

Redes de Microcontroladores: Definición, Implementación y Evaluación

Ricardo A. López, Fernando G. Tinetti¹

Departamento de Informática Sede Trelew, Facultad de Ingeniería – UNPSJB
III-LIDI, Facultad de Informática - UNLP

¹ Investigador Asistente Comisión de Investigaciones Científicas de la Prov. de Bs. As.
lopez.ricardo@gmail.com, fernando@info.unlp.edu.ar

Resumen. Este artículo presenta la definición, implementación y utilización de una red de microcontroladores. Necesariamente se han observado en el desarrollo del trabajo problemas inherentes a las temáticas de redes, sistemas operativos embebidos y sistemas distribuidos. Desde una perspectiva más general, contando con una red de microcontroladores, siempre es posible definir y avanzar en líneas de investigación más relacionadas con los sistemas distribuidos en general, en temas tales como la respuesta a eventos complejos con múltiples puntos de adquisición de datos, con un sistema de tipo SCADA (*Supervisory Control and Data Acquisition*). El desarrollo de un protocolo abierto, sencillo y basado en un modelo referencial, también es un elemento atractivo, que permite mayor maniobrabilidad en el logro de los objetivos planteados para el proyecto dentro del cual se ha desarrollado este trabajo: docencia e investigación.

Palabras Clave: Redes de interconexión, software embebido, sistemas de tiempo real, sistemas SCADA (*Supervisory Control And Data Acquisition*).

1 Introducción

El proyecto dentro del cual se ha desarrollado el trabajo presentado en este artículo contempla la definición y diseño de una red local de bajo costo implementada con microcontroladores. No solamente se define la red independientemente de un microcontrolador en particular sino que también se desarrollan y evalúan hardware y software de un prototipo con microcontroladores específicos como prueba de concepto (*proof of concept*) de la definición realizada [5].

La red se concibe como la interconexión de Unidades de Control (UCs), cada una de las cuales con procesos que interactúan con la aplicación a controlar/monitorear (campo) y a la vez con un Centro de Control de mayor jerarquía a través de la red. Existen dos actividades centrales que deben resolver las UCs: 1) procesos locales (control de lazo cerrado) de interacción con el campo, de mayor prioridad y 2) Procesos relacionados con la red: sincronización, y de eventos relacionados con un SCADA. En la fig. 1, se otorga una disposición de los bloques constitutivos del sistema donde se destacan los siguientes componentes:

1. Microcontroladores, denominados Unidades de Control (UC1, UC2...UCn), que interactúan con una porción del campo y ejecutan su aplicación, sin conexión entre sí, pero todas ellas vinculadas a un dispositivo que las concentra.
2. Un microcontrolador con la función de Concentrador de Comunicaciones que se vincula a las UCs mediante una red multipunto.
3. Un motor de Control de Supervisión y Adquisición de Datos (SCADA), con su propia aplicación que corre, por ejemplo, sobre una PC.
4. La Red 1, que vincula las UCs con el concentrador y la Red 2, que vincula el Concentrador con el centro de control.

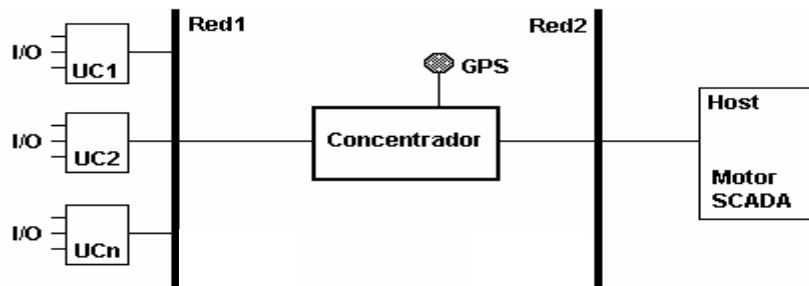


Fig. 1. Topología de la Red de Microcontroladores y Módulos Asociados.

Como en todo sistema distribuido, se deben tener en cuenta características tales como ausencia de tiempo global, pérdida de mensajes en las comunicaciones, y fallo de unidades. Por lo tanto, muchos de los razonamientos y políticas a implementar serán análogos a efectuados ante este tipo de sistemas. El módulo GPS (Geographical Positioning System) en la fig. 1, en particular, aparecer específicamente con el propósito de contar con un tiempo de referencia común en el sistema distribuido.

2 Requerimientos y Definiciones

Relevando distintos sistemas SCADA, y tomando como referencia el esquema de organización del sistema de la sección anterior se explicitan tanto los requerimientos como las definiciones del sistema, que tendrán impacto directo sobre el diseño del hardware y software a proponer e implementar.

Unidades de Control: las UCs tienen la misión de interactuar con el campo de aplicación, correspondiendo las variaciones de cantidad de unidades y configuración particular de cada una con la complejidad de la aplicación [1]. Como criterio general, no exhaustivo, se tendrán las siguientes características mínimas para una UC:

1. Adquisición de estados binarios (Digital Input).
2. Adquisición de valores analógicos (Analog Input).
3. Adquisición de Secuencia de Eventos, con precisión mejor que 1ms.
4. Acciones de control sobre el campo (Digital Output).
5. Al menos un Puerto Asíncrono (UART), para comunicación con el Concentrador de Comunicaciones.

También será posible mediante la selección del componente adecuado, la interacción

con otros dispositivos a través de puertos serie sincrónico y asincrónico, en modo punto a punto o multipunto. Estas facilidades resultan interesantes para conexión con transductores que efectúan la medición de cierta magnitud (temperatura, presión, tensión eléctrica) y se exteriorizan mediante un canal de comunicaciones (típicamente RS232 o RS485) en un protocolo [4] [8].

Concentrador de Comunicaciones: tendrá dos funciones esenciales para el sistema completo: 1) Puente o *Gateway*, entre la Red 1 que lo conecta a las UCs y la Red 2 que lo conecta con el Centro de Control, y 2) sincronización de las UCs de la Red 1, para que sea posible reconstruir una SOE (Sequence Of Events), de todo el sistema a partir de las SOEs de cada UC. La función de puente entre las dos redes, además, permite una configuración como la que se muestra en la fig. 2, donde puede apreciarse cómo las UCs se agrupan en un esquema en árbol con dispositivos concentradores en su raíz. A su vez una agrupación de estos, se vincula al menos con un Centro de Control. Lógicamente, la agrupación de UCs por concentrador en cada Red 1 implica la aparición de un flujo importante de datos que fluye virtualmente desde cada concentrador e impone necesariamente un rendimiento mayor de la Red 2.

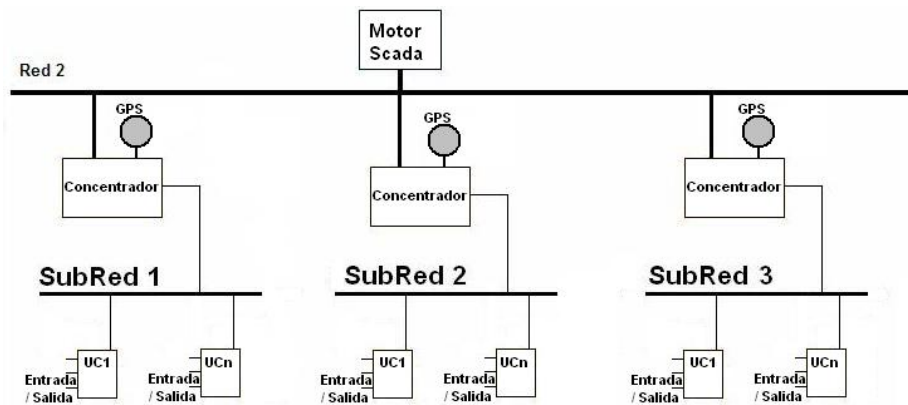


Fig. 2. Topología de Subredes.

La segmentación de la red en subredes mostrada en la fig. 2 es beneficiosa, dado que permite un grado alto de confiabilidad ante fallas en alguna de las subredes y/o sus concentradores. En este punto se hace más importante la utilización del GPS para lograr ahora sincronización externa de todo el conjunto de UCs, dado que con un GPS por concentrador las subredes son totalmente independientes entre sí.

Red 1 (cada subred de la fig. 2): si bien es importante que la red tenga mucha capacidad de transmisión se considera más importante aún, su bajo costo. Esto es debido a que cada UC particular no posee habitualmente muchos puntos de vinculación al campo (menos de 50), por lo que no es un factor importante la velocidad. Por otro lado, esta red debería tener también una confiabilidad elevada y sencillez de instalación. Con estas premisas, se llega casi directamente a la elección de una conexión multipunto con cable balanceado. En principio, conviene recordar que sobre esta red no solamente se transferirán los datos correspondientes a las secuencias de eventos de de cada UC sino también lo necesario para sincronizarlas.

Red 2: en este caso, la variabilidad puede ser bastante grande, dependiendo de la

cantidad de concentradores y de UCs por concentrador. Dado que a priori no se tiene el requerimiento de transporte de datos de sincronización como en el caso de la Red 1, si hay solamente un concentrador, esta red podría ser de muy bajo ancho de banda. Por otro lado, si hay muchos concentradores y cada uno de ellos con muchas UCs, esta red necesariamente tiene que escalar para ser capaz de transportar todos los datos de todas las UCs.

3 Protocolos de Comunicaciones

Teniendo en cuenta las definiciones y los requerimientos dados en la sección anterior, se determinaron dos posibilidades de implementación para la capa física de la Red 1: Ethernet, por ser de amplia disponibilidad [2] y RS485 por tener muchas de las características apropiadas para resolver lo referente a comunicación y sincronización de las UCs:

1. Se implementa con un bus físico y se puede implementar un protocolo maestro-esclavo de manera bastante sencilla para arbitrar el bus.
2. Provee un medio de broadcast físico de señales, que puede ser utilizado para el transporte de mensajes especiales, como los necesarios para la sincronización.
3. Es una de las redes físicas más confiables y con cableado de mayor longitud posible por ser una red balanceada altamente inmune al ruido.
4. Es de muy bajo costo y alta disponibilidad.
5. La utilización de esta norma física es muy sencilla a partir de la disponibilidad de hardware con el que es posible conectar UARTs de las UCs.

Inicialmente, se decide avanzar en la utilización de la red RS485 por ser de más bajo costo que Ethernet y otra característica que quizás sea más apropiada: requiere menos recursos de hardware/procesamiento que Ethernet y la implementación de la pila TCP/IP que normalmente implica utilizar Ethernet a nivel físico. A partir de la capa física se opta por efectuar una definición completa de protocolo teniendo en cuenta los siguientes aspectos:

1. Contar con una definición completamente abierta y no comercial de protocolo.
2. Poder definir soluciones propias y expandibles a medida de la marcha del proyecto.
3. Formular un protocolo sencillo y robusto a efectos de estudio, experimentación y evaluación de rendimiento.
4. Tener mayor grado de libertad para la implementación de un sistema broadcast de sincronización de hora para sistemas embebidos.

3.1 Definición Siguiendo un Modelo en Capas

Se ha buscado desde el principio la mayor simplicidad posible por varias razones:

1. Normalmente a mayor simplicidad de la definición también corresponde menor complejidad de implementación
2. Los microcontroladores a utilizar suelen tener muy poca capacidad de procesamiento y de almacenamiento, por lo que no podría imponérseles una carga excesiva solamente para mantener las comunicaciones.
3. Tanto para investigación como para docencia, conviene tener en principio una

definición sencilla para luego adaptar o hacer más compleja dependiendo de aplicaciones y/o conceptos específicos.

Por lo tanto, se sigue, en principio, el modelo de tres capas [3]: capa física, capa de transporte y capa de aplicación. Esto no significa que puedan luego identificarse bastante claramente elementos o definiciones similares o correspondientes a las capas del modelo ISO/OSI.

Capa de Aplicación: tiene definido lo más necesario para la encuesta del estado y para la configuración de los microcontroladores, además de incluir lo referente a la sincronización de los mismos. En general, se define un protocolo master/slave donde el Centro de Control es el master de la Red 2 y cada Concentrador es el master de la Red 1. Por la estructura jerárquica del sistema (mostrada en la fig. 1 y en la fig. 2), el Centro de Control evidentemente es el que define toda la actividad de las redes del sistema. En la capa de aplicación se distinguen *Comandos*. Estos son entidades de un byte que en muchos casos están seguido de uno o varios bytes que actúan como *argumentos*. Algunos comandos de la capa de aplicación son:

| Nro | Objeto | Descripción |
|-----|-------------------------|--|
| 00h | Toma de estado | Encuesta el estado de una UC. La UC devuelve bytes de su estado de proceso. Estados de los puertos binarios y analógicos definidos. Eventos desde la última lectura. |
| 02h | Puesta en Hora | Este comando lo emite el concentrador a la dirección de broadcast (FFh). Pone en hora las UC de la subred hasta nivel de décimas de milisegundo. No hay respuesta de UCs. |
| 03h | Control | Este comando pone en I/O un bit de un puerto binario. Posee como argumentos la dir. del puerto, nro. de bit y estado a establecer. Se devuelve 0/1 por éxito o no de la operación. |
| 08h | Escribe datos en EEPROM | Este comando posee como argumentos dir. de inicio, cantidad y bytes a escribir. Como respuesta se recibe 0 o 1 indicando el éxito o no de la operación |
| 09h | Lee datos en EEPROM | Este comando posee como argumentos dir. de inicio, cantidad y bytes a leer. Como respuesta se recibe la cantidad de bytes leídos. |
| 0Ah | Toma de hora/sesgo | Este comando encuesta la Hora de la UC. Se devuelve la hora real y el sesgo calculado de la última puesta en Hora. |

Capa de Transporte: en la capa de transporte se define todo lo necesario para que las comunicaciones de la capa de aplicación se realicen correctamente tanto en la secuencia de mensajes como en el control de errores de que podrían suceder en capas inferiores de la red. La fig. 3 muestra esquemáticamente el segmento de capa de transporte, que es similar a otros protocolos de esta misma capa de referencia, donde:

- Sof (Start of frame), tiene como finalidad indicar el inicio del segmento. El valor de este byte se ha elegido como FEh, por ser un valor de ocurrencia poco frecuente se utiliza como preámbulo y determina que todos los nodos de la red que estén en estado de recepción (sólo uno de los nodos está en transmisión), sincronicen el comienzo de lectura a partir de esta señal de inicio.
- Qty (Quantity) es utilizado para que el receptor / emisor, pueda determinar cuántos bytes posee el marco y en consecuencia, dar por concluida la recepción /

transmisión cuando esta cantidad sea alcanzada.

- Dst (Destination) define la dirección de red de cualquiera de las UCs que sea destino del mensaje. El valor especial FFh se utilizará para mensajes broadcast. Un ejemplo de utilización de esta dirección lo otorga el comando de Puesta en Hora (02h) explicitado anteriormente. Este comando es emitido a la dirección de broadcast, asumiendo que todas las UCs del canal lo reciben. Para ello el concentrador efectúa un silenciamiento preventivo de 50ms del canal, a efectos de asegurar el éxito de recepción del mensaje.
- Src (Source) define la dirección de red del nodo que actúa como emisor.
- Sec (Secuencia), utilizado para determinar el número de secuencia del mensaje.
- Dato (*Pay-Load*) para alojar los datos de la capa de aplicación.
- BCC (Check-Sum) es conformado por el emisor efectuando el complemento a uno de la suma módulo 64K de cada par de bytes componentes del segmento tomándolos de valores no signados de 16 bits. El campo BCC la cantidad Qty los incluye. El receptor efectuará la suma de todas las palabras del segmento y el resultado deberá ser igual a FFFFh para un segmento libre de errores

| Denominación | Sof | Qty | Dst | Src | Sec | Dato | BCC |
|----------------|-----|-----|-----|-----|-----|---------------|-----|
| Cant. de bytes | 1 | 1 | 1 | 1 | 1 | 1 a 246 bytes | 2 |

Fig. 3. Formato del Segmento de Transporte.

Los elementos sombreados en la fig. 3 corresponderían a lo que es propio de un segmento de capa de transporte en el contexto de las capas de referencia del modelo ISO/OSI. Los campos que no están sombreados, sin bien están definidos dentro del segmento de transporte, se pueden asociar también a lo que sería específicamente la (sub)capa de enlace en una red con broadcast físico.

Capa Física: la Red 1 se orienta hacia la norma RS485 por las razones expuestas de simplicidad, disponibilidad y bajo costo. La Red 2 se orienta hacia la norma Ethernet dado que, por un lado es de esperar que los concentradores tengan mayor capacidad que las UCs y puedan contener, por ejemplo, la implementación de la pila TCP/IP. Por otro lado, Ethernet escala correctamente para contener los mensajes de múltiples UCs a través de Concentradores en un esquema como el de la Fig. 2.

3.2 Confiabilidad: Time-Outs y Verificación de Contenidos

Debido a que toda la operatoria se ha definido en términos del esquema Master/Slave, donde finalmente el Centro de control fija el sincronismo del flujo, se instrumentan diferentes Time-Outs en cada uno de los nodos:

1. En el nodo Master: un Time-Out que fija la ventana temporal de respuesta del Slave. Si el Slave no responde dentro de la ventana prefijada, el Master repite el mensaje con el mismo número de secuencia. Si no hay respuesta a tres mensajes consecutivos de la misma secuencia, se considera el nodo en falla.
2. En el nodo Slave: el Slave sólo responde cuando se recibe un mensaje consistente. Por ende, no hay requerimientos de re-envío por parte del Slave. A efectos de incorporar los bytes que componen un marco de enlace en recepción, el Slave lanza

un Time-Out a partir de la recepción del byte de preámbulo. Este Time-Out se reinicia ante la recepción de cada byte. Antes que se reciba la cantidad de bytes indicada por Qty, si entre byte y byte hay un tiempo mayor que el de duración de la transferencia de aproximadamente 3 bytes, el mensaje completo se descartará, iniciando nuevamente la secuencia de espera de un byte de preámbulo.

Puede observarse entonces, considerando el funcionamiento cooperativo de ambos Time-Outs que, si el Slave aborta la recepción, ya sea porque no recibe el mensaje en tiempo (time-out) o que calculado el Check-Sum (BCC) el mismo no fue satisfactorio, dicho Slave no emitirá ninguna señal. Esto conllevará a la misma acción por parte del Master: repetir el requerimiento luego de vencido su propio time-out, por asumir que la demora excesiva surge de algún problema detectado en el receptor y que ya no habrá respuesta por parte del mismo.

Como se ha descrito, las capas inferiores del protocolo cuentan con mecanismos que permiten convertir en fiable el mensaje producido hacia la capa de aplicación: 1) Preámbulo indicador de comienzo del marco de Enlace, 2) Time-Out entre byte y byte en la capa de enlace, 3) Check-Sum como trailer en el segmento de transporte, y 4) Número de secuencia en el segmento de transporte. Se entiende así que con ellos, sumados a la calidad inherente que posee un canal de difusión con par trenzado, balanceado y blindado en capa física y con un protocolo que arbitra la ocupación del canal (lo que redundo en una muy baja probabilidad de colisión), se tiene un esquema de comunicación muy robusto y muy seguro

4 Definición del Software

Aunque con tareas específicas bastante diferentes, tanto en las UCs como en el Concentrador se implementa un sistema procesos concurrentes que podría ser considerado un cuasi - paradigma de sistemas operativos: multitarea colaborativa. En estos sistemas embebidos, cada tarea a realizar se concibe como una parte de un programa general, que debe ser resuelta en un tiempo relativamente corto y balanceado frente al resto de las tareas, evitando la inanición de las demás tareas.

Las UCs son necesariamente las que deben tener mayor flexibilidad, dado que son las que están directamente conectadas al campo de aplicación y deben implementar el control de lazo cerrado específico. Estas tareas de control no se pueden definir en general y, por lo tanto, el software se estructura en dos áreas o conjuntos de rutinas, con diferente jerarquía:

1. El Area del Sistema: una serie de rutinas que ofician de BIOS (Basic Input Output System) de la unidad. Estas rutinas son invocadas por el programa principal o las interrupciones, de forma tal que cumplen el rol de servicio.
2. El Area de Usuario: las rutinas que conforman la aplicación específica de usuario, objeto principal de la implementación. Estas rutinas de usuario, pueden ser las que utilizan las rutinas del sistema, tomándolas como servicios para su procesamiento.

En todos los casos, el software de las UCs es totalmente conocido y con cotas de tiempo de procesamiento muy bien definidas tanto para las operaciones de entrada / salida (E/S) como para las de control de lazo cerrado. Las operaciones de E/S se realizan vía interrupciones y la implementación de los protocolos definidos es

sencilla. De hecho, quizás lo más complejo sea el control de errores en cuanto al cálculo de BCC y control de time outs. El cálculo de BCC se lleva a cabo básicamente de instrucciones lógicas, que todo microcontrolador posee muy bien implementadas en hardware y no se necesitan rutinas de software. Por otro lado, los time outs no representan ningún problema especial, dado que los microcontroladores también se caracterizan por poseer al menos un *timer* con capacidad de interrupción que sin lugar a dudas simplifica la implementación.

El Concentrador, por otro lado, tiene quizás dos extremos muy definidos, dependiendo la cantidad de UCs y la distribución de las UCs en Concentradores. El caso más complejo para el software de los Concentradores está dado en la fig. 2, donde hay múltiples concentradores y cada uno de ellos de alguna manera “virtualiza” y sincroniza un conjunto de UCs. En este caso, el Concentrador tiene dos tareas bastante específicas y diferenciadas: a) puente entre la Red 2 y la Red 1 y b) sincronismo de las UCs de la Red 1 de la que es Master, posiblemente utilizando un dispositivo GPS que le provea el tiempo UTC. En el otro extremo, el caso más sencillo podría reducirse casi exclusivamente a un puente entre las dos redes, posiblemente con la delegación del sincronismo directamente en el Centro de Control.

A los efectos de investigación y docencia, se ha asumido que será necesario que el Concentrador sea complejo, aunque a priori no se haya definido una aplicación específica que así lo requiera. De esta manera, se llega a asumir que el Concentrador tiene la función de puente Ethernet-RS485 y la función de sincronizar todas las UCs usando para ello un dispositivo GPS que provee la hora UTC (fig. 2). Para la comunicación sobre Ethernet se ha asumido disponible una implementación de TCP/IP sin pérdida de generalidad dado que la mayoría de los microcontroladores posee una [9].

Es importante recalcar que los protocolos definidos en la sección anterior determinan un método de comunicaciones *half-duplex* y en consecuencia el sistema de comunicaciones a implementar se encuentra en determinado momento en fase de recepción o de transmisión. Por lo tanto, las áreas de memoria necesarias para la constitución de los *buffers* de cada capa, pueden superponer. Esto es posible por el hecho que sólo un *buffer* es utilizado a la vez, reduciendo el uso de memoria, que en los microcontroladores de baja gama (como las UCs), es un recurso escaso.

El software relativo al Centro de Control, en principio, se puede considerar estándar en términos de un SCADA. Lo que es en sí mismo interesante de esta parte del sistema es contar con un SCADA no propietario y orientado a open-source y multi-plataforma. Si bien estas características quedan fuera del alcance de este artículo, es importante recalcar que todo lo relativo al Centro de Control y su interacción con las UCs a través de los Concentradores está definido de manera independiente del hardware por los protocolos, como se recomienda en todo sistema distribuido.

4 Implementación de Referencia (*Proof of Concept*)

Como escenario de recreación de lo expuesto en las secciones anteriores, se ha considerado muy necesario el desarrollado de una implementación completa de

referencia. La Fig.4 otorga un esquema funcional de la placa de hardware.

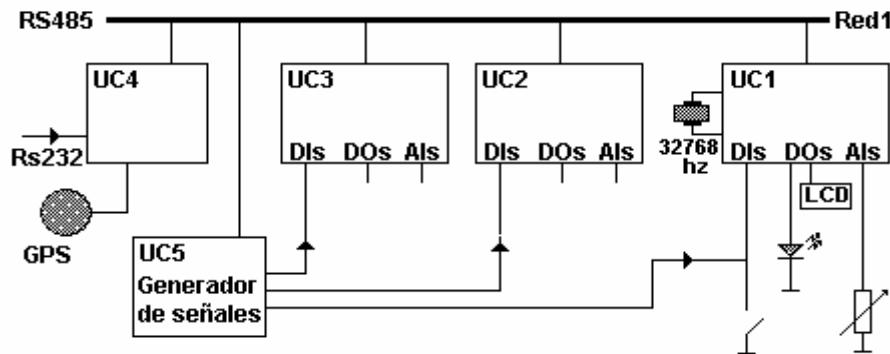


Fig. 4. Esquema de la placa Prototipo.

Esta placa prototipo contiene cinco microcontroladores de la familia Microchip (PIC 16F877, 16F876 y 16F628 [6]), UC1, UC2 y UC3 son las *unidades principales*. Contienen CPU, capacidad de memoria y dispositivos de E/S suficientes como para las pruebas de software del sistema. En la fig. 4 se destaca como ejemplo para la UC1, un detalle de la periferia que poseen las unidades principales: a) Entradas analógicas b) Entradas digitales y c) Leds como salidas digitales. También se destaca el cristal de precisión para el reloj. La UC1 posee además, un *display* LCD de 2 x 16 caracteres. La unidad UC4 actúa como puente entre una red RS232 [7] y RS485 y posee además una entrada Rs232 o serie TTL adicional para GPS. La unidad UC5 está vinculada a la red y conectada a dos puertos de cada una de las unidades principales. Esto le permite actuar como generador de señales, también para diagnóstico y evaluación. Por ejemplo, se puede provocar un evento simultáneo en cada una de las puertos de entrada de las unidades principales verificando de esta forma sus corrimientos relativos de reloj. Por otro lado, se ha tomado una placa construida con la que se ha desarrollado el software del Concentrador sobre una arquitectura basada en un microcontrolador de la línea microchip 18F97J60 con conexión Ethernet embebida.

Todo el software involucrado y que se describe en la sección anterior se implementó de manera efectiva. Las pruebas más relevantes realizadas son:

- Arranque, Puesta en Hora del Concentrador (utilizando el GPS adosado) y sincronización de las UCs en su Red 1.
- Verificación del Concentrador como puente Ethernet – RS 485, con la emisión de comandos desde el Centro de Control a las UCs y recepción de respuesta.
- Sincronización automática de una unidad respecto al resto, ante re-arranque.
- Verificación de los tiempos programados en cada uno de los movimientos que se habiliten en la Unidad de Control.
- Direccionamiento y emisión de comandos hacia las UCs: emisión de comandos en broadcast, provocación de eventos y recepción en el centro de control y funciones de diagnóstico de la unidad y adquisición mediante comandos del protocolo.

5 Conclusiones y Trabajos Futuros

Se definió un sistema completo de interconexión de microcontroladores que puede ser utilizado casi directamente en sistemas del tipo SCADA. Con este fin se han definido en función de requerimientos relativamente generales para sistemas de este tipo los protocolos de nivel de aplicación, transporte y físico de manera clara y sencilla, con el fin de poder adaptarlos a aplicaciones específicas en el caso de sistemas en producción y también para ser utilizados para investigación y docencia en el ámbito universitario.

Se considera muy satisfactoria la definición anterior, dado que se ha podido implementar casi sin inconvenientes, más allá de detalles casi todos relacionados con características físicas del hardware de implementación. En este sentido, se considera la implementación de referencia no solamente una *proof of concept* de la definición sino una plataforma de desarrollo en sí misma, dado que tiene todas las características implementadas tanto en hardware como en software. Desde la perspectiva de software se debe destacar que en cada microcontrolador no solamente se tienen implementados los protocolos definidos sino también en sistema operativo embebido con multitarea colaborativa sobre el cual se puede seguir trabajando e investigando. También es de destacar la posibilidad de sincronización de todo el sistema por la utilización de mensajes *broadcast* que simplifica la tarea.

Quedan por investigar múltiples detalles y conceptos importantes, como los límites de la escalabilidad, la capacidad de establecer cotas rigurosas de errores de sincronización y pruebas bajo cargas extremas de procesamiento, entre otros.

Referencias

1. F. M. Cady, *Microcontrollers and Microcomputers: Principles of Software and Hardware Engineering*, Oxford University Press, 1997, ISBN: 0195110080.
2. Institute of Electrical and Electronics Engineers, *Local Area Network-CSMA/CD Access Method and Physical Layer Spec. ANSI/IEEE 802.3-IEEE Computer Society*, 1985.
3. A. S. Tanenbaum, *Computer Networks*, 4th Ed, Prentice Hall Ptr, ISBN 0130661023, 2002.
4. Telecommunications Industry Association, "Application Guidelines for TIA/EIA-485-A", *TIA/EIA Telecommunications Systems Bulletin*, 1998
5. F. G. Tinetti, R. A. López, "Redes de Microcontroladores: Definición, Evaluación y Perspectivas de un Sistema Distribuido", X Workshop de Investigadores en Ciencias de la Computación, Fac. de Ing., Univ. Nac. de La Pampa, Gral. Pico, Argentina, Mayo 2008.
6. Microchip Technology Inc., *PICmicro™ Mid-Range MCU Family Reference Manual*, 1997. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/33023a.pdf>
7. Microchip Technology Inc., *Tutorial USART - Using in Asynchronous Mode*, 2001. Disponible en <http://ww1.microchip.com/downloads/en/DeviceDoc/usart.pdf>
8. Texas Instruments, *SN65176B, SN75176B Differential Bus Transceivers*, 2003. Disponible en <http://focus.ti.com/docs/prod/folders/print/sn75176b.htm>
9. Adam Dunkels, *uIP, Networked Embedded Systems group*, Swedish Institute of Computer Science, disponible en http://www.sics.se/~adam/uip/index.php/Main_Page