

# Sincronización Broadcast en Redes Multipunto

Ricardo A. López, Fernando G. Tinetti<sup>1</sup>

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

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

**Resumen.** Este artículo se enfoca en una mejora sustancial al problema de sincronización de relojes de controladores en red respecto a anteriores implementaciones. En esta nueva instancia se utiliza nuevo hardware que posee módulos embebidos con algunas nuevas tecnologías y se efectúan cambios al software para incorporar estas mejoras. En particular, se realiza una profunda experimentación que evidencia el trabajo de mejora efectuado. Teniendo en cuenta la utilización de la red de microcontroladores para adquisición de datos y control, es importante remarcar que el problema de sincronización de relojes es un área activa de investigación en el contexto de los sistemas distribuidos. Debido a ello, el logro de valores de significativa calidad en las mediciones efectuadas, se considera un avance en las técnicas de sincronización.

**Palabras Clave:** Sincronización por broadcast, Sincronización multipunto, Redes de Microcontroladores, Sistemas SCADA.

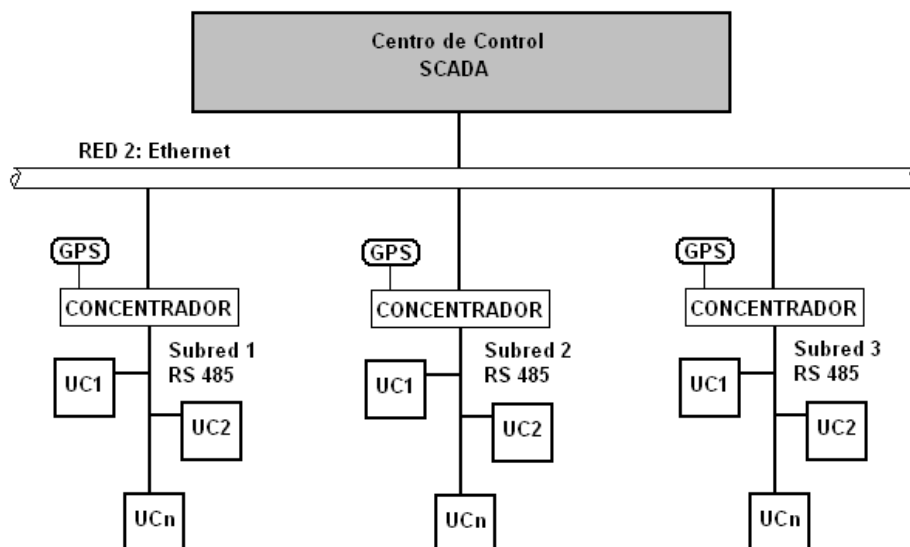
## 1 Introducción

En un artículo anterior [6], se aborda la problemática de sincronización en redes de procesadores. Más específicamente, el trabajo se enfoca en microcontroladores (se utilizarán los términos procesadores y microcontroladores como sinónimos) que se interconectan mediante una red multipunto, por ejemplo bajo norma RS485. En esta red, los procesadores tienen como misión principal la interacción con el campo de aplicación, realizando una tarea general de supervisión y control (SCADA). Este sistema distribuido tendrá una cantidad de nodos acorde a la del sistema a supervisar y/o controlar y es necesaria una sincronización con una precisión elevada, para los procesos que se ejecutan en cada nodo.

Como puede apreciarse en la Fig.1, los dispositivos se encuentran interconectados entre sí y a un dispositivo maestro denominado **Concentrador**, conformando una subred [7]. Varias de estas subredes pueden ser configuradas y a través de su Concentrador es posible vincularlas a una red de mayor velocidad (Red 2), en la que se puede encontrar uno (o varios) Centro(s) de Control.

El Concentrador tiene entonces dos misiones fundamentales: 1) Ser puente o (*gateway*) entre su subred y la Red 2, y 2) Efectuar **sincronización interna** en broadcast en cada uno de los microcontroladores (UC1...UCn) que conforman la subred asociada, valiéndose de un protocolo maestro-esclavo especialmente construido a tal efecto, denominado **Mara-1**. El elemento clave en este protocolo es el Concentrador, que introduce regularmente mensajes específicos de sincronización

entre los mensajes que se transmiten normal y aleatoriamente por la red. El Concentrador además, posee **sincronización externa** mediante GPS.



**Figura 1:** Red y Subredes de Microcontroladores.

Como se anticipara, se han efectuado un número importante de mejoras en el hardware de la instalación respecto a las implementaciones anteriores, entre ellas:

- Utilización de placas industriales multi-capa, con procesadores de 16 bits. En el prototipo del estudio anterior se utilizaron procesadores de 8 bits, sobre una placa artesanal.
- La velocidad de proceso se aumentó a 40 MIPS (con cristal de 80 MHz). Anteriormente se había utilizado una velocidad de 5 MIPS (con cristal de 20 MHz).
- Los procesadores poseen interrupciones vectorizadas, lo que permite otorgar alta prioridad a las rutinas que se relacionan específicamente con la sincronización.
- Se aumentó la velocidad de la red serie multipunto al máximo permitido por las UARTS de los procesadores. El valor establecido fue de 256 Kbps, que guarda una relación de **26,6** respecto a la velocidad anteriormente utilizada (9600 bps).
- La UART de los procesadores posee también mejoras, ya que posee interrupción por varias fuentes específicamente en la transmisión. Es particularmente interesante la interrupción por último carácter completamente serializado.
- Dicha UART también posee un buffer FIFO que suministra caracteres en forma in-interrumpida al registro que efectúa la serialización (Shift Register), esto permite que la trama de un mensaje conexo, sea cuasi-sincrónica y por ende de duración predecible con mucha precisión.

La línea de procesadores utilizados en esta nueva versión del desarrollo es la denominada **dsPIC33** de la firma Microchip [9]. Estos microcontroladores son

pertenecientes a la categoría **DSP** (Digital Signal Processors), que si bien no aportan nada relevante específicamente para sincronización, sí adoptan nuevas tecnologías y mejoras en algunos de sus módulos periféricos (por ejemplo el uso de DMA integrado y asociado a muchos de sus módulos: UART, ADC, SPI, etc.).

Una de las tareas primordiales del sistema es la adquisición de datos (analógicos o digitales) y algunos de ellos que sean definidos como eventos deben tener asociada una marca de tiempo (*timestamp*) en su registro, para que pueda ser posible crear una secuencia de ellos y establecer una relación causa / efecto. Un evento se define como un cambio de estado de una variable asociada. En particular, un evento digital sería el pasaje de 0 a 1 ó de 1 a 0. Un evento analógico, podría definirse como la superación de una determinada banda por encima o por debajo de un estado anterior. En el procesador se programa lo necesario a efectos que ocurra una interrupción de alta prioridad ante la existencia de un evento. La interrupción ejecuta la rutina de servicio asociada en unos pocos microsegundos y en ese tiempo, se lee el valor del reloj de tiempo real (RTC: *Real Time Clock*) contenido en la unidad, se determina el nuevo estado de la variable y todo el conjunto (valor y timestamp), se almacena en una pila de eventos destinada a tal fin.

Como es conocido, los sistemas distribuidos tienen un reloj por cada CPU del sistema, por lo que es fundamental una coordinación entre todos los relojes para tener una referencia de hora única. Por ello, se debe definir alguna representación de tiempo y los dispositivos UC1, ..., UCn de cada subred, deben aceptar sincronización horaria por un dispositivo externo. Se aprovecha que la subred es multipunto, donde es posible emitir un mensaje broadcast que todos los procesadores pueden recibir.

Un factor de mérito deseable es la dispersión de hora que presentan cada una de las unidades respecto a un reloj patrón. Este factor se denomina **exactitud de hora** y los valores usuales se sitúan por debajo de 1 ms. En [6] se había configurado un escenario en el que de hecho se logró una exactitud de +/- **0,16 ms**. La sincronización así definida haría posible presentar una secuencia de eventos ocurridos en un ambiente (industrial o similar), vinculado a microcontroladores de un SCADA, con consistencia independientemente de la unidad que hubiere registrado cada evento. La exactitud de esta secuencia, será determinada por la convergencia que guardaren los RTC de cada una de las unidades a través del tiempo. Es por ello que la sincronización es elemento vital en la presentación de los resultados o en la evaluación del estado del sistema que es presentado al usuario.

Los cristales de los osciladores de cada microcontrolador, típicamente poseen variaciones en su frecuencia que están alrededor de 30 partes por millón (ppm) y como consecuencia todos los relojes sufren un desfase y deben ser sincronizados periódicamente. La sincronización no es trivial, porque se realiza a través de mensajes por la red, cuyo tiempo de transmisión puede ser variable debido principalmente, a variabilidad en el dispositivo que sincroniza por un lado y las propias derivas de cristales de cada unidad por el otro. Dado que todos los dispositivos de la red tienen una deriva de hora, debe elegirse uno de ellos como "patrón" de hora y el resto de los individuos de la red debe ser sincronizado mediante algún procedimiento con la hora que indica ese patrón [3] [4]. El Concentrador funcionará entonces como *gateway* de cada subred y como encargado de la sincronización interna de una cantidad acotada (alrededor de 30) de dispositivos en una zona o área de trabajo. Dado que cada concentrador puede tener asociado un dispositivo que otorgue el UTC (Tiempo Universal Coordinado), como ser un GPS, podría lograrse la llamada sincronización externa de todo el sistema.

## 2 Desarrollo

Un elemento clave para la sincronización es el aprovechamiento de las ventajas de un sistema que podría considerarse de tiempo real estricto, con cotas de tiempo de respuesta a los eventos conocidas de antemano. Esta característica puede considerarse propia de los sistemas basados en microcontroladores [1] [2], donde en muchos casos la cantidad de eventos por segundo es escasa y el software que se ejecuta es relativamente pequeño y rápido y es perfectamente acotado su tiempo de ejecución. A los efectos de obtener una figura de mérito del desarrollo, ha sido importante la experimentación con un entorno real implementado con placas de desarrollo.

### 2.1 Concentrador y Estrategias de Sincronización

El Concentrador es un elemento definido para operar entre las dos redes. Éste hace de puente de comunicaciones entre un Centro de Control situado sobre la red 2, normalmente una red Ethernet de alta velocidad (a la que pertenecen tanto el Centro de Control como todos los concentradores) y las unidades de control, cada una de ellas pertenecientes a una subred normalmente del tipo multipunto.

El trabajo esencial del Concentrador es encapsular los mensajes que se generan en su subred al protocolo de la Red 2 con destino al Centro de Control. De manera análoga, los mensajes que se generan en la Red 2 por el Centro de Control, son desencapsulados por el Concentrador y transferidos a la subred correspondiente. El hardware utilizado para la función de Concentrador posee un módulo Ethernet, así como también una interfase serie RS232 (con conversión a RS485). Además posee una segunda interfase serie más una entrada digital, a efectos de adosar el GPS.

El reloj de tiempo real que tiene el Concentrador para mantener su propia hora es obtenido mediante un temporizador (*timer*) alimentado directamente por un cristal de cuarzo de 32768 Hz. Para la sincronización externa se usa un GPS con PPS (*Pulse per Second*). Este mecanismo consiste en la emisión de un flanco en cada pasaje por cero de un segundo, con una precisión del orden de un microsegundo. Este flanco es recibido por la entrada digital, provocando una interrupción de alta prioridad que inmediatamente pone a cero el contador de fracciones de segundo, redondeando hacia arriba el segundo en caso que el ajuste fuera positivo. Todo ello se lo hace en una rutina "atómica" a efectos de evitar inconsistencias y en este punto se utilizan los mecanismos de interrupciones priorizadas y vectorizadas que poseen estos procesadores, garantizando un período **inferior a 1 us** entre el instante que se detecta el flanco y la puesta a cero del contador aludido. Por lo tanto, el ajuste del *timer* está muy por debajo de su granularidad, que es de 1/32768 de segundo. A continuación, a través del segundo puerto serie, se impone la hora hasta el nivel de segundo. De esta forma el Concentrador obtiene su hora relacionada con el UTC.

Con un período regulable y cuyo valor será precisado en la experimentación, el Concentrador realiza el sincronismo de su subred dependiente. Anteriormente se había utilizado un tiempo de 5 segundos, el cual era suficiente en relación con la velocidad relativamente baja de transmisión del mensaje. En este caso, para emitir el mensaje de sincronización, el Concentrador espera la respuesta a un eventual requerimiento que se pudiere estar cursando desde el Centro de Control, a efectos de asegurarse que la subred multipunto se encuentre en silencio para la emisión del mensaje en broadcast. Cuando tal situación está asegurada, el Concentrador lee su

hora y la emite a la Subred en un mensaje denominado **PeH** (Puesta en Hora) dentro del protocolo Mara-1.

## 2.2 Estrategia de Sincronización Interna

Como se ha observado, se decidió utilizar una característica importante de la red de interconexión que es el broadcast físico de datos. De manera prácticamente simultánea, todos los microcontroladores reciben la información necesaria para la sincronización en un mensaje de sincronismo en broadcast. Con esta metodología, cada unidad de la subred que recibe el mensaje de sincronismo efectúa el ajuste de su reloj interno con el mínimo retraso posible.

Un mensaje de Puesta en Hora (PeH) contiene desde el año actual hasta una fracción de segundo de casi cuatro cifras significativas hexadecimales (equivalente a 30 microsegundos). Para el ajuste de hora de cada UC se cumple:

$$\Delta t = N \times 10 \times 1/Vt \quad (1)$$

$$h_{pdo} = h_o + \Delta t \quad (2)$$

donde:

$\Delta t$  = tiempo de transmisión del mensaje en segundos.

$N$  = Cantidad de bytes del paquete de PeH.

$Vt$  = Velocidad de transmisión en bps, del canal RS485.

$h_o$  = Hora patrón (hora enviada en la PeH).

$h_{pdo}$  = Hora PeH a imponer al destino.

Mientras que la ecuación (1) determina el tiempo que el mensaje tarda en llegar al destino, la ecuación (2) determina la hora corregida a partir de la hora patrón  $h_o$  del concentrador. En el análisis de las ecuaciones se evidencian varios factores técnicos de implementación que es necesario comentar:

1. **Velocidad de la UART:** Esta velocidad en cada controlador está determinada por el cristal principal, que típicamente tiene una precisión del orden de 30 ppm (equivalente al cristal del reloj) y por un timer interno de la UART respondiendo a la siguiente fórmula:

$$UxBRG = Fcy / (4 \cdot BaudRate) - 1$$

En este caso:

$$UxBRG = 40 \text{ MHz} / (4 \cdot 256000 \text{ bps}) - 1 = \mathbf{38,0625}$$

$UxBRG$  es el factor que debe cargarse en el timer de la UART. El mismo debe ser entero, por lo que se debe redondear. Así se tendrá:

$$BaudRate = Fcy / (4 \cdot (UxBRG + 1)) = \mathbf{256410,25}$$

Esta velocidad debe ser fijada tanto en el concentrador como en las UCs y para minimizar el error deben poseer la misma frecuencia de proceso  $Fcy$ . En estas condiciones el tiempo de transmisión del mensaje resulta:

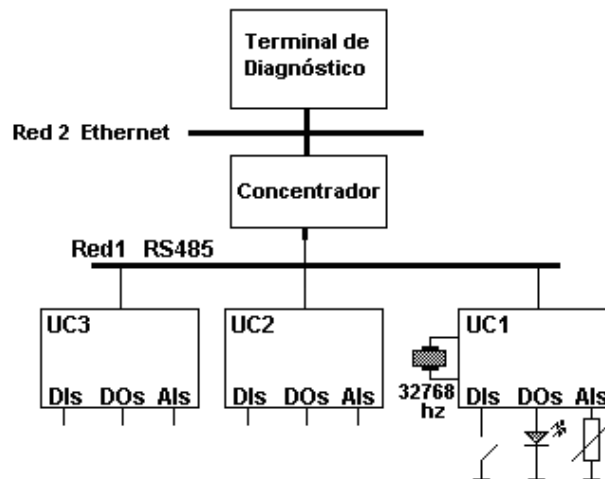
$$\Delta t = N \times 10 \times 1/Vt = 15 \cdot 10 / 256410,25 = \mathbf{585 \text{ us}}$$

Este factor es una constante que debe sumarse en todas las unidades que reciben el mensaje en broadcast e influye solamente en la sincronización

externa y su peso relativo de error disminuye proporcionalmente al aumentar la velocidad de transmisión del mensaje. Se destaca a su vez que el uso de un buffer FIFO de 4 caracteres asociado a la UART, determina que ésta siempre cuenta con caracteres para emisión, por lo cual los mismos salen uno a continuación de otro (el start bit de un caracter se emite inmediatamente a continuación del stop bit del caracter anterior). Ello presupone que el error de cálculo de la duración del mensaje estará en el orden de precisión del cristal y por ende por debajo del microsegundo.

2. **Final del último carácter:** Otro detalle importante es cómo cada una de las UCs determinan cuándo termina el último carácter del mensaje (en particular la duración de su stop bit). En la nueva instalación, una mejora importante es que las UARTs poseen una interrupción por final de serialización del último carácter, que se aprovecha para reconocer el mensaje en broadcast iniciando a continuación la PeH (Puesta en Hora).
3. **Rutina de PeH:** A diferencia del resto de los comandos que poseen un derrotero variado, la rutina que efectúa la PeH en cada controlador se dispara a continuación de la detección consistente del mensaje, de manera tal que es predeterminado el momento en que ésta se ejecuta. Se aprovecha tanto la mayor velocidad de proceso como las interrupciones vectorizadas.

Con el fin de evidenciar las mejoras que suponen teóricamente los cambios propuestos en la descripción anterior, se realizaron una serie de pruebas en un entorno de trabajo que respeta la filosofía esbozada en la fig. 1, con tres unidades de control y un concentrador sobre la base de mismo microcontrolador: el dsPIC33FJ256GP710.



**Figura 2.** Esquema del Ambiente de Desarrollo para Evaluación de Sincronización.

Además, una Terminal de diagnóstico oficia de Centro de Control y emite mensajes mediante el protocolo Mara-1. Dicho entorno se presenta en la Fig. 2 y se destaca como ejemplo para la UC1, un detalle de la periferia que poseen las unidades principales: a) Entradas analógicas, b) Entradas digitales, c) Leds como salidas digitales y un cristal de 32768 Hz que constituye su reloj interno.

El Concentrador está vinculado a la Terminal de diagnóstico montado sobre una PC

mediante una conexión Ethernet. Éste emite cada 2 segundos un comando de PeH cuya estructura resumida es la siguiente:

Denominación	Preámbulo de comunicaciones + Comando	Año, día, Hora, Segundo + Día de la semana	Fracción de segundo	BCC
Cant. de bytes	6	5	2	2

Donde:

- El preámbulo de comunicaciones incluye la cantidad de bytes del mensaje (15) y las direcciones de origen y destino de UCs (en este caso FFh: Broadcast).
- En un bloque de 5 bytes se codifica el año, el día dentro del mismo, la hora, el segundo dentro de la misma y el día de la semana.
- En un bloque de dos bytes se codifica la fracción de segundo (de 0 a 7FFFh).
- El bloque final del mensaje de 2 bytes es el Check-Sum de todo el paquete.

### 3 Experimentación

Se describe a continuación la modalidad de las pruebas realizadas:

- A efectos que la deriva se mantenga en las especificaciones de los cristales se fija en 1 segundo el período de la PeH. De esta forma se garantiza que cada cristal podrá derivar una cantidad inferior a la menor cifra significativa (equivalente a 0,03ms) y por ello los ajustes de hora nunca implicarán retrasar la hora (aunque si en algún caso puedan adelantarla para efectuar el redondeo).
- En la llegada del mensaje broadcast, en cada UC se calcula el sesgo existente entre su hora local y la que impone el mensaje de PeH (un valor positivo indica un adelanto de la UC).
- Se recaba el **estado** de cada UC, con intervalos de 500 ms (con redundancia a efectos de no perder ningún nuevo estado). Entre esos valores de estado de una UC se incluye el sesgo calculado en la PeH anterior y un número identificador de cada PeH, a efectos de comparar sesgos correspondientes para cada UC.
- En el ensayo se tomaron 1200 muestras durante un intervalo de 10 minutos.

#### 3.1 Evaluación de la Sincronización Interna

Se compararon los sesgos correspondientes entre sí y el resultado de caso peor arrojó que las tres UCs diferían entre sí a lo sumo en una unidad, como se presenta en las muestras 520 a 522 del extracto que se presenta en la Tabla 1, para las UCs de direcciones 02, 03 y 04. Como es lógico, cada fracción de la unidad de sesgo en un sentido, se compensa luego en el sentido contrario (puede ser en el ciclo siguiente o en cualquiera ulterior). Esto último se aprecia con las muestras 523 a 525. Debe recordarse que nunca se tendrá sesgo positivo pues significaría que el corrimiento supera la unidad, lo cual se ha asegurado que no ocurra al fijar un período de 1 segundo para la PeH.

El cálculo se condice con una fracción de 1/32768 de segundo hacia un lado u otro de la considerada como centro en este caso. Una unidad del sesgo equivale a +/- 0.03 ms. Si se calcula la excursión máxima del sesgo, se obtiene un valor de exactitud global para el sistema de broadcast de:

$$P = +/- 0.03ms * (1 - (-1))/2 = +/- 0.03 ms$$

Nro. Muestra	Extracto de Trama del mensaje de respuesta de cada UC	Sesgo (Hex)	Sesgo (Dec)
520	02-00 09 02 77 FF FF 03 0A 08 01	FFFF	-1
521	03-00 09 00 77 00 00 03 0A 08 01	0000	0
522	04-00 09 02 77 00 00 03 0A 08 01	0000	0
523	02-00 09 02 11 00 00 03 0A 05 01	0000	0
524	03-00 09 00 11 FF FF 03 0A 05 01	FFFF	-1
525	04-00 09 02 11 00 00 03 0A 05 01	0000	0

Tabla 1. Tramas Resumidas de los Mensajes con los Sesgos de Reloj.

### 3.2 Evaluación de la Sincronización Externa

A efectos de reforzar la apreciación ya observada respecto a la convergencia, al prototipo de diagnóstico se le incorporó un bloque de electrónica que permite la emisión de eventos simultáneos a las UCs interconectadas en red, cuyo esquema funcional se presenta en la Fig. 3. Asimismo, se incorporó lo necesario para registro de eventos en el Concentrador al solo efecto de esta experimentación. Con la finalidad de provocar una pulsación simultánea para las tres unidades, el esquema global fue modificado con el módulo que se presenta en la parte inferior derecha de la Fig. 3, que consta de un circuito integrado que reúne tres compuertas buffer con característica tri-state. Las salidas de las compuertas son aplicadas en paralelo con los pulsadores SW2 a SW4 de cada placa, provocando un evento simultáneo mediante la operación de una única llave SW1. Con este arreglo se asegura que el ingreso de cada evento diferirá en alrededor de 1 nanosegundo, que es el tiempo de propagación especificado por el fabricante para cada compuerta. A los efectos de la granularidad que se dispone para discriminación del tiempo de eventos, el ingreso de eventos con esta nueva modalidad puede considerarse simultáneo. Pulsando el botón y liberándolo, se provoca el registro de dos eventos en cada una de las UCs.

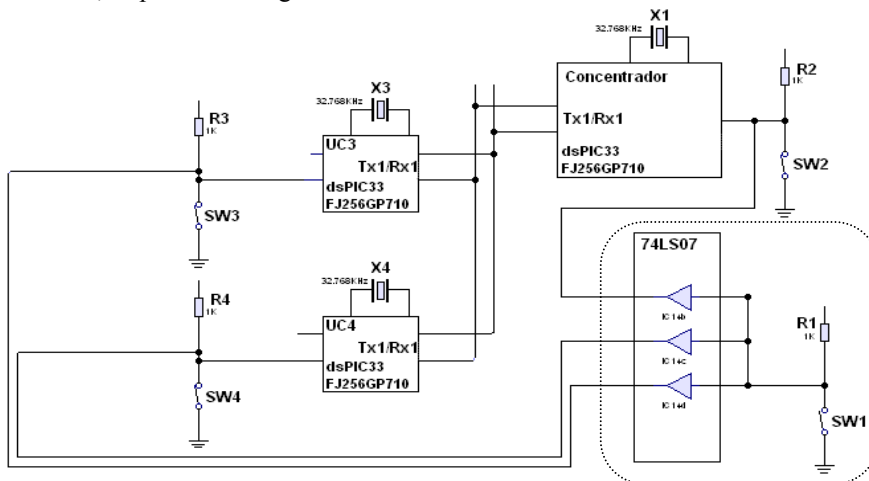


Figura 3: Esquema del Ingreso de Eventos en las UCs.



En efecto, se efectuó la operación que nos arrojó los valores que se presentan en la tabla 2, destacando que los eventos difieren entre sí a lo sumo en +/- 1 (equivalente a +/- 0.03 ms), para la sincronización externa.

Tipo	UC	Puerto	Bit	Estado	TimeStamp
0	01	3	7	0	2011-06-20 15:21:48.3481
0	03	3	7	0	2011-06-20 15:21:48.3480
0	04	3	7	0	2011-06-20 15:21:48.3481
0	01	3	7	1	2011-06-20 15:21:56.1894
0	03	3	7	1	2011-06-20 15:21:56.1896
0	04	3	7	1	2011-06-20 15:21:56.1895

**Tabla 2:** listado de eventos producidos por pulsación simultánea de llaves.

En la Tabla 2 se destacan los siguientes campos según lo establecido en el protocolo Mara-1:

- El Tipo de evento 0, indicativo de un evento de campo.
- La Dirección de cada UC involucrada (el Concentrador siempre tiene dirección 01), con el orden en que les llega la consulta por parte de la terminal (ordenamiento cíclico del round robin).
- El puerto de entrada del evento en cada UC: 03h, relativo debido a la tabla de indirección que relaciona una dirección lógica a una física.
- El Bit (físico) del Puerto.
- El Estado al que va el bit provocado por el evento. Al pulsar se pone el bit a potencial 0 (0 lógico). Al liberar se lo pone a 3.3 volts (1 lógico).
- El Tiempo de ocurrencia del evento o Time Stamp. El experimento fue realizado el 20 de Junio de 2011 entre las 15:21:48 y las 15:21:57 hs. (en ocho segundos).

## 4 Conclusiones

Se ha llegado a una definición y un desarrollo experimental bien definidos de sincronización mediante broadcast en microcontroladores en red multipunto. El valor de exactitud de la Puesta en Hora mediante broadcast es altamente satisfactorio, y en este caso no se puede evaluar más allá debido a la granularidad de los relojes. El sesgo en cualquiera de las unidades respecto al dispositivo que oficia de patrón otorga valores comparables a la propia exactitud del patrón (30 ppm).

Se efectuó también la sincronización de hora mediante un patrón externo de mayor exactitud que provee el UTC, logrando así la sincronización externa de una subred y, de hecho, todas las UCs del sistema distribuido completo. La tabla 3 presenta una comparación de escenarios donde pueden evidenciarse los cambios efectuados al hardware y/o al software para llevarlo al escenario actual y de esta forma justificar las mejoras obtenidas con el sistema.

<b>Cambio en:</b>	<b>Escenario Anterior</b>	<b>Escenario Actual</b>
Hardware	Se evaluó con una placa artesanal con procesadores de 8 bits trabajando a 5 MIPS.	Se evaluó con tres placas multicapas con procesadores de 16 bits trabajando a 40 MIPS.
Duración del mensaje de sincronización	15 bytes–9,6 kbps <b>15,62 ms</b>	15 bytes-256,41 kbps <b>585 us</b>
Período de sincronización	<b>5 s</b>	<b>1 s</b>
Eficiencia del canal.	$1 - [(15,62 + 50) \text{ ms} / 5000 \text{ ms}] = 98,7 \%$	$1 - [0,585 \text{ ms} / 1000 \text{ ms}] = 99,94 \%$
Exactitud de hora	<b>+/- 0,16 ms</b>	<b>+/- 0,03 ms</b>

**Tabla 3:** Comparación de Cambios y Resultados de Experimentos de Sincronización.

## 5 Referencias

1. S. F. Barrett, D. J. Pack, Microcontrollers Fundamentals for Engineers and Scientists, Morgan & Claypool Publishers, 2006, ISBN: 1598290584.
2. F. M. Cady, Microcontrollers and Microcomputers: Principles of Software and Hardware Engineering, Oxford University Press, 1997, ISBN: 0195110080.
3. A. S. Tanenbaum, Computer Networks, 4th Edition, Prentice Hall Ptr, ISBN 0130661023, 2002.
4. A. S. Tanenbaum, M. van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2nd Edition, ISBN 0132392275, 2006.
5. Telecommunications Industry Association, “Application Guidelines for TIA/EIA-485-A”, TIA/EIA Telecommunications Systems Bulletin, 1998.
6. Fernando G. Tinetti, Ricardo A. López, Marcelo E. Gómez, Sebastián Wahler. XV Congreso Argentino de Ciencias de la Computación 2009, Universidad Nacional de Jujuy, Jujuy, Argentina, Octubre 5 al 9 de 2009, ISBN 978 897 24068-4-1. Título: "Sincronización de Microcontroladores en red, Implementación y Evaluación".
7. Ricardo A. López, Fernando G. Tinetti. XV Congreso Argentino de Ciencias de la Computación 2009, Universidad Nacional de Jujuy, Jujuy, Argentina, Octubre 5 al 9 de 2009, ISBN 978 897 24068-4-1. Título: "Redes de Microcontroladores, Definición, Implementación y Evaluación".
8. López R. Protocolos en Redes de Microcontroladores. 1ra Edición - Septiembre 2010. Tesis de Magister: Redes de datos. Director: Fernando F. Tinetti. Publicó: Fac. Informática UN La Plata.
9. Microchip Technology Inc., dsPIC33 Family Reference Manual. Section 17: USART – 70188D. June 2008. Disponible en [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2573](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2573).