

Implementación del protocolo DALI en FPGAs de bajo consumo de energía para uso en redes inalámbricas de sensores

Oscar Osvaldo Ordaz-García^{1,2,*}, Manuel Ortiz-López¹,
Francisco Javier Quiles-Latorre¹, José Guadalupe Arceo Olague²,
Francisco José Bellido-Outeiriño¹

¹ Universidad de Córdoba, Depto. Ingeniería Electrónica y de Computadores, España
oscardord27@hotmail.com, {ellorlom, ellqulaf, fjbellido}@uco.es

² Universidad Autónoma de Zacatecas, Unidad Académica Ingeniería Eléctrica, México
arceojg@uaz.edu.mx

Resumen. Este trabajo expone el diseño, descripción e implementación del hardware requerido por el protocolo *DALI* (*Digital Addressable Lighting Interface*) usado en la comunicación de nodos en redes inalámbricas de sensores para control de luminarias. La descripción se realiza en el lenguaje descriptivo de hardware (*VHDL*) para ser embebido sobre *FPGAs* (*Field Programmable Gate Array*), con el objetivo de que funcione sobre un nodo inalámbrico basado en el estándar *IEEE 802.15.4.*, con codificación *Manchester* diferencial *bi-phase*. El funcionamiento se comprobó con simulación y para contrastarlo se obtuvo una arquitectura de referencia implementada en cuatro plataformas de desarrollo con *FPGAs*, demostrando su correcta ejecución. La implementación del diseño en *FPGAs* de bajo consumo energético y económico, hace viable la disminución del consumo de energía generado en la comunicación, y realizar un prototipo de un nodo inalámbrico para su uso en una red inalámbrica de sensores.

Palabras clave: protocolo *DALI*, *FPGA*, redes inalámbricas de sensores, *VHDL*, codificación *Manchester* diferencial.

Implementation of the DALI Protocol in FPGAs of Low Energy Consumption for use in Wireless Sensor Networks

Abstract. This work exposes the design, description and implementation of the hardware required by the *DALI* (*Digital Addressable Lighting Interface*) protocol used in the communication of nodes in wireless sensor networks for luminaire control. The description is made in the hardware descriptive language (*VHDL*) to be embedded on *FPGAs* (*Field Programmable Gate Array*), with the aim of operating on a wireless node based on the *IEEE 802.15.4* standard, with *Manchester* differential *bi-phase* coding. The operation was verified with simulation and to contrast it, a reference architecture was obtained, implemented in four development platforms with *FPGAs*, demonstrating its correct execution. The implementation of the design in *FPGAs* of low energy and economic

consumption, makes viable the decrease of the energy consumption generated in the communication, and make a prototype of a wireless node for use in a wireless sensor network.

Keywords: DALI protocol, FPGA, wireless sensor networks, VHDL, Manchester differential coding.

1. Introducción

Los sistemas de comunicación para transmitir datos para la administración de la iluminación pública consumen gran cantidad de energía eléctrica. Además, estos sistemas dependen en gran medida de la regulación de la tensión y del mantenimiento [1], y en muchas ocasiones esto implica un mayor costo [2], en contraste con nuevos sistemas de ahorro de energía, como la iluminación basada en leds [3]. Una estrategia para reducir el consumo de energía es el protocolo de datos *DALI*, protocolo de comunicación desarrollado por empresas de equipamiento de iluminación, para asegurar interoperabilidad entre diferentes fabricantes, funcionalidad, simplicidad, ahorro de energía y bajos costos [4].

Aunado el protocolo *DALI*, con el uso de dispositivos lógicos programables para el control de la comunicación en sistemas de iluminación se puede reducir el alto consumo de energía que este genera. Un *FPGA* es un circuito integrado diseñado para que su interconexión y funcionalidad sea configurada por un lenguaje de descripción de hardware (*HDL*) [5]. *VHDL* es un lenguaje de descripción de hardware de *VHSIC*, utilizado para describir circuitos digitales y automatizar diseños electrónicos en el cual la programación se puede ejecutar en paralelo [6].

El objetivo de este trabajo es describir el protocolo de datos *DALI* en el lenguaje descriptivo de hardware *VHDL* para su funcionamiento en un nodo inalámbrico basado en el estándar *IEEE 802.15.4*, permitiendo realizar una arquitectura de referencia para implementarla en *FPGAs* de bajo consumo energético y económico.

En el documento se comentan los antecedentes y trabajos previos; posteriormente se explican los materiales y métodos del procesamiento y representación de la descripción de la interfaz; enseguida los resultados y finalmente las conclusiones.

2. Antecedentes y trabajos previos

En investigaciones referentes a redes inalámbricas de sensores, se menciona que el mayor consumo energético se debe a las comunicaciones; además, en muchas ocasiones esta energía se malgasta al mantener un nodo encendido a la espera de recibir algún mensaje (un módulo de comunicaciones típico de redes de sensores gasta la misma energía en modo emisor que en modo receptor) [7], por lo que Rosello *et al.* [7] propone una arquitectura de procesamiento para comunicaciones bajo demanda en *FPGA* de bajo consumo, evaluando la utilización de dispositivos de lógica programable para realizar el procesamiento de mensajes. Cárdenas *et al.* [8] metodológicamente describen a nivel cualitativo los protocolos de enrutamiento basados en la estructura de red jerárquica para la búsqueda de un equilibrio, donde los protocolos pretenden

mejorar los algoritmos de funcionamiento, procesamiento y transmisión de información. En [9] se diseña e implementa un servidor *Web* empotrado en un *FPGA* para proporcionar una interfaz remota y monitorizar una red inalámbrica de sensores.

Se han realizado trabajos para obtener mediciones de varios parámetros con dispositivos programables, como en [10] donde se identifica la necesidad de mejorar la sensibilidad, estabilidad y presentación de datos. De forma semejante, en [11] se utiliza un *FPGA* para resolver problemas de adquisición y presentación de datos; sin embargo, no los resuelven del todo, ya que se encuentran diferentes problemas de conversión, procesamiento y presentación. Se emplea en [12] un lenguaje de descripción de hardware de alto nivel para ser usado en placas de desarrollo de la familia *Zynq* de *Xilinx*, con el objetivo de diseñar e implementar un *MCA (Multi-Channel Analyzer)* de 4096 canales para espectrometría nuclear, obteniendo datos de un generador de funciones.

Hoy en día, antes de fabricar en silicio algún procesador, se realiza un riguroso análisis, diseño, implementación y pruebas en un dispositivo lógico reconfigurable; los *FPGAs* son utilizados para estos procesos [13]. Diferentes ejemplos se muestran en trabajos realizados en [14, 15, 16, 17]. Como se ha constatado, diversas investigaciones tratan de mejorar la obtención de datos sensibles a una magnitud y otras, su comunicación a través de dispositivos electrónicos, por lo que el uso de *FPGAs* para este fin y para el que se pretende en este trabajo es viable.

El desarrollo de redes inalámbricas de sensores implica el estudio de varios aspectos, como lo relacionado a plataformas de hardware que soporten sensores y comunicación, y lo referente al software. Este trabajo se enmarca en referencia al hardware, para lo cual se necesitan los materiales descritos en la siguiente sección.

3. Materiales y métodos

En el proyecto se describió en *VHDL*, el protocolo de datos *DALI* para el control de luminarias, con una estructura *master – slave*. La topología de bus está basada en un canal de comunicación serie de dos vías, donde hay al menos un controlador *master* y generalmente múltiples *slaves*. El *master* envía paquetes de datos de 16 bits y recibe respuestas con paquetes de datos de 8 bits en codificación *Manchester* diferencial *bi-phase*. Donde “01” corresponde a ‘1’ y “10” corresponde a ‘0’. Se usan dos símbolos por cada bit de información, el *bit rate* está especificado en 1200 *bps* con un tiempo de error de $\pm 10\%$, el tiempo de bit es 833,33 μs , y la frecuencia a la cual funciona el canal es de 2400 *Hz*. El bit más significativo (*MSB*) se envía primero, como se ilustra en la Fig. 1. La sincronización indica que el tiempo medio por bit (T_e) es de 416.67 μs , por lo que un paquete de envío tiene una duración de 38 T_e , que es igual a 15.83 *ms*. Un *Backward frame* toma 22 T_e o 9.17 *mseg* [18].

En terminología *DALI*, un *Forward Frame* es un paquete enviado del dispositivo de control al equipo receptor; y un *Backward Frame* es un paquete de respuesta enviado del equipo receptor al dispositivo de control [18]. El *Forward Frame* consta de un bit de inicio, ocho bits de dirección, ocho bits de datos y dos bits de alto, como se muestra en la Fig. 2, donde: *S* es el bit de inicio (1 lógico); *YAAA AAAS* es el *Byte* de dirección;

XXXX XXXX es el Byte de datos; y / / son dos bits de alto (línea inactiva) [19]. El *Backward Frame* consta de un bit de inicio, ocho bits de datos y dos bits de alto. Donde: S es el bit de inicio (1 lógico); XXXX XXXX es el Byte de datos; y / / son dos bits de alto (línea inactiva), como se observa en la Fig. 3 [18].

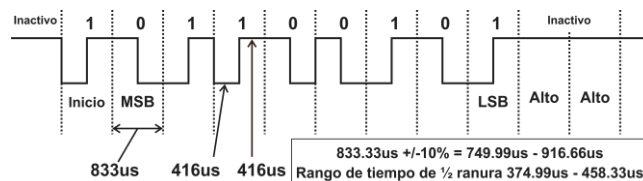


Fig. 1. Transmisión de DALI.

Como la transferencia de información es en codificación *Manchester* diferencial *bi-phase*, es necesario la decodificación para determinar la dirección y entonces procesar el mensaje. El código es un formato de codificación digital en el cual el símbolo ‘1’ está representado por una transición ascendente (estado en bajo seguido de un flanco a alto) y el símbolo ‘0’ está representado por un flanco descendente (estado en alto seguido de un flanco a bajo). Ambos pulsos, altos y bajos, tienen un periodo de tiempo igual a la mitad del período de bit.

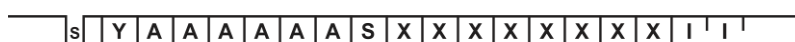


Fig. 2. Forward Frame.

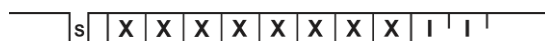


Fig. 3. Backward Frame.

Para implementar una arquitectura en un *FPGA* se diseña y describe su funcionamiento mediante el uso de diferentes herramientas, como diagramas esquemáticos, lenguajes descriptivos de hardware, o una combinación de los dos. Para describir el nodo de transmisión/recepción *DALI* se utilizó el lenguaje VHDL 93 estándar. El diseño, los bloques y su interconexión se muestran en la Fig. 4. La entidad principal nombrada *Rx_Tx_dali.vhdl* consta de dos bloques (*Transmisor_Manchester.vhdl* y *Receptor_Manchester.vhdl*). Depende de seis entradas (*Clk*, *Reset*, *WR_Manch*, *Dato_in_Manch(15:0)*, *RXD_Manch* y *Rx_ena_Manch*) y entrega cuatro salidas (*Ready_trans_Manch*, *TXD_Manch*, *Dato_rec_Manch* y *Dato_sal_Manch(7:0)*).

Con el objetivo de realizar el proceso de transmisión de datos, se desarrolló la entidad *Transmisor_Manchester.vhdl*, la misma consiste en una máquina de estados finitos (*FSM* por *finite state machine*) que se encarga de las fases necesarias para la transmisión, que es la sincronización de la frecuencia en *bps* del nodo de transmisión, la coordinación de la ejecución del protocolo con un codificador/decodificador *Manchester* y tiene un registro de datos paralelo/serie para el envío de bits.

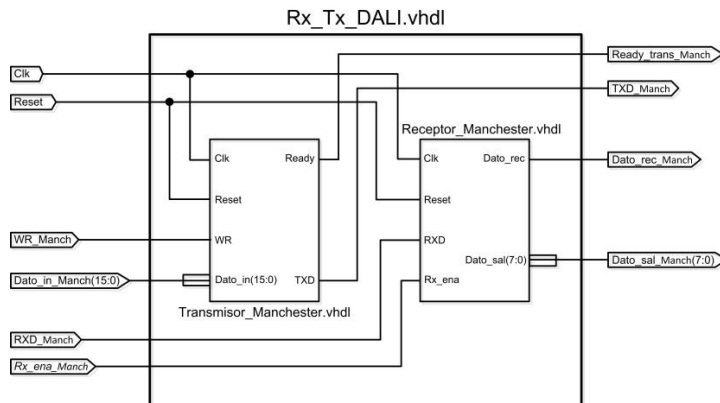


Fig. 4. Diagrama de la Descripción del Transmisor/Receptor para el protocolo DALI.

La entidad *Receptor_Manchestre.vhdl*, es descrita con una *FSM*; efectúa la recepción de datos por medio del puerto de entrada *RXD_Manch*, los 8 bit se reciben de forma secuencial e internamente con un registro se realiza la conversión de datos serie/paralelo para que el valor sea mostrado en el puerto de salida *Dato_sal_Manch(7:0)*, la terminal de salida *Dato_rec_Manch* indica el fin del proceso de conversión. La entidad, también sincroniza la velocidad para el dato de recepción que es de 1200 *bps*, realizando el muestreo del punto más preciso de *RXD_Manch* para obtener un valor válido del bit recibido, esto mediante la espera de un cuarto de bit una vez detectado el bit de “inicio” (*Rx_ena_Manch*) y después se espera un tiempo de bit de, que es el inverso de la velocidad de recepción y así detectar el valor del bit recibido de forma precisa, esto se realiza con un registro para determinar el instante de muestreo y un registro para contar el número de bit recibidos; además la entidad coordina la ejecución del protocolo con un codificador/decodificador *Manchester*.

4. Implementación del protocolo en FPGAs

Después de diseñar y describir la interfaz *DALI*, se realizó la simulación en el software de desarrollo *ISim* de *Xilinx ISE*. La simulación de la Fig. 5 comprueba que el diseño para la transmisión y la recepción de datos funciona adecuadamente. Para el proceso de transmisión, cuando la señal *wr* es activada (*wr* = 1) se cargan en un registro los datos que se desean enviar, en este ejemplo “1010101010101010”. En seguida se codifican los datos para enviar el bit de inicio del protocolo *Manchester*. Después, se envían uno a uno los bits, tal como se observa en la señal de salida *txd*. En la fase de recepción, es necesaria la activación de *rx_ena*, para comenzar a recibir la serie de bits, que se visualizan en la señal *rx_d*. Para la simulación, tanto el envío como la recepción son de 16 bits, dado que el puerto de salida *TXD* de la entidad *Transmisor_Manchester.vhdl*, es instanciado al puerto de entrada *RXD* de la entidad *Receptor_Manchestre.vhdl*, esto con la finalidad de que el dato de envío sea el mismo que se recibe, para comprobar que teóricamente existe sincronía en ambas entidades.

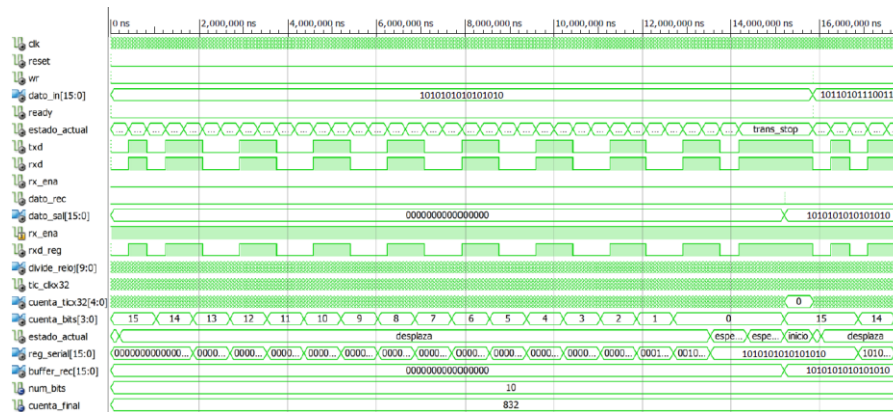


Fig. 5. Simulación de la Transmisión/Recepción de la Codificación Manchester Diferencial.

La descripción se implementó en cuatro diferentes placas de desarrollo, para comprobar su funcionamiento en cualquier *FPGA*, con reconfiguración respecto a la frecuencia del oscilador de cada placa de desarrollo. En la descripción se incluye el uso de “*generics*” para simplificar la reconfiguración de la implementación en las diferentes plataformas y adicionalmente hacer más compacto el diseño. Cabe mencionar que el propósito del presente trabajo es el desarrollo del hardware requerido para la transmisión/recepción de los datos, dejando de lado el aspecto de corrección de errores, el cual está a cargo de un bloque de corrección de errores.

Una de las plataformas de computación reconfigurable en la que se implementó la descripción fue la *UCOS3E5*, que se desarrolló por académicos-investigadores de la Universidad de Córdoba, España, con propósitos educativos y como plataforma de prototipado rápido en trabajos de investigación. La placa *UCOS3E5* contiene una *FPGA XC3S500E8 (PQ208)* de la familia *Spartan-3E* de *Xilinx* semiconductor [19], un oscilador a 32 MHz, y varios componentes más.



Fig. 6. Transmisión/Recepción del protocolo DALI implementado en la plataforma UCOS3E5.

Para la implementación de la descripción de la interfaz DALI en la plataforma *UCOS3E5*, los datos a transmitir se ingresaron con los *switch*'s. Como los bits a transmitir son 16 y la placa solamente cuenta con 8 *switch*'s, la descripción se reconfiguró para que al momento de hacer la carga al “*buffer*” de datos, los 16 bits correspondan al complemento de los *switch*'s concatenado con el dato original de los

switch's. Los datos ingresados fueron "11111110", por lo tanto, los datos que se transmitieron fueron: "000000011111110". En la Fig. 6 se puede observar la transmisión codificada en *Manchester* diferencial del dato mencionado. Se analizó la sincronización del envío y recepción de los datos, ya que como se indica de manera teórica, el tiempo medio por bit (T_e) debe ser de $416.67 \mu s$, y con la información obtenida por el osciloscopio, en la implementación es de $416 \mu s$.

Otra plataforma para implementar la descripción del interfaz *DALI*, fue la *Basys 2* de *Digilent*, que contiene una *FPGA XC3S100E (CP132)* de la familia *Spartan-3E* de *Xilinx* [20]. Incluye un oscilador de silicio configurable, a frecuencias de $25 MHz$, $50 MHz$ o $100 MHz$, y más componentes que permiten implementar diferentes diseños, entidades y sistemas digitales.

La descripción se reconfiguró para hacer la implementación para enviar los 16 bits. Se usaron los 8 *switch*'s para cargar los datos a enviar. Ahora la configuración de los datos se realizó con el dato de los *switch*'s concatenado con su complemento. El valor de los *switch*'s fue "10001001" por lo que la transmisión se generó para la siguiente cadena de bits "1000100101110110". En la Fig. 7 se corrobora el dato en codificación *Manchester* diferencial.

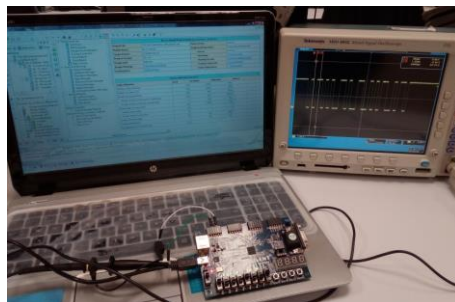


Fig. 7. Implementación del protocolo *DALI* en la plataforma *Basys 2* de *Digilent*.

El tiempo de bit en teoría debe ser de $833,33 \mu s$, y a través de una aproximación por medio de las líneas verticales de medición del osciloscopio en las transiciones de la codificación *Manchester*, se puede ver que el resultado obtenido es de $840 \mu s$, valor aceptable dentro del espacio de tiempo para error que es del $\pm 10 \%$. Todo esto se visualiza en la Fig. 8.

Al hacer una medición ajustada a los flancos, se puede observar en la Fig. 9 que el tiempo medio por bit (T_e) es de $418 \mu s$, y en comparación con el ideal, se determinó que solamente se tiene un 0.32% de error.

En las plataformas de desarrollo *iCEblink40-HX1K* [21] y *iCEblink40-LP1K* [22] *Evaluation Kit* de *Lattice Semiconductor* se realizaron dos implementaciones más. Ambas placas tienen un oscilador de precisión (*LT1799*) con tres diferentes frecuencias configurables de $3.33 MHz$, $333 KHz$ y $33.3 MHz$.

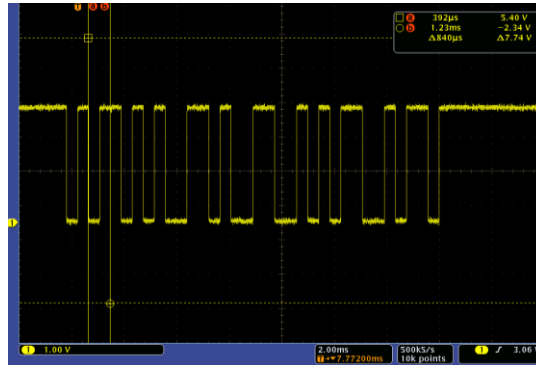


Fig. 8. Resultados de la transmisión del protocolo DALI en la plataforma Basys 2 de Digilent.

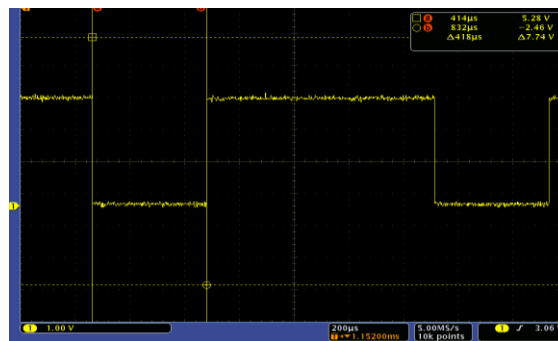


Fig. 9. Medición del tiempo medio por bit, del dato de recepción (Basys 2 de Digilent).

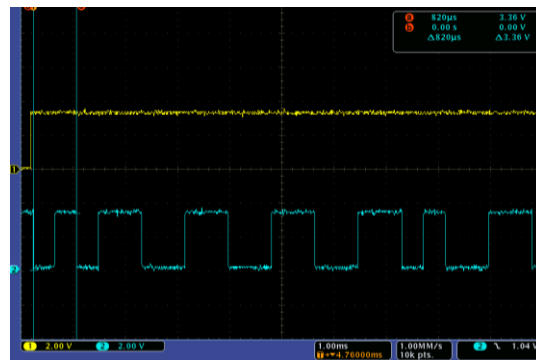


Fig. 10. Secuencia del protocolo DALI implementado en la plataforma iCEblink40-HX1K.

Para las implementaciones en las placas de *Lattice Semiconductor*, se utilizó un registro para el envío de los datos con un valor constante. En la Fig. 10 se muestra un fragmento de los datos enviados en codificación *Manchester* diferencial. Después de la transición del bit de inicio, se pueden observar transiciones referentes al dato con valor de “10101010010”. Lo relativo a la sincronización se puede observar en un

acercamiento aproximado con las líneas de medición verticales del osciloscopio un tiempo por bit de $820 \mu s$ en la transición inicial. Este valor, se encuentra por mucho dentro del porcentaje de error permitido.

5. Conclusiones

Se obtuvo la descripción en el lenguaje descriptivo de hardware *VHDL* de la interfaz *DALI* para transmisión y recepción de paquetes de datos en codificación *Manchester* diferencial; la cual puede funcionar sobre un nodo inalámbrico basado en el estándar *IEEE 802.15.4*. El objetivo principal se cumplió, ya que se realizó una arquitectura de referencia reconfigurable que se implementó en varias plataformas de desarrollo, en las cuales como dispositivo principal es un *FPGA*. Los resultados de la implementación muestran que el diseño y descripción cumplen con los requerimientos necesarios para la transmisión/recepción.

En particular, la implementación de la descripción en las plataformas de desarrollo de *Lattice Semiconductor* que tienen *FPGAs* de la familia *iCE40* que son de ultra bajo consumo energético, fabricadas con tecnología *CMOS* estándar de baja potencia de 40 nm y con altas prestaciones de desempeño, permite reducir el consumo de energía que generan los procesos de comunicación inalámbrica en los dispositivos que actualmente soportan el protocolo *DALI*, y entonces realizar el prototipo de un nodo inalámbrico o mote para su uso en una red inalámbrica de sensores asociado al bajo costo económico de éstas permitiría aumentar la cobertura de la red.

La cantidad de casos de prueba realizados permitió perfeccionar a detalle el diseño y la descripción para realizar las correcciones pertinentes. La variación de los valores obtenidos para la sincronización en el tiempo por bit para la transmisión/recepción con relación a los valores ideales es mínima y en algunos casos se puede considerar despreciable. El usar *VHDL 93* estándar permitió sintetizar la descripción en los *FPGAs* sin la necesidad de realizar cambios en la descripción (código) y, en consecuencia, garantizar la implementación en *FPGAs* de bajo consumo de energía.

Como trabajo futuro se pretende realizar una etapa donde el dato de transmisión pueda ser administrado por medio de un dispositivo de comunicación universal que controla puertos y dispositivos, y con el mismo conocer el dato de recepción.

6. Recomendaciones

Para continuar con el desarrollo del nodo inalámbrico, es necesario abordar temas de optimización de recursos de hardware, con el objetivo de disminuir la cantidad de hardware utilizado y con ello el consumo de energía eléctrica que tienen los nodos. Como se ha plasmado, también es necesario un sistema de administración para la arquitectura de referencia implementada en este trabajo, el cual puede ser a través de dispositivo de comunicación universal controlado por un programa para conexión de dispositivos, por lo que es necesario conocimientos en temas de comunicación y luminarias. Un punto clave para realizar el prototipo del nodo es la codificación del conjunto de instrucciones del protocolo *DALI* para el control de luminarias.

Referencias

1. Bellido-Outeiriño, F.J., Quiles-Latorre, F.J., Moreno-Moreno, C.D., Flores-Arias, J.M., Moreno-García, I., Ortiz-López, M.: Streetlight Control System Based on Wireless Communication over DALI Protocol. *Sensors* 16, 597 (2016)
2. Quintero, J. R., Prieto, L. F., *Sistemas inteligentes de transporte y nuevas tecnologías en el control y administración*, pp. 53–62 (2015)
3. Domínguez, L. I., Ordaz, O. O., Arceo, J. G., Hernández, V. M., Solís, R.: Prototipo de un sistema para el control de tráfico vehicular y peatonal implementado en un FPGA. *IEEE Sec Morelos. XIV CIINDET 2018*, pp. 130–134 (2018)
4. DALI Standard. International Electrotechnical Commission IEC 62386.
5. Wisniewski, R.: Synthesis of compositional microprogram control units for programmable devices. *Zielona Góra: University of Zielona Góra*. p. 153 (2009)
6. Department of Defense: Military Standard, Standard general requirements for electronic equipment. (1992)
7. Rosello, V., Portilla, J., Riesgo, T.: *Arquitectura de Radios Wake-up para redes de sensores inalámbricas basada en FPGA. SAAEI 2013*, Madrid, Spain (2013)
8. Cárdenas, J.: Protocolos de Enrutamiento basados en la Estructura de Red Jerárquica hacia la Eficiencia Energética en Redes de Sensores Inalámbricos. *Ciencia, Innovación y Tecnología* 20162 (1), pp 37–54
9. Rodríguez Valido, M., Gutiérrez Castañeda, M., Cardell Bilbao, A., Ayala Alfonso, A., Díaz Gopar, J. J., Sobota Rodríguez, C., Magdalena Castelló, E.: Web server empotrado en FPGA para monitorización de una red de sensores inalámbricos. En: *IX Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica* (2010)
10. Solís, R., et al.: Beneficios y Desafíos en el uso de una Red Inalámbrica de Sensores para el Monitoreo de una Red de Distribución de Agua Potable. En: *Memorias del CONCYE 2011*, pp. 21–23 (2011)
11. García Duran, A., Ordaz García, O., Arceo Olague, J. G., Hernández, C.: Adquisición de datos de sensores de una red de distribución de agua potable. *C. Internacional de I Academia Journals*, V. 7, No. 3, pp. 255–259 (2015)
12. García Durán, A., Hernández Dávila, V. M., Vega Carrillo, H. R., Ordaz García, O. O., Bravo Muñoz, I.: Analizador Multicanal embebido en FPGA. En: *XVII International Symposium on Solid State Dosimetry ISSSD 2017 Dominican Republic* (2017)
13. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Desarrollo del CORE de un Procesador de Imágenes de tipo SIMD embebido en un FPGA. *IEEE Internacional Sección Centro Occidente ROPEC*, pp. 463–468 (2012)
14. Ordaz, O. O., Rico Sabag, A. A., Arceo, J. G., González Carrillo, L. J.: Implementación de un Procesador Elemental en un FPGA. En: *ENINVIE 2010, Zacatecas, Zac.*, pp. 76–83 (2010)
15. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Eficiencia del uso de recursos en un FPGA para la descripción de un Procesador Elemental. En: *XII Reunión Internacional de Otoño de Potencia, Electrónica y Computación 10* (2010)
16. Nava, D., Ordaz, O., Hernández, M.: Desarrollo de ensamblador para procesador de imágenes tipo SIMD. En: *C. Internacional de I. - Academia Journals*, V. 5, No. 3, T. 16, pp. 2364–2369 (2013)
17. Ordaz, O. O., Hernández, M., Benavides, J. I., Arceo, J. G.: Diseño de la Unidad Elemental de un Microprocesador en un FPGA. *CONCYE 2011*, pp 57–63. Zacatecas, México (2011)
18. Husain, S.: Digitally Addressable Lighting Interface (DALI) Communication. AN1465. Microchip Technology Inc. (2012)
19. Xilinx: Spartan-3 FPGA Data Sheet.

20. Digilent: BASYS 2 Reference Manual (2010)
21. User's Guide iCEblink40-HX1K Evaluation Kit Lattice Semiconductor.
22. User's Guide iCEblink40-LP1K Evaluation Kit Lattice Semiconductor.