XI Congreso Nacional de Teledetección, 21-23 septiembre 2005.
Puerto de la Cruz. Tenerife.

Red de sensores inteligentes para medidas de parámetros ambientales usando una arquitectura distribuida.

Eduardo Magdaleno[(1)], Manuel Rodríguez[(1)] , Alejandro Ayala[(1)], Carles Ferrer[(2)]

[(1)]Dep. Física Fundamental y Exp., Electrónica y Sistemas. Universidad de La Laguna. Tenerife. Avda. Francisco Sánchez s/n, 38200 La Laguna, Tenerife, Canarias. emagcas@ull.es, mrvalido@ull.es.
[(2)]Dep. Microelectrónica i Sistemas Electrònics. Universidad Autónoma de Barcelona. Edificio Q, Campus de Bellaterra 08193 Bellaterra, Barcelona.

Resumen

Hoy en día en el mundo de la teledetección es necesario medir cada vez un mayor número de parámetros medioambientales con el fin de determinar con la mayor precisión posible las condiciones atmosféricas o terrestres de una determinada zona terrestre. El gran elevado número de sensores necesarios para tales objetivos exigen a la industria un bajo coste y una alta calidad de los mismos, lo que conlleva a la necesidad de encontrar unos sistemas de producción más sofisticados.

Sin embargo, el aumento del número de sensores en un sistema distribuido hace que tome gran importancia el sistema de transmisión, por lo que en este trabajo se propone un sistema de comunicación donde cada uno de los sensores inteligentes actuaría como un nodo esclavo, gobernados por un maestro que establecería la comunicación con cada uno de los nodos.

El medio de transmisión está formado por un único canal, permitiendo que la comunicación en un sistema distribuido de sensores pueda realizarse de manera inalámbrica.

1. Introducción

Avances en las tecnologías relacionadas con los sensores, incluyendo los sistemas electromecánicos, procesado de señal, interfaces y redes, han hecho posible construir sistemas versátiles y altamente funcionales denominados sensores inteligentes [1].

Se denomina transductor inteligente a la integración de un sensor o actuador analógico o digital en una unidad que lo contendría junto con un procesador, una memoria y un controlador de red [2]. El sensor inteligente transmite una señal calibrada, checkeada y digital mediante un protocolo de comunicaciones. Estos dispositivos están pensados para trabajar en redes con un número de sensores mediano o elevado, por lo que el conexionado punto a punto no sería adecuado.

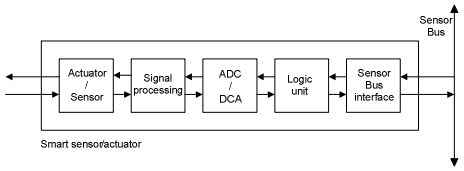

Figura 1: *Esquema de un sensor inteligente para la medida de parámetros ambientales.*

Es por ello que se propone una arquitectura distribuida de dos niveles para reducir costes eliminando una gran cantidad de cableado. En un primer nivel los dispositivos denominados entonces esclavos son conectados mediante un bus de sensores a un nodo maestro, formando lo que se denomina un microinstrumento [3].

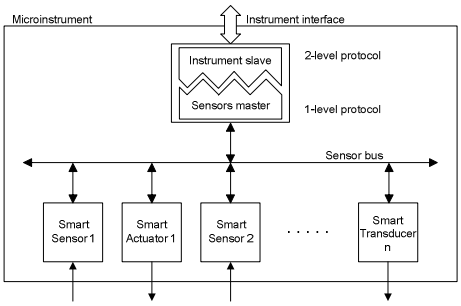

Figura 2: *Esquema de un microinstrumento con los nodos/sensores integrados.*

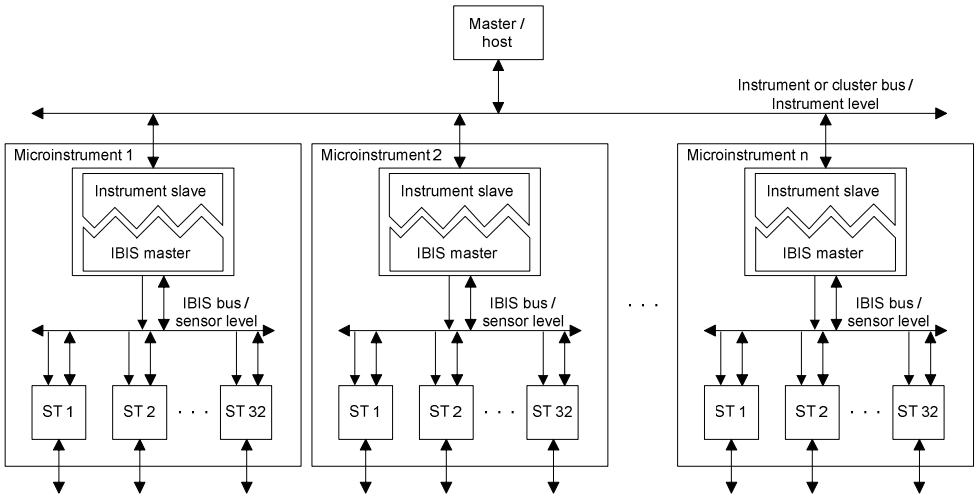

Figura 3: *Arquitectura distribuida*

2. Modelo de arquitectura propuesto

Los sistemas a implementar son muy complejos, difíciles de construir, verificar, reparar y cambiar, por lo que se pretende reducir la complejidad del sistema mediante la táctica de "divide y vencerás". Por tanto, se desarrolla una arquitectura distribuida de dos niveles.

El primer nivel es el nivel de sensores o bus de sensores y contiene los correspondientes sensores inteligentes que toman las medidas de los distientos parámetros medioambientales equipados con su correspondiente interfaz inteligente, agrupados formando un microinstrumento. El bus elegido para este nivel es el bus IBIS [4]-[5] desarrollado en la UAB que permite conectar en cada microinstrumento hasta 32 sensores. El segundo nivel es el nivel de cluster o nivel de instrumentación, que integra todos los microinstrumentos en una red distribuida con un máximo nivel de modularidad.

3. El bus de instrumentación

El bus del segundo nivel implementado se basa en un protocolo maestro-esclavos y es de la clase time-triggered [6] como el bus LIN y el TTP/A [7]-[9]. Un nodo es el maestro activo que provee el sincronismo para los esclavos. La comunicación se organiza en rondas en los que los conflictos se evitan por medio de un estricto acceso al medio por división de tiempo. Cada ronda consta de varios slots o campos que se transmiten con una UART [10] con el objetivo de abaratar el diseño. La transmisión se realiza por un único hilo reduciendo la interconexión entre todos los componentes reduciendo costes y permitiendo el desarrollo de una red inalámbrica.

El bus diseñado se basa en mensajes con el formato que se especifica en la figura 4. Se compone de una cabecera que siempre es enviada por el nodo maestro y un campo respuesta que genera el maestro o uno de los esclavos. La cabecera consta de un campo de ruptura de sincronismo, uno de sincronización y otro de identificación. Cada campo de datos contiene un byte de datos, un bit de start y otro de stop. Así, las funciones del nodo maestro son: transmitir los campos de cabecera y transmitir o recibir el campo respuesta. Las funciones de los esclavos son: detectar la ruptura de sincronismo, medir el ritmo de transmisión con el campo de sincronismo y enviar o recibir los campos de datos y checksum.

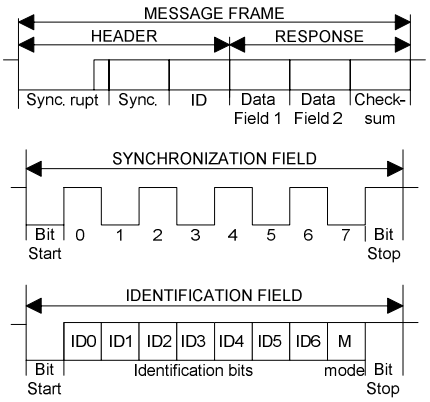

Figura 4: *Formato de los mensajes.*

4. Implementación del sistema

A la hora de desarrollar el sistema descrito anteriormente se ha usado el lenguaje de descripción hardware VHDL. Esta solución permite independencia tecnológica, realizar sistemas escalables, configurables y fácilmente integrables en sistemas jerárquicamente más grandes.

La implementación del protocolo consistió en el diseño de dos bloques: una unidad maestro y otra esclavo.

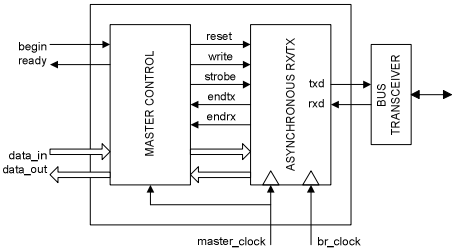

Figura 4: *El módulo maestro*

El módulo maestro consta de dos buses (*data_in* y *data_out*), dos líneas de interfaz serie (*txd* y *rxd*), dos señales de control, *begin* para el comienzo de un mensaje y *ready* para indicar el fin del mismo. El módulo se diseñó con dos relojes: el reloj local del maestro (*master_clock*) y el reloj de ritmo de transmisión (*br_clock*).

La arquitectura del maestro está dividida en 2 bloques que se comunican a través de unas señales de control. El primer bloque es el submódulo Controlador del maestro y el segundo es una pequeña unidad de recepción-transmisión serie asíncrona. El controlador es, básicamente, una máquina de estado que controla los procesos de transmisión y recepción, calculando los bytes a recibir o transmitir y verificando el checksum o suma de comprobación.

En lo que al otro bloque se refiere, se podría haber empleado una macrofunción UART, pero se optó por un diseño propio mucho más sencillo con el objetivo de reducir la cantidad de recursos usados y optimizar el área que ocupa el dispositivo.

El esquema del módulo esclavo es similar que el del maestro. Está compuesto por un Controlador del esclavo y una unidad de transmisión-recepción asíncrono. Este diseño carece de líneas de control al ser el maestro el que gobierna el bus, ni tampoco existe la señal *br_clock*, ya que el ritmo de transmisión es obtenido a través del campo de sincronismo.

5. Resultados experimentales

Se realizó una simulación exhaustiva de los códigos en VHDL con el fin de depurar el mismo antes de la implementación de un prototipo en FPGA. La figura 5 muestra la operación del maestro. Cuando la señal *begin* se pone a uno, el maestro envía la cabecera del mensaje, con el campo de ruptura de sincronismo al principio, de 13 bits de duración, reduciendo el ritmo de transmisión. Las señales *enabrx* y *enabtx* muestran este comportamiento. A continuación se envía el campo de sincronismo y el de identificación. Por último se calcula el checksum y se transmite.

TABLA I
RECURSOS DEL MÓDULO MAESTRO

	Usados	Total	Porcentajes
Pines de entrada	4	6	66%
Pines de entrada/salida	5	183	2%
Celdas lógicas	380	1152	32%
Celdas embebidas	0	48	0%
EABs	0	6	0%

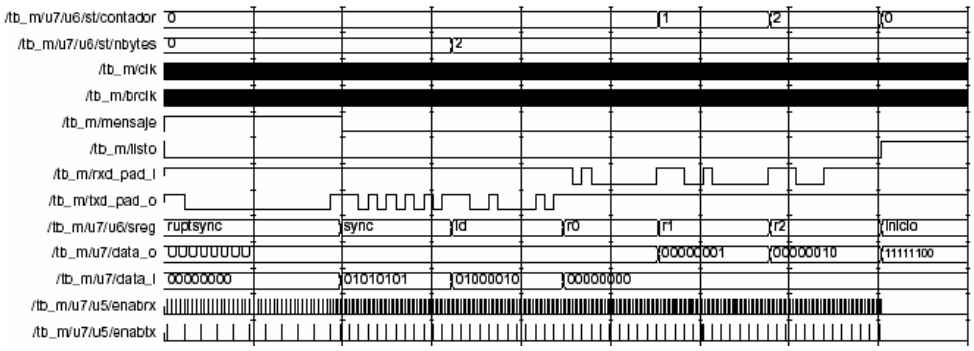

Figura 5: *Cronograma de funcionamiento*

Se desarrolló un prototipo implementado en una FPGA. La FPGA en cuestión fue un dispositivo lógico programable de la familia Flex10k de la casa Altera.. Los recursos que se emplearon para la implementación de un maestro y un esclavo figuran en la tablas I.

6. Conclusiones

Se diseñó una arquitectura distribuida para microinstrumentación de dos niveles jerárquicos. La metodología empleada en el desarrollo del bus de instrumentación consistió en la creación de un modelo IP usando VHDL. Los bloques VHDL aseguran portabilidad de los diseños, escalabilidad, configurabilidad e independencia de la tecnología. Además es fácilmente modificable para hacerlo más eficiente.

Esta arquitectura podría usarse en redes de sensores distribuidos para la medida de parámetros ambientales y de manera inalámbrica.

7. Referencias

[1] W. Lang, Reflexions on the future of Microsystems, Sensors and Actuators 72 (1999) 1-15.

[2] D. L. Polla, MEMS technoly and applications, VLSI Technology, Systems and Applications, 1999, International Symposium on, 8-10.

[3] S. Pitzed, W. Elmenreich, Configuration and Management of a Real-Time Smart Transducer Network, Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA'03. IEEE Conference, 16-19.

[4] C. Ferrer, B. Lorente, Smart sensors development based on a distributed bus for Microsystems applications, in: Proceedings of the SPIE's First International Symposium on Microtechnologies for the New Millennium 2003: Smart Sensors, Actuators and MEMS, Maspalomas, Gran Canaria, Spain, 19-21 May 2003.

[5] B. Lorente, J. Oliver, C. Ferrer, IBIS bus: towards a distributed architecture for MEMS integration, Sensors and Actuators 115 (2004) 470-475.

[6] H. Kopetz, G. Bauer, The Time-Triggered Architecture, Proceedings of the IEEE, volume 91, Jan. 2003, 112-126.

[7] LIN Specification Package Revision 1.2, http://www.lin-subbus.org

[8] H. Kopetz, M. Holzmann, W. Elmenreich, A Universal Smart Transducer Interface: TTP/A, Object-Orientd Real-Time Distributed Computing (ISORC2000). Proceeding. Third IEEE International Symposium on, 16-23.

[9] TTP/A (2000). TTP/A Specification of March 30, 2000, www.ttpforum.org

[10] W. Elmenreich, M. Delvai, Time-Triggered Communication with UARTs, Factory Communication Systems, 2002. 4th IEEE International Workshop on, Aug. 2002, 97-104.