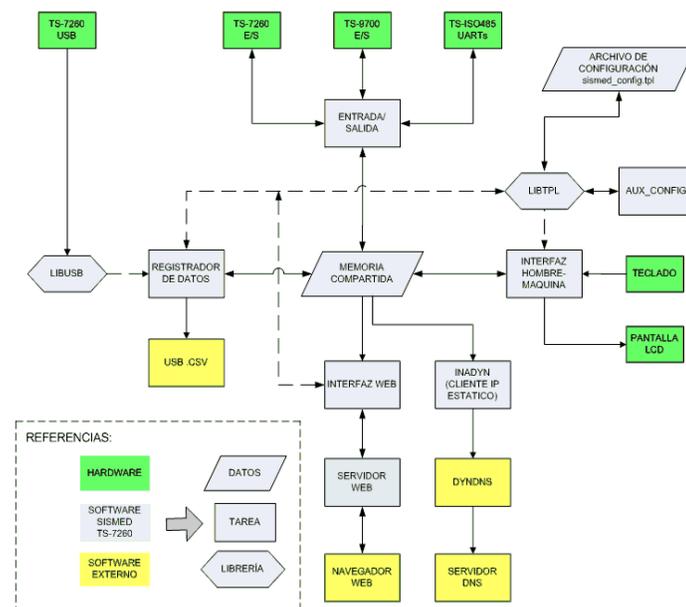


Figura 3: Esquema general del Sistema de Medición SISMED-Glencross<sup>(5)</sup>

leen y/o escriben (con control de accesos) en memoria compartida, donde se almacenan muestras y la configuración del sistema. Para no perder información ante fallas de energía, se envía dicha memoria periódicamente a almacenamiento no volátil, utilizando la librería en C denominada *LibTpl*. Otra librería utilizada es *LibUSB*. La misma permite leer información de los dispositivos conectados a los puertos USB desde el *espacio de usuario* sin realizar llamadas al S.O. Se lee el tipo de dispositivo, fabricante, número de serie, etc. Con esto, automáticamente se detecta cualquier cambio en dichos puertos, de manera de escribir los registros en el lugar correcto y no perder datos.

### CONCLUSIONES

El presente trabajo enfoca avances y aspectos de diseño relacionados con sistemas de toma de datos para energías renovables, fundamentalmente los criterios de selección de S.O. y hardware en los casos en que se decida no adquirir equipos llave en mano sino construir y programar los propios.

### REFERENCIAS

FreeRTOS Project (2009) Pagina principal ([www.freertos.org](http://www.freertos.org))  
 Japenga, B. (2009) Why Use Linux for Real-Time Embedded Systems, White paper published at <http://mtibugs.com/>  
 Labrosse, J. (1999) MicroC/OS-II – The Real-Time Kernel, pp 37-43. R&D Books – Miller Freeman, Lawrence KS, EE.UU.  
 Oliva, R.B., Luna Pont, C.A. (2000). Development and first results of a data acquisition system for low power wind-diesel generators in South Patagonia. Proceedings (CD) de Wind Power – EWEA Conference, Kassel, Alemania, 25/27-9-2000.  
 Proyecto SISMED/C- Glencross (Blog-2008/9) : <http://ts7260.blogspot.com/>  
 Technologic Systems (2009) Periféricos TS7260 : <http://www.embeddedarm.com/products/pc104-peripherals.php>  
 Wang, C.L., Yao, B., Yang, Y., Zhu, Z. (2001). A survey of Embedded Operating Systems. CSE 221 Graduate Operating System Projects, University of California / San Diego (<http://cseweb.ucsd.edu/classes/fa01/cse221/projects/group2.pdf>)

### ABSTRACT

The present work analyses some design aspects of a high-reliability embedded data acquisition system for a renewable-energy supply at an isolated rural school in Patagonia. The system uses ARM-processor hardware and runs Linux, a well-known free, open-source multitasking operating system (OS). Advantages and disadvantages of employing such an OS are discussed, as well as a number of issues related with task and memory distribution.

**Keywords:** Linux, Embedded System, Operating System, Data Logging, Remote Supervision.