



UNIVERSIDAD DE CÓRDOBA

PROGRAMA DE DOCTORADO EN COMPUTACIÓN AVANZADA,
ENERGÍA Y PLASMAS

TESIS DOCTORAL

**Viabilidad del uso de FPGAs en redes
de sensores inalámbricos**

**Viability of FPGAs in wireless sensor
networks**

MEMORIA PRESENTADA PARA OPTAR AL GRADO DE DOCTOR POR:

OSCAR OSVALDO ORDAZ GARCÍA

DIRECTORES:

DR. D. FRANCISCO JOSÉ BELLIDO OUTEIRIÑO

DR. D. MANUEL AGUSTÍN ORTIZ LÓPEZ

DR. D. JOSÉ GUADALUPE ARCEO OLAGUE

CÓRDOBA, JUNIO DEL 2022

TITULO: *Viabilidad del uso de FPGAs en redes de sensores inalámbricos*

AUTOR: *Oscar Osvaldo Ordaz García*

© Edita: UCOPress. 2022
Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

[https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es](https://www.uco.es/ucopress/index.php/es/ucopress@uco.es)



TÍTULO DE LA TESIS: Viabilidad del uso de FPGAs en redes de sensores inalámbricos

DOCTORANDO: Oscar Osvaldo Ordaz García

INFORME RAZONADO DE LOS DIRECTORES DE LA TESIS

Esta Tesis Doctoral se centra en el diseño y desarrollo de un *bridge* para protocolo *Digital Addressable Lighting Interface* (DALI), utilizado en sistemas inteligentes de alumbrado público, que será implementado en una FPGA. El *bridge* se ha descrito en VHDL de manera que se ha conseguido un diseño portable y que optimiza los recursos. Esto ha permitido su implementación en una FPGA de bajo consumo de potencia y bajo coste económico, condiciones necesarias para ser utilizada en un nodo de una red de sensores inalámbricos. El *bridge* se ha probado en un nodo inalámbrico para demostrar que es posible utilizar FPGAs en redes inalámbricas y las ventajas que supone una implementación en hardware del protocolo DALI.

Durante el desarrollo de la Tesis Doctoral, el doctorando ha demostrado una gran capacidad de organización y trabajo autónomo adaptándose y buscando soluciones para resolver los diferentes problemas que han ido apareciendo en el transcurso de la misma. Por tanto, se pone de manifiesto que el doctorando ha adquirido las competencias necesarias para el desarrollo de una prometedora carrera científica. Por todo lo reflejado en este informe, los directores consideramos que esta Tesis Doctoral posee una gran calidad y prueba de ello es que los resultados derivados de ella han sido publicados en una revista indexada (JCR 2020: Q3, T2), Journal Citation Indicator (JCI 2020, Q2, T2).

Ordaz-García Oscar Osvaldo, Ortiz-López Manuel, Quiles-Latorre Francisco Javier, Arceo-Olague José Guadalupe, Solís-Robles Roberto and Bellido-Outeiriño Francisco José. DALI Bridge FPGA-Based Implementation in a Wireless Sensor Node for IoT Street Lighting Applications. Electronics, Volume 9, Issue 11 (November 2020) 1803; <https://doi.org/10.3390/electronics9111803> JCR category rank: 145/273 in Engineering, Electrical & Electronic ISSN: 2079-9292.

Por todo ello, se autoriza la presentación de la Tesis Doctoral.

Córdoba, España, 2 de junio de 2022

Firma de los directores

Fdo.: Dr. Francisco José Bellido Outeiriño

Fdo.: Dr. Manuel Agustín Ortiz López

Fdo.: Dr. José Guadalupe Arceo Olague

*A Francisco Javier Quiles Latorre y a Manuel Agustín Ortiz López
Esplendidos y magníficos seres humanos, maestros y amigos.
Grandes guías en este destino.*

Agradecimientos

Después de invertir tanto tiempo para poder concluir este trabajo, quiero agradecer a mis directores de tesis, Manuel Agustín Ortiz López, José Guadalupe Arceo Olague y Francisco José Bellido Outeiriño, por ser forjadores de este proyecto, por su guía académica a lo largo del proceso de investigación, y ofrecer sus consejos para realizar esta memoria de tesis. Para mí es un honor trabajar con ellos.

Quiero manifestar mi agradecimiento de manera infinita a Francisco Javier Quiles Latorre por su paciencia, por compartirme todo su sabiduría y conocimiento, además de ser un orientador que ha dejado una gran huella en mi vida.

Agradezco el aliento para seguir adelante de mi esposa Claudia Olimpia. A mis hijos Oscar Osvaldo, Bruno Octavio y Christian Emilio por su apoyo. A mi padre Roberto y a mi madre Amparo (f) por la educación que me brindaron y enseñarme a tratar de superarme día con día.

También quiero agradecer a la Universidad de Córdoba y a la Universidad Autónoma de Zacatecas, instituciones educativas que me han otorgado las facilidades para realizar los estudios de doctorado.

Por último, reconozco a todas las personas que de muchas formas me alentaron a iniciar, continuar y concluir este proyecto de vida, denominado doctorado.

Gracias.

Resumen

Dentro del contexto del proyecto de tesis planteado, se aborda el problema del control y las comunicaciones en las luminarias de los sistemas inteligentes de alumbrado público. En los trabajos reportados en la literatura sobre el tema planteado, se encuentra el problema particular de la implementación de protocolos digitales de comunicaciones en las luminarias, y especialmente en la implementación del protocolo *Digital Addressable Lighting Interface* (DALI) en redes de sensores inalámbricos. Desafortunadamente los microcontroladores que habitualmente se emplean en los nodos de sensores inalámbricos, no integran periféricos para este tipo de protocolos, por lo que suelen implementarse por software. Un punto de oportunidad es implementar este tipo de protocolos en hardware mediante una FPGA y de esta forma demostrar su viabilidad en nodos de sensores inalámbricos.

En esta tesis se presenta una solución para implementar por hardware el protocolo DALI para controlar sistemas de iluminación inteligente. La novedad de este trabajo es la descripción portable del protocolo DALI implementado en forma de un *bridge* de comunicaciones en una FPGA de bajo coste, bajo consumo de energía y poca cantidad de recursos lógicos, para ser embebida en un nodo sensor inalámbrico. El protocolo se ha descrito en lenguaje VHDL siguiendo los estándares 1076-93 y 1076.3-97. Las contribuciones de este trabajo demuestran la viabilidad de la utilización de FPGAs en redes de sensores inalámbricos, que en este caso se utilizan para implementar el *bridge* DALI y resolver el problema de la interfaz de comunicación.

Abstract

Within the context of the proposed thesis project, the problem of control and communications for luminaires in intelligent public lighting systems is addressed. In the works reported in the literature on the subject, we can find the particular problem of the implementation of digital communication protocols for luminaires, and specifically the implementation of the Digital Addressable Lighting Interface (DALI) protocol in wireless sensor networks. Unfortunately, the microcontrollers that are commonly used in wireless sensor nodes do not integrate peripherals for this type of protocol, so they are usually implemented by software. A point of opportunity is to implement this type of protocols in hardware through an FPGA and thus demonstrate its viability in wireless sensor nodes.

In this thesis, a solution is presented to implement the DALI protocol by hardware to control intelligent lighting systems. The novelty of this work is the portable description of the DALI protocol implemented in the form of a communications bridge in a low-cost FPGA, with low power consumption and few logical resources, which will be embedded in a wireless sensor node. The protocol has been described in VHDL language following the 1076-93 and 1076.3-97 standards. The contributions of this work demonstrate the feasibility of using FPGAs in wireless sensor networks, which in this case are used to implement the DALI bridge and solve the communication interface problem.

Índice

Resumen.....	i
Abstract.....	i
Índice.....	iii
Índice de Figuras.....	v
Índice de Tablas	vii
Acrónimos empleados.....	ix
Capítulo 1. Introducción	1
1.1. Internet de las cosas e iluminación	1
1.2. Matrices de puertas programables en campo	4
1.3. Motivación y objetivos del trabajo.....	6
1.4. Estructura de la tesis	8
Capítulo 2. Conceptos generales de las WSNs y de las FPGAs.....	11
2.1. Redes inalámbricas de sensores	11
2.1.1. Evolución de las WSNs	12
2.1.2. Concepto del IoT.....	14
2.1.3. Arquitectura de las redes inalámbricas de sensores	16
2.1.4. Topología	18
2.1.5. Nodo inalámbrico.....	21
2.2. Iluminación inteligente	23
2.2.1. Protocolos utilizados en el control de iluminación	25
2.2.2. Protocolo DALI	26
2.3. FPGAs.....	27
2.3.1. Evolución de las FPGAs	29
2.3.2. Descripción general de la arquitectura de las FPGAs	34
2.3.3. Tipos de tecnologías utilizadas	35
2.3.4. Comparación de FPGAs y otros dispositivos	36
2.3.5. <i>Glue logic</i> e interfaz personalizada	39
2.4. Máquina de estados finita	40
2.5. Protocolo de comunicación en serie.....	43
Capítulo 3. Trabajos relacionados	47
3.1. Trabajos que incluyen nodos inalámbricos para sistemas de alumbrado público	47
3.2. Trabajos relacionados con sistemas de iluminación inteligente que utilizan DALI	50
3.3. Trabajos que describen nodos inalámbricos que contienen FPGAs	52
Capítulo 4. Arquitectura del <i>bridge</i> DALI.....	57

4.1.	Protocolo DALI.....	57
4.2.	Diseño y descripción del <i>bridge</i> DALI	59
4.2.1.	Comunicación entre el <i>host</i> y el <i>bridge</i> DALI	61
4.2.2.	UART	64
4.2.3.	Procesador DALI.....	72
4.2.4.	Transmisor / Receptor Código Manchester	75
Capítulo 5.	Resultados	85
5.1.	Descripción del nodo sensor inalámbrico UCODALIB	85
5.1.1.	FPGA.....	87
5.1.2.	Memoria flash EPROM de configuración	89
5.1.3.	Oscilador	89
5.1.4.	Módulo SoC de radiofrecuencia	91
5.1.5.	Interfaz DALI	92
5.1.6.	Fuente de alimentación de energía eléctrica.....	94
5.1.7.	Identificador del nodo, interfaz con el usuario y elementos auxiliares	97
5.2.	Resultados de la implementación del <i>bridge</i> DALI en el nodo sensor inalámbrico.....	97
5.2.1.	Recursos lógicos utilizados	98
5.2.2.	Latencia	99
5.2.3.	Frecuencia de operación	100
5.2.4.	Estimación de consumo de potencia.....	101
5.2.5.	Tiempo de bit.....	102
5.3.	Resultados de las pruebas electrónicas del nodo sensor inalámbrico.....	105
Capítulo 6.	Conclusiones.....	109
Referencias		115

Índice de Figuras

Figura 2.1 Arquitectura básica de las WSNs	16
Figura 2.2 Entorno local de una WSN	17
Figura 2.3 Fases iniciales de implementación de una WSN	18
Figura 2.4 Topologías de las WSNs	19
Figura 2.5 Arquitectura general del HW de un nodo sensor inalámbrico.....	22
Figura 2.6 Componentes del SSL.	24
Figura 2.7 Organización simplificada de las FPGAs.	28
Figura 2.8 Arquitectura general de las FPGAs	34
Figura 2.9 Elemento Lógico (LE) básico configurable.....	35
Figura 2.10 Ejemplo de un diagrama de estados de una FSM	41
Figura 2.11 Componentes de una FSM.....	42
Figura 2.12 Diagrama de un ASM.	43
Figura 2.13 Protocolo de la comunicación en serie.	44
Figura 2.14 Formato de la transmisión asíncrona.	45
Figura 4.1 Cambio de estado de la codificación Manchester.....	58
Figura 4.2 Formato de la trama DALI.	58
Figura 4.3 Transmisión DALI.....	58
Figura 4.4 Diagrama de bloques de la conexión del <i>bridge</i> DALI.	60
Figura 4.5 Diagrama de bloques del <i>bridge</i> DALI.	61
Figura 4.6. Formato de la trama de peticiones del <i>host</i>	62
Figura 4.7. Formato de la trama de respuesta al <i>host</i>	62
Figura 4.8 Representación a nivel de bloque de la UART.....	65
Figura 4.9 Diagrama de bloques de la UART.....	66
Figura 4.10 Diagrama de bloque del transmisor de la UART.....	67
Figura 4.11 Ruta de datos de la FSM del transmisor de la UART.....	67
Figura 4.12 Unidad de control de la FSM del transmisor de la UART.....	68
Figura 4.13 Diagrama de bloque del receptor de la UART.	69
Figura 4.14 Ruta de datos de la FSM del receptor de la UART.	70
Figura 4.15 Unidad de control de la FSM del receptor de la UART.	71
Figura 4.16 Representación a nivel de bloque del registro de estado.	71
Figura 4.17 Representación a nivel de bloque del procesador DALI.	73
Figura 4.18 Diagrama de estados de la FSM del procesador DALI.	74
Figura 4.19. Representación a nivel de bloque del TRCM.	76
Figura 4.20. Diagrama de bloques de la arquitectura del TRCM.	77
Figura 4.21 Diagrama de bloque del transmisor del TRCM.	78
Figura 4.22 Ruta de datos de la FSM del transmisor del TRCM.	78
Figura 4.23 Unidad de control de la FSM del transmisor del TRCM.	80
Figura 4.24 Diagrama de bloque del receptor de la TRCM.	81
Figura 4.25 Muestreo de los datos de recepción en codificación Manchester.	81
Figura 4.26 Ruta de datos de la FSM del receptor del TRCM.	82
Figura 4.27 Unidad de control de la FSM del receptor del TRCM.	83
Figura 5.1 Diagrama de bloques del nodo sensor inalámbrico UCODALIB.	86
Figura 5.2 Fotografía del nodo sensor inalámbrico UCODALIB.	86
Figura 5.3 Esquema eléctrico de la configuración del oscilador de frecuencia.	90
Figura 5.4 Módulo RF.....	91

Figura 5.5 Niveles lógicos DALI.	93
Figura 5.6 Esquema eléctrico de la interfaz DALI.	94
Figura 5.7 Esquema eléctrico para fuente de voltaje de 3,3 V.	95
Figura 5.8 Esquema eléctrico para generar el voltaje auxiliar de la FPGA.	95
Figura 5.9 Esquema eléctrico para generar el voltaje de los elementos lógicos de la FPGA.	96
Figura 5.10 Esquema eléctrico para fuente de voltaje de 15 V.	96
Figura 5.11 Resultados de la síntesis en términos de la utilización de recursos lógicos.	98
Figura 5.12 Resultados de recursos ubicados para el enrutamiento.	99
Figura 5.13 Tiempo de latencia real del <i>bridge</i> DALI.	99
Figura 5.14 Tiempo de respuesta del <i>bridge</i> DALI incluyendo el tiempo de comunicación.	100
Figura 5.15 Resultado de frecuencia de reloj.	101
Figura 5.16 Medición del tiempo medio por bit (T_e).	103
Figura 5.17 Resultado de una trama “Forward” ejecutada por el <i>bridge</i> DALI.	104
Figura 5.18 Decodificación del TRCM de un comando DALI.	105
Figura 5.19 Pruebas de tensión en la salida de la interfaz DALI.	106
Figura 5.20 Resultado de las pruebas de estabilidad y precisión del oscilador.	107
Figura 5.21 Prueba de precisión y estabilidad del voltaje de 3,3 V.	108

Índice de Tablas

Tabla 2.1 Evolución de las WSNs	14
Tabla 2.2 Diferentes tipos de FPGAs.	33
Tabla 2.3 Diferentes tipos de tecnología de configuración.....	36
Tabla 2.4 Comparación entre FPGAs y CPUs.....	37
Tabla 2.5 Diferencias entre FPGAs y ASICs.....	38
Tabla 4.1 Comandos generales del <i>bridge</i> DALI y propios de la interfaz DALI.	63
Tabla 4.2 Codificación de la respuesta DALI.....	64
Tabla 5.1 Información de la FPGA iCE40 -VQ100 de Lattice Semiconductor.....	88
Tabla 5.2 Estimación de potencia.	102
Tabla 5.3 Comparación de voltajes.....	107

Acrónimos empleados

ADC	Analog-to-Digital Converter
ASIC	Application-Specific Integrated Circuit
ASM	Algorithmic State Machine
BRAM	Block Random Access Memory
CI	Circuito Integrado
CLB	Configurable Logic Block
CLK	Clock
CPU	Processor o Central Processing Unit
DAC	Digital-to-Analog Converter
DALI	Digital Addressable Lighting Interface
DSP	Digital Signal Processor
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GPIO	General Purpose Input/Output
HDL	Hardware Description Language
HPC	High Performance Computer
HW	Hardware
IOBs	Input-Output Blocks
IoT	Internet of Things
IP Core	Intellectual Property Core
LC	Logic Cell
LE	Logic Element
LUT	Look-Up Table
MCU	Microcontroller
MP	Microprocessor
PC	Personal Computer
PCB	Printed Circuit Board
RAM	Random Access Memory
SFM	Serial Flash Memory
SoC	System-on-a-Chip
SSL	Smart Street Lighting
SW	Software
TRCM	Transmisor / Receptor Código Manchester
UART	Universal Asynchronous Receiver-Transmitter
VHDL	VHSIC Hardware Description Language
VHSIC	Very High-Speed Integrated Circuit
WSN	Wireless Sensor Network

Capítulo 1. Introducción

1.1. Internet de las cosas e iluminación

El Internet de las Cosas (Internet Of Things (IoT), en inglés) requiere protocolos estandarizados para integrar dispositivos heterogéneos de forma que se puedan interpretar y diseminar los datos que se transmiten de un extremo a otro, permitiendo la interconexión de dispositivos inteligentes. Se estimó que en 2010 la cantidad de dispositivos conectados a Internet, superó por más de seis veces la cantidad de personas conectadas. Desde entonces, el número de dispositivos móviles y de IoT continúa creciendo a un ritmo acelerado [1]. Aprovechando el paradigma IoT y la conectividad que permite, aparecen continuamente nuevas aplicaciones para el servicio público, la industria, las finanzas o el usuario final.

Los dispositivos IoT contienen sensores para la adquisición de datos y actuadores para realizar tareas de control, pudiendo intercambiar información entre ellos formando Redes Inalámbricas de Sensores (Wireless Sensor Networks (WSNs), en inglés), o enviar información a la nube para su posterior procesamiento. IoT, es soportado por millones de WSNs locales interconectadas. Dentro de cada WSN hay diferentes tipos de dispositivos inteligentes, los cuales se comunican a través de un conjunto de normas y reglas definidas en los diferentes protocolos de red. El nodo sensor inalámbrico o dispositivo inalámbrico inteligente, debe ser capaz de realizar procesamiento, además de transmitir y recibir información de diferentes nodos sensores inalámbricos o acceder directamente a Internet [2].

Las aplicaciones de las WSNs están determinadas por las capacidades de sensado y de comunicación de los nodos sensores inalámbricos [3]. El desarrollo de las WSNs se inspiró en aplicaciones militares, en particular la vigilancia en zonas de conflicto. Las WSNs se utilizan en diferentes campos, como el monitoreo de procesos industriales, monitoreo de la calidad del agua, monitoreo del estado de salud o sensores corporales, monitoreo del hábitat

de las personas y los animales, etc. [4,5]. También se utilizan para para monitorear las condiciones físicas y la automatización. Además, existen redes de sensores con el objetivo de monitorear y controlar el tráfico de vehículos [6].

En el proceso de comunicación de las WSNs existen protocolos estandarizados para el intercambio de información entre los dispositivos heterogéneos que se conectan a la red, de forma que, se puedan interpretar y disseminar los datos que se transmiten de un extremo a otro. La evolución de las WSNs conducirá en los próximos años a cambios importantes en la vida cotidiana de las personas. De hecho, nuevas aplicaciones aparecen diariamente, aprovechando la conectividad de dispositivos inteligentes, proporcionando así nuevas características y servicios para el servicio público, la industria, las finanzas o el usuario final [2].

En el estado del arte de las WSNs se observan distintas vertientes que dependen de la aplicación. En particular, para el grupo de investigación en el que se está desarrollando este trabajo, la utilización de redes inalámbricas aplicadas al alumbrado público resultan de un interés especial. Las redes de comunicación y los nodos sensores juegan un papel vital en la transferencia de datos entre los sistemas de Alumbrado Público Inteligente (Smart Street Lighting (SSL), en inglés). Ofrecen un medio compartido en el que los sensores y actuadores pueden transferir sus datos [7]. El SSL se define como un sistema de alumbrado público basado en red que está equipado con sensores y actuadores, ofreciendo así un amplio conjunto de capacidades e interfaces de conectividad [8], infraestructuras de comunicación avanzadas y multipropósito, con el objetivo principal de iluminar carreteras, autopistas, áreas de estacionamiento y espacios públicos [9]. Con el aumento de las opciones de conectividad digital, los sistemas de SSL tienen diferentes interfaces inalámbricas, orientadas a aumentar la conectividad en los sistemas de redes inteligentes [10]. Estos sistemas permiten controlar y monitorear equipos eléctricos de iluminación modernos y heterogéneos [8].

En el contexto del IoT y SSL, los nodos deben soportar numerosas conexiones de sensores con diferentes interfaces. Además, se les exige una elevada capacidad de procesamiento sobre los datos adquiridos por los sensores e incluso aplicar técnicas de Inteligencia Artificial y Aprendizaje Automático (Machine Learning, en inglés) sobre ellos.

Cada vez más, en los sistemas de SSL se está utilizando la red de datos del alumbrado público como una red de *backbone* para ofrecer otros servicios y funciones, como: control de flujo de personas [6,11], control ambiental [12], búsqueda de aparcamiento, monitoreo remoto [13]. También, ofrecen servicios propios de los sistemas de SSL como son:

mantenimiento predictivo [14], control en tiempo real [11], mejora de la visibilidad [12] y la capacidad de control del sistema [7], que obliga a que los nodos realicen tareas propias de la red como encaminar los paquetes [15] y realizar control de flujo [12]. El alumbrado público en red debe ofrecer una base para aplicaciones de ciudades inteligentes mediante la introducción de IoT de vanguardia, para la planificación urbana [7], la gestión del tráfico [6,11,16,17], la identificación de patrones de movilidad [6,16,18], vigilancia [19], seguridad [20] y la asistencia de emergencia [12], por nombrar algunos. Actualmente, un punto importante de investigación es determinar la movilidad de las personas y la distancia entre las mismas.

En consecuencia, la infraestructura de las WSNs y el acceso rápido al conocimiento se vuelve crítico en aplicaciones, como el control de iluminación, la conducción autónoma, el mantenimiento predictivo y la monitorización de anomalías, donde el tiempo de respuesta del sistema es fundamental para garantizar la calidad del servicio [21].

Para cubrir las aplicaciones mencionadas, los nodos que forman las redes utilizadas en los sistemas de SSL deben aumentar cada vez más el rendimiento y el número de conexiones a sensores, además de mantener un bajo consumo, coste y facilidad de sustitución y mantenimiento, pero sin olvidar el control robusto y fiable que deben realizar sobre la iluminación de las lámparas. Una de las interfaces principales creadas para los sistemas de iluminación es la *Digital Addressable Lighting Interface* (DALI) [8], desarrollada especialmente para controlar digitalmente, luminarias, balastos electrónicos y fluorescentes regulables. En Europa se utiliza para solucionar problemas de energía, principalmente para fines comerciales e industriales. Es parte de la especificación IEC 60929 y se identifica en el Anexo E del estándar EN60929 europeo de balasto electrónico [22]. Los balastos tipo DALI pueden funcionar con niveles de potencia más bajos que los balastos magnéticos estándar [23-26]. El protocolo DALI ofrece comunicación bidireccional, por lo que garantiza una transmisión más fiable y segura de los datos, en comparación con otros protocolos de control de iluminación [27]. Algunas de las principales ventajas de utilizar el protocolo DALI son:

- Monitorización de los fallos eléctricos o los fallos en las lámparas
- Notificación del estado de la lámpara
- Regulación del nivel de potencia de la lámpara
- Optimización del consumo de potencia

- Permite enviar información sobre las características eléctricas de la lámpara (voltaje, corriente y potencia).

1.2. Matrices de puertas programables en campo

La utilización de las Matrices de Puertas Programables en Campo (Field Programmable Gate Array (FPGAs), en inglés), se ha generalizado en todas las áreas donde se implementan sistemas digitales, por lo que existe una gran diversidad de sectores de uso como son: aeroespacial, automotriz, difusión, electrónica de consumo, defensa militar, computación de alto rendimiento, aplicaciones industriales, aplicaciones médicas y comunicaciones alámbricas e inalámbricas [28]. Se utilizan específicamente en nuevas aplicaciones como:

- Creación de prototipos para ASIC.
- Comunicación por cable y comunicación inalámbrica: Desarrollo de sistemas que permiten la evolución de los estándares de comunicaciones, además de un incremento en el proceso de comunicación para vincular nodos sensores y sistemas robóticos, gracias a que las FPGAs, permiten realizar cálculos algorítmicos en paralelo.
- Sistemas de video y visión artificial: Cambio de paradigma al implementar algoritmos en *Hardware* (HW) que se ejecutaban por *Software* (SW), para obtener mayor velocidad y menor consumo de energía, lo que se conoce como Co-diseño de SW a HW (HW/SW Codesign, en inglés).
- Interfaces inteligentes de E/S que incluyen sistemas de Conversión de señales Analógicas a Digitales (Analog-to-Digital Converter (ADC), en inglés) y Conversión de señales Digitales a Analógicas (Digital-to-Analog Converter (DAC), en inglés).
- Sistemas automotrices: Procesamiento de video para asistencia al conductor.
- Control de datos: Interfaces para Discos de Estado Sólido (Solid-State Disks (SSD), en inglés) y para algoritmos relacionados con Machine Learning [29].
- Aplicaciones militares y aeroespaciales: Procesamiento de ondas de radio y de gran cantidad de datos de sensores inalámbricos [29]. También se utilizan para reducir el tiempo y el coste en el desarrollo de sistemas complejos de comunicaciones para satélites y naves espaciales que utilizan protocolos de alta velocidad [30].

Pang y Membrey en [31], señalan que las FPGAs tienen la capacidad de “moldearse” para realizar y asumir cualquier función de HW que se pueda imaginar, y que se utilizan ampliamente para solucionar problemas de cálculo computacional, ya que se benefician de su arquitectura paralela, debido a que actualmente existe dificultad para diseñar un circuito integrado (CI) que ejecute un funcionamiento personalizado. Mencionan que una ventaja de las FPGAs es que se puede implementar en ellas arquitecturas que respondan a diseños o funcionamientos personalizados, puesto que no es como un MCU que tiene una arquitectura definida en la que se ejecuta una tarea por SW. Añaden que la gran diferencia es que las FPGAs brindan lo último en flexibilidad de HW, lo que permite diseñar cualquier modelo digital. Es entendible que, para el funcionamiento correcto de un sistema, es necesario diseñar todas sus características y las reglas a seguir, para que entonces se ejecute dentro de la FPGA. En otras palabras, la FPGA tiene la capacidad de definir las características que requiera el diseño.

Las herramientas de las FPGAs reducen operaciones complejas en operaciones simples, a este proceso se le denomina reducción de órdenes. Al efectuar esta reducción, las operaciones complejas se ejecutan más rápido y consume menos energía, pues las FPGAs paralelizan las tareas, que al ejecutarse por SW en una CPU son mucho más lentas. Además, el flujo de datos se puede expandir dentro y fuera de la FPGA, lo que hace que se logre transmitir una gran cantidad de datos a la FPGA, para realizar al mismo tiempo más procesamiento, como por ejemplo paquetes de imágenes, videos o ethernet que necesitan un gran ancho de banda para soportar el flujo de datos, sin perder de vista que el procesamiento se realiza por HW en lugar de SW [31].

Hasta hace una década, el esfuerzo de los fabricantes de FPGAs ha consistido en desarrollar FPGAs cada vez más rápidas y con un mayor número de recursos lógicos, debido a que la mayoría de las aplicaciones estaban enfocadas a aumentar el rendimiento computacional. En la actualidad, los entornos de desarrollo con FPGAs facilitan enormemente su utilización, que junto con las familias de bajo coste que poseen la mayoría de los fabricantes, permiten que el campo de aplicación de las FPGAs no se limite solamente al HPC. La evolución de FPGAs ha permitido que el HW/SW *Codesign* sea una realidad, pudiendo trasladar al HW tareas que se han realizado habitualmente por SW, consiguiéndose no solamente aumentar el rendimiento del sistema, sino aliviar al procesador de otras tareas críticas que deben ejecutarse en tiempo real y que cualquier modificación del programa tiene un impacto en estas tareas.

Hasta hace poco tiempo, no era muy común utilizar FPGAs como un componente más del nodo inalámbrico, por varias razones, sobre todo, por su elevado coste y consumo de potencia. Incluso, en la actualidad solamente se utilizan en aquellos casos en los que se quiere aumentar enormemente el rendimiento computacional del nodo y se han olvidado de otras aplicaciones, que como se verá a lo largo de este trabajo, resultan muy eficientes. En la actualidad el campo de aplicación de las FPGAs se ha extendido enormemente y se integran en un sistema electrónico cuando se quiere:

- Aumentar el rendimiento (HPC).
- Realizar tareas con un control estricto de tiempo real.
- Implementar periféricos que no contienen la CPU o el microcontrolador (MCU) seleccionado.
- Realizar la interfaz a periféricos externos.
- Lógica de pegamento (glue logic, en inglés) entre componentes del sistema.

Los cuatro últimos casos de uso de las FPGAs están relacionados con la necesidad de la creación de un periférico ad-hoc por HW. Estos dispositivos permiten su configuración HW, incluso de forma dinámica, por lo que se utilizan habitualmente para realizar tareas de procesamiento de elevada complejidad, y que deben tener una respuesta adecuada en tiempo real [32]. Una FPGA permite implementar por HW cualquier periférico o circuito de aplicación específica y hasta una o varias CPU, funcionando todos en paralelo. Por tanto, es un dispositivo muy versátil, que permite implementar sistemas electrónicos digitales muy eficientes y a medida, en comparación con una clásica CPU [33]. Las ventajas de las FPGAs en términos de coste, tamaño y consumo de energía facilitan su utilización en plataformas para comunicación inalámbrica [34,35].

1.3. Motivación y objetivos del trabajo

La motivación de este trabajo es mostrar que, en algunas aplicaciones, como son las redes inalámbricas del alumbrado público, las FPGAs no solo se pueden utilizar como un componente más del nodo para HPC, sino también, para realizar otras tareas, relacionadas con la creación de periféricos e interfaz a periféricos. Se demuestra así, que la utilización de FPGAs, de bajo coste y consumo de potencia, se pueden utilizar como parte del nodo inalámbrico al igual que se han venido utilizando en cualquier otro sistema electrónico.

El grupo de investigación de Instrumentación y Electrónica Industrial (IEI) de la Universidad de Córdoba, dentro del cual se desarrolla este trabajo, tiene especial interés en los sistemas de alumbrado público, en los que la comunicación se realiza vía inalámbrica y el control de las luminarias se realiza con el protocolo DALI. Los sistemas de iluminación basados en el protocolo DALI son los más adecuados para los sistemas de alumbrado público por las ventajas indicadas anteriormente, permitiendo realizar operaciones de control digital de la iluminación y otras tareas de gestión (por ejemplo, monitoreo de energía), además de lograr un ahorro de energía y adaptar la iluminación a las condiciones lumínicas y a las necesidades de cada momento [36-38].

El alumbrado público tiene un consumo de energía eléctrica elevado y aunque existen propuestas de solución para reducir el consumo adaptando la luminaria a las necesidades reales mediante el sensado y los algoritmos de control inteligente, estas presentan desventajas palpables debido a que tienen dificultades en la implementación de aplicaciones ante las altas exigencias de procesamiento computacional, de las restricciones de ejecución y de comunicación, entre otras. Algunos de los problemas en las WSNs para controlar sistemas de SSL, es que los nodos deben soportar numerosas conexiones a sensores y con diferentes interfaces. Además, necesitan aumentar la capacidad para el pre-procesamiento de los datos adquiridos por los sensores y aplicar incluso sobre los datos técnicas de Inteligencia Artificial, *Edge Computing* o *Machine Learning*.

El grupo de investigación IEI ha desarrollado nodos inalámbricos con interfaz DALI [27], pero, al igual que la mayoría de los trabajos publicados hasta el momento, los MCUs utilizados en los nodos, no tienen la interfaz DALI integrada, por lo que el protocolo DALI ha sido implementado por SW y utilizado en diferentes prototipos de iluminación [39-41]. La implementación de DALI por SW como un periférico virtual tiene como limitante el consumo de recursos del MCU (habitualmente temporizadores HW) y el consumo de tiempo de procesamiento. Además, la realización de “periféricos virtuales” por SW tiene unos requisitos temporales estrictos, debiéndose reservar un cierto tiempo de procesamiento y consume recursos HW del MCU. La integración de periféricos virtuales como parte de las aplicaciones, hace un uso excesivo de *interrupt services routines* asociadas a las interrupciones y requiere un desarrollo cuidadoso del SW. Cualquier nuevo servicio que se implemente en el nodo, debería tener en cuenta el impacto en las interfaces y periféricos virtuales realizados por SW. Por tanto, surge la necesidad de liberar al MCU de la implementación de DALI como un periférico virtual. Una de las maneras más sencillas y eficientes es crear un *bridge* entre el MCU y DALI Bus. Este *bridge* se debería comunicar

con el MCU a través de un enlace como *Universal Asynchronous Receiver-Transmitter* (UART) o *Serial Peripheral Interface* (SPI) que están implementados en casi todos los MCUs comerciales.

Programar el protocolo DALI como un *bridge* en dispositivos como PLCs o en placas de desarrollo comerciales, reducen su capacidad de cómputo por la necesidad de ejecutar diferentes procesos obligatorios relacionados con el protocolo DALI. Además, el alto consumo de energía y el coste de los dispositivos los hacen imprácticos para los requerimientos de portabilidad y flexibilidad de los nodos sensores inalámbricos para aplicaciones de IoT.

Una de las opciones para implementar un *bridge* DALI es la implementación por HW. La implementación por HW tiene muchas ventajas frente a la realización del *bridge* por SW. Todos los problemas de tiempo real, que puedan aparecer en el diseño del *bridge* por SW, desaparecen si se implementa por HW. Además, la implementación por HW tiene otras ventajas para implementar las capas bajas de comunicaciones. La realización de tareas por HW no es nueva y es la base del HW/SW *Codesign* [42]. Es una de las mejores opciones cuando estas tareas se tienen que ejecutar en tiempo real estricto, son de naturaleza paralela o deben tener un rendimiento elevado y un consumo de energía limitado.

En definitiva, el objetivo de este trabajo es la implementación del protocolo DALI (*bridge* DALI) en HW mediante una FPGA de bajo coste y bajo consumo de potencia en un nodo sensor inalámbrico para aplicaciones de alumbrado público en IoT, que permita la ejecución del protocolo DALI en tiempo real. Se trata de crear un IP *Core* que implemente el protocolo DALI abierto, portable y que se pueda utilizar en cualquier nodo inalámbrico que disponga de una FPGA.

1.4. Estructura de la tesis

El presente documento está estructurado de la siguiente manera: El capítulo 2 está dedicado a la definición de los conceptos y consideraciones generales de las WSNs, sistemas de iluminación inteligente y de las FPGAs.

El capítulo 3 contiene una revisión de los trabajos relacionados con el trabajo presentado en esta tesis.

La Sección 4 presenta la arquitectura del *bridge* DALI para ser implementado en cualquier FPGA, tomando en cuenta los protocolos de comunicación con el MCU y la interfaz DALI.

La Sección 5 muestra los resultados en un nodo real, comenzando por la arquitectura del nodo sensor inalámbrico, los resultados de la síntesis de la implementación en la FPGA del protocolo DALI, del tiempo de transmisión del flujo de datos a través del *bridge*, la frecuencia de funcionamiento máxima que puede soportar el *bridge*, el consumo de potencia y los tiempos de sincronización de las transiciones del protocolo DALI. También se exponen los resultados experimentales y las pruebas eléctricas de los componentes que se encuentran embebidos en la Placa de Circuito Impreso (Printed Circuit Board (PCB), en inglés) que engloba al nodo sensor inalámbrico.

Finalmente, en la Sección 6 se establecen las conclusiones de la tesis, así como también se enumeran las publicaciones de los resultados obtenidos en la etapa de diseño y descripción del *bridge* DALI

Capítulo 2. Conceptos generales de las WSNs y de las FPGAs

En este capítulo se presenta una descripción de los conceptos de las WSNs, resaltando la relevancia que ha tenido la evolución, la arquitectura y las topologías en el desarrollo de las WSNs. También se puntualiza el elemento central de las WSNs que es el nodo sensor inalámbrico, delineando su estructura y sus aplicaciones. Se hace notar la trascendencia de los nodos sensores inalámbricos en los sistemas de SSL y se considera la importancia de los protocolos para el control de iluminación. Asimismo, se plasma un contexto general de las FPGAs destacando las ventajas que genera la implementación de sistemas en HW, tomando en cuenta la evolución, la arquitectura y la integración en diferentes aplicaciones de las FPGAs. Para lo cual, se señalan las características de las Máquinas de Estados Finitos (Finite State Machine (FSM), en inglés). Se da realce a las características que proporcionan las FPGAs de “gama baja”, esto debido a que en el proyecto que se presenta se pretende implementar un *bridge* de comunicación en un dispositivo de este tipo.

2.1. Redes inalámbricas de sensores

Existen innumerables aplicaciones de las WSNs y los potenciales de mercado son enormes, por lo que es un campo de investigación y desarrollo de la microelectrónica que evoluciona de manera vertiginosa [43]. Las WSNs se diferencian fácilmente con respecto de las redes tradicionales por las ventajas de proporcionar un extenso rango de transmisión, la facilidad de implementación y la auto-organización [44], que contrastan con las limitaciones de energía y la baja velocidad del flujo de datos [45]. Las ventajas mencionadas permiten la aplicación de las WSNs en el monitoreo inalámbrico de servicios públicos, ambientales y el seguimiento de objetos, pero con la necesidad de disminuir el tamaño y coste, y aumentar la capacidad de procesamiento en los nodos sensores inalámbricos. Los

nodos sensores inalámbricos también deben disponer de diferentes interfaces para comunicarse con otros dispositivos debido a que presentan inconvenientes en la limitada capacidad de procesamiento y de memoria [46,47]. Además, deben cumplir con las necesidades de la aplicación específica, el objetivo del diseño, el HW, las limitaciones del sistema y otros factores como su ubicación [48]. A partir de lo anterior se observa que los nodos sensores inalámbricos tienen necesidades y requisitos estrictos, por lo cual la investigación, desarrollo e implementación en las WSNs genera un desafío, por la amplia cantidad de conocimientos requeridos en diversas áreas [49].

Existen varias formas de definir a las WSNs, como se puede observar en [44,50,51] donde mencionan que las WSNs se componen de una gran cantidad de sensores autónomos inteligentes, de bajo coste y consumo de energía, distribuidos espacialmente con una o más estaciones base que monitorean condiciones ambientales o físicas de manera cooperativa. Por ejemplo, en [52] indican que las WSNs consisten específicamente de sensores autónomos distribuidos para monitorear y transmitir la información obtenida a un destino específico. Una WSN también se considera un conjunto de dispositivos interrelacionados, distribuidos, autónomos, pequeños y de poco consumo de energía que se denominan nodos sensores inalámbricos. Físicamente se encuentran dispersos, interconectados en red para recopilar, procesar y enviar datos, con capacidades de procesamiento y computo limitado [53]. En conclusión, las WSNs se consideran como una infraestructura formada por pequeños dispositivos dotados de elementos de sensado, medición, monitorización, instrumentación, procesamiento computacional y comunicación, con la capacidad de observar y reaccionar a eventos y fenómenos en un entorno específico [3,54,55].

2.1.1. Evolución de las WSNs

Existen diferentes perspectivas para exponer la evolución de las WSNs. Se puede considerar a la Agencia de Proyectos de Investigación Avanzada de Defensa (Defense Advanced Research Projects Agency (DARPA), en inglés) de Estados Unidos de América (EUA) como punta de lanza en la investigación de las WSNs. En el transcurso del año 1980 realizaron para el ejército de los EUA el proyecto denominado *Redes de Sensores Distribuidos* (Distributed Sensor Networks (DSN), en inglés) [56]. En esos años no existían las PCs como tal ni las estaciones de trabajo (workstations, en inglés). Entonces el procesamiento de datos, que es la parte más importante de los sistemas computacionales, se ejecutaba generalmente en miniordenadores o mainframes [57]. DARPA continuo con la

investigación en WSN al desarrollar el programa *Sensor Information Technology* (SensIT) [58]. Gracias al avance tecnológico continuo en la computación, la microelectrónica y las comunicaciones, ha crecido exponencialmente la investigación y el desarrollo de las WSNs.

La Universidad de California en Los Ángeles, en colaboración con el Centro de Ciencia Rockwell, presentaron el proyecto de Red Inalámbrica de Sensores Integrados (Wireless Integrated Network Sensors (WINS), en inglés) [59], donde desarrollaron en 1996 un Microsensor Integrado Inalámbrico de Baja Potencia (Low Power Wireless Integrated Microsensor (LWIM), en inglés) [60]. El microsensor fue realizado con base a chips Semiconductores de Óxido Metálico Complementario (Complementary Metal Oxide Semiconductor (CMOS), en inglés), integrando sensores, interfaces, circuitos de procesamiento de señales digitales, radio inalámbrica y el MCU. También, la Universidad de California en Berkeley efectuó el proyecto denominado *Smart Dust* [61], donde desarrollaron pequeños nodos sensores inalámbricos o motas con el objetivo de demostrar que el sistema completo podía ser soportado por los pequeños sensores. Las motas se pueden definir como la colección de dispositivos multifuncionales y energéticamente eficientes para soportar una red [53].

El Centro de Investigación Inalámbrica de Berkeley (Berkeley Wireless Research Center (BWRC), en inglés) ejecutó el proyecto PicoRadio [62], teniendo como objetivo realizar sensores de baja potencia, con la innovación de que la alimentación de energía eléctrica pudiera realizarse con energía solar. En el transcurso del año 2000, el Instituto de Tecnología de Massachusetts (Massachusetts Institute of Technology (MIT), en inglés) desarrolló μ AMPS (micro-Adaptive Multi-domain Power-aware Sensors, por sus siglas en inglés) utilizando HW de bajo consumo de potencia y un MCU con soporte a variación de voltaje dinámico, con el objetivo de reconfigurar algoritmos para reducir el consumo de energía a nivel de SW [63,64].

La Tabla 2.1 muestra la evolución de los nodos sensores inalámbricos comerciales divididos en tres generaciones [65]. Los factores de fabricación de los nodos sensores inalámbricos que se consideran son: el tamaño, el peso, la implementación, la arquitectura, los protocolos soportados, la topología utilizada, las formas de alimentación de la energía eléctrica y el tiempo de vida. La evolución de la implementación de los sensores, ha generado un cambio en la arquitectura de los nodos sensores inalámbricos y por lo tanto la disminución del tamaño y peso. Los protocolos de comunicación más recientes, permiten en gran medida el despliegue de las WSNs. Por ejemplo, los estándares IEEE 802.11a/b/g, Bluetooth, ZigBee y WiMax, soportan una conectividad confiable y generalizada. Las

topologías han cambiado y se han adecuado a la comunicación de diferentes tipos de datos. El perfeccionamiento en el suministro de la energía ha extendido el tiempo de vida de los nodos sensores inalámbricos.

Tabla 2.1 Evolución de las WSNs

	Primera generación (Década de 1980 a 1990)	Segunda generación (Década de 1990 a 2000)	Tercera generación (Del 2000 en adelante)
Tamaño	Grande	Mediano	Pequeño
Peso	Algunos kilogramos	Muy pocos kilogramos	Gramos
Modo de implementación	Instalado físicamente o lanzado al aire	Colocado a mano	Embebido
Arquitectura del nodo sensor	Sensado, procesamiento y comunicación separados	Sensado, procesamiento y comunicación parcialmente integrados	Sensado, procesamiento y comunicación integrados
Protocolos o estándares	Propietario	Propietario	Wi-Fi, ZigBee, WiMax, LoRa, etc.
Topología	Punto a punto, estrella y multisalto	Cliente-servidor y <i>peer-to-peer</i>	Totalmente <i>peer-to-peer</i>
Fuente de energía eléctrica	Baterías grandes o a través de la línea eléctrica	Baterías tamaño AA o similar	Baterías de diferentes tamaños o solar
Tiempo de vida	Horas, días o un poco más.	Días o hasta semanas	Meses o años

Actualmente, el progreso en los sistemas embebidos ha facilitado el avance en las tecnologías de comunicación y las capacidades de procesamiento de la información en los sistemas de red. Específicamente, las WSNs tienen un funcionamiento más conveniente en ambientes altamente dinámicos. La evolución de las WSNs tiene un nuevo paradigma, que es el desarrollo de sensores de bajo coste y consumo de energía desde las perspectiva de los sistemas embebidos [46,66]. De aquí que las investigaciones más recientes en WSN se enfocan en mejorar el funcionamiento de los nodos sensores inalámbricos. El objetivo es reducir el tamaño y el coste. Además, históricamente el desarrollo de los nodos sensores inalámbricos inteligentes ha globalizado el interés por las WSNs [48].

2.1.2. Concepto del IoT

Millones de WSNs locales interconectadas dan soporte al IoT, y gracias a esto, en la actualidad el IoT continúa creciendo rápidamente. De hecho, el IoT es la base de lo que se conoce como la cuarta revolución industrial y la transformación digital de los negocios y la sociedad [67], por la facilidad de interconectar máquinas y cosas, además de administrar la información sobre materiales y el uso de la energía que fluyen en sistemas ciber-físicos [68]. El término IoT fue formulado por Kevin Ashton, citando que el IoT “tendría el potencial de cambiar el mundo”. Posteriormente, en 2005 el IoT fue presentado

formalmente en el Informe de Internet de la Unión Internacional de Telecomunicaciones [69].

El IoT lleva muchos años en desarrollo, de hecho, el concepto de utilizar dispositivos sensores para recopilar datos y luego transferirlos a aplicaciones a través de una red existe desde hace varias décadas. Sin embargo, los continuos cambios en la última década en el mercado y las tendencias tecnológicas han alterado drásticamente la propuesta de valor de los sensores y actuadores interconectados. La combinación de HW de bajo coste y tecnologías de WSNs ha permitido una nueva generación de dispositivos sensores compactos con conectividad a través de Internet, lo que da como resultado que el IoT facilite la recopilación y el intercambio de datos en tiempo real, ofreciendo una visibilidad y un control sin precedente de los activos, el personal, las operaciones y los procesos. Al mismo tiempo, el utilizar de forma globalizada infraestructuras basadas en la nube para procesamiento computacional y/o almacenamiento de datos, permite introducir capacidades de análisis de datos, lo que ha marcado el comienzo de una nueva era en la toma de decisiones, control y automatización inteligentes [68].

A medida que Internet evoluciona hacia el IoT, cambia el enfoque, de la conectividad, a los datos. El IoT trata principalmente de obtener información procesable a partir de datos, desde una perspectiva tecnológica, lo que se puede conseguir con la disponibilidad de protocolos de red que cumplan con los requisitos y satisfagan las limitaciones de los nuevos dispositivos de IoT, y lo que es más importante, con la disponibilidad de interfaces y estándares para servicios de aplicaciones, incluido el acceso a datos, almacenamiento, análisis y gestión. El camino hacia el IoT basado en estándares está en marcha, sin embargo, quedan muchas líneas de oportunidad para la investigación y el desarrollo, especialmente con respecto a la interoperabilidad de las aplicaciones, las interfaces programables comunes y la semántica de datos [68].

La idea principal del IoT es conectar físicamente cualquier cosa, en pocas palabras “todo” (por ejemplo, sensores, dispositivos, máquinas, personas, animales, árboles) y realizar procesos a través del Internet para monitorear y/o controlar su funcionalidad. Las conexiones no deben limitarse a sitios de información, sino que deben ser conexiones reales y físicas que permitan a los usuarios llegar a “cosas” y tomar el control de ellas cuando sea necesario. En síntesis, el objetivo del IoT es permitir que los objetos que rodean a las personas monitoricen su entorno de manera eficiente, se comuniquen de forma económica y creen un mejor ambiente para la gente, en donde los objetos que utilizan cotidianamente actúen en función a las necesidades y gustos del usuario, sin instrucciones explícitas [69].

2.1.3. Arquitectura de las redes inalámbricas de sensores

La arquitectura general de la WSNs se compone de un conjunto de nodos de sensores inalámbricos, un campo de sensado, enlace de radio, enrutadores, algunos nodos recolectores también denominados como nodos *sink* (estaciones base o puerta de enlace); la infraestructura de red como Internet o vía satélite y el nodo administrador de tareas o usuario [56,70-72]. Lo anterior se muestra en la Figura 2.1. En la actualidad las WSNs incluyen actuadores para activar diferentes procesos [70].

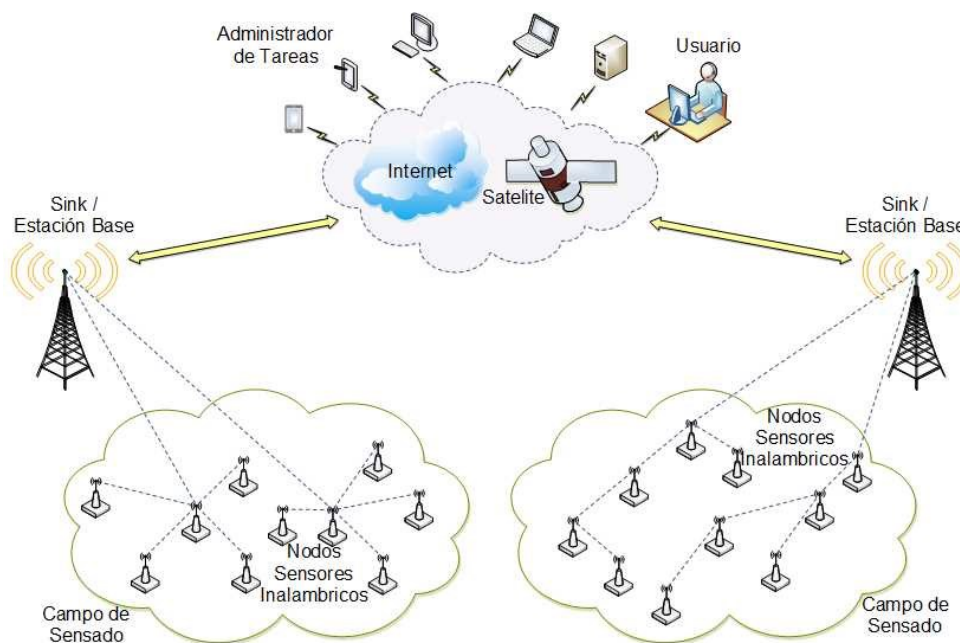


Figura 2.1 Arquitectura básica de las WSNs

De forma general la arquitectura de las WSNs se clasifica en estructuras básicas y jerárquicas [54]. Donde las arquitecturas básicas se pueden configurar para realizar comunicación entre nodos sensores inalámbricos homogéneos con conexiones Muchos a Uno (M:1) o Muchos a Muchos (M:N). En conexión M:1, los nodos sensores inalámbricos homogéneos envían datos a un nodo *sink*. En conexión M:N, varios nodos de sensores inalámbricos envían datos a varios nodos *sink*. En las arquitecturas jerárquicas, los nodos *sink* reciben y almacenan datos de diferentes nodos sensores inalámbricos, para después enviarlos a un punto de acceso para realizar el procesamiento [49]. En otras palabras, los nodos *sink* reciben información de los nodos sensores inalámbricos y generan el vínculo

con otros nodos *sink* [46,73], pudiendo ser la conexión en la misma WSN o en otra diferente [51,74,75].

El funcionamiento de las WSNs se realiza mediante una estación base que se comunica por medio de un enlace de radio inalámbrico con los diferentes nodos sensores inalámbricos que se encuentran dispersos en el campo de monitorización [76]. Los datos que se capturan por medio de un nodo sensor inalámbrico se comprimen para transmitirlos a la estación base, de ser posible en forma directa o, en caso contrario, haciendo uso de otros nodos sensores inalámbricos para realizar una ruta de conexión hacia la estación base. La estación base también tiene comunicación con la infraestructura de red para transmitir los datos desde la WSN a un sistema central alojado en un servidor y/o en la nube para ser utilizados por el IoT. Se puede decir que las estaciones base actúan como *gateway* entre los nodos sensores inalámbricos y el usuario final [72]. Los enrutadores se encargan de procesar, calcular y distribuir las tablas de enrutamiento, ya que cada nodo sensor inalámbrico debe soportar algún algoritmo de enrutamiento para ejecutar los saltos necesarios [70]. La aplicación de la WSN determina el tipo de arquitectura a implementar. Como se puede observar, las WSNs tienen una dependencia total de los nodos sensores inalámbricos y de la estabilidad en la comunicación de los mismos, por lo que el desarrollo de nodos sensores inalámbricos genera por sí mismo un desafío.

En la Figura 2.2 se muestra la conexión de los diferentes dispositivos de una WSN. Donde, el nodo sensor inalámbrico (SI) monitoriza información y la envía a un dispositivo inteligente (DI) para que la almacene y la procese. El DI tiene la capacidad de generar y recibir información desde otros dispositivos, como D1 y D2, además de tener comunicación con la estación base o *gateway* (G). D1 y D2 serían dispositivos inteligentes con sensores y funciones programables que pueden ser controlados por DI o por el *gateway*. El actuador (A) se activa y desactiva por medio del *gateway*, que es el encargado de dar acceso a los elementos de la WSN hacia la infraestructura de red como Internet o vía satélite [2].

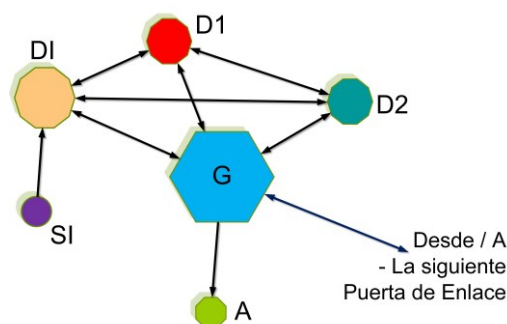


Figura 2.2 Entorno local de una WSN [2]

2.1.4. Topología

El proceso de implementación general de una WSN se inicia con el reconocimiento del entorno de la red de los nodos sensores inalámbricos, al transmitir y recibir su estado a otros nodos sensores inalámbricos. Después se organiza la conexión de los nodos sensores inalámbricos de la red de acuerdo con una topología determinada. Es así que se pueden calcular las rutas adecuadas para transmitir los datos sensados [71]. Lo anterior se puede observar en la Figura 2.3.

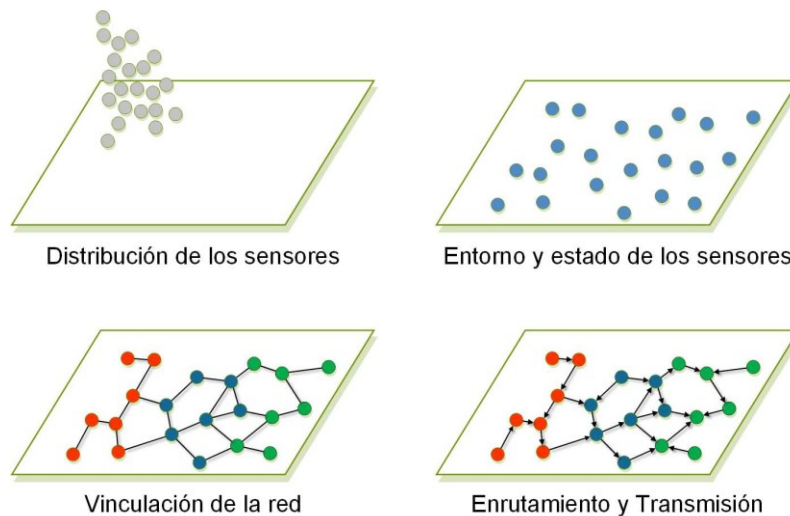


Figura 2.3 Fases iniciales de implementación de una WSN [77]

Con el objetivo de tener una cobertura más grande, la WSN puede utilizar el modo de transmisión multisalto. De esta forma los nodos sensores inalámbricos actúan como receptores y transmisores. El primer nodo sensor inalámbrico denominado nodo “origen” envía datos a un nodo sensor inalámbrico contiguo para transmitirlos a la puerta de enlace. Existe la posibilidad de que el nodo sensor inalámbrico contiguo reenvíe los datos a algunos de los nodos sensores inalámbricos colindantes que se encuentren en la ruta hacia la puerta de enlace. El proceso de reenviar los datos puede ser redundante, esto hasta que los datos lleguen a la puerta de enlace [71].

Los protocolos y las técnicas de implementación de las WSNs son parametrizables de acuerdo a la arquitectura de la WSN, teniendo en cuenta las características especiales de las WSNs que son: la auto-organización, la auto-adaptación, la energía limitada de los nodos sensores inalámbricos y los enlaces de transmisión inestables [71].

Las redes inalámbricas ZigBee por lo general se organizan en topologías: estrella, malla (*peer-to-peer*) y clúster-árbol [65]. Las redes inalámbricas que utilizan la especificación del estándar IEEE 802.15.4 de forma habitual se establecen con topologías: estrella y *peer-to-peer*. La topología clúster-árbol se considera un caso específico de la topología malla (*peer-to-peer*) [78]. Las topologías tienen un gran impacto en el rendimiento, en la extensión o alcance y en el consumo de energía de las WSNs [49]. La Figura 2.4 muestra las principales topologías de las WSNs.

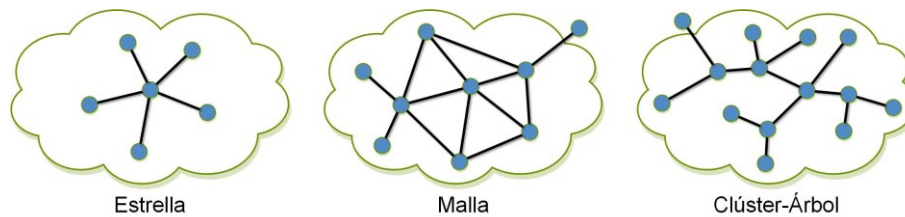


Figura 2.4 Topologías de las WSNs

La topología estrella es la topología más básica, en la cual todos los nodos sensores inalámbricos están enlazados directamente o conectados por un solo salto a un nodo *sink* [49]. Esta topología restringe la longitud o distancia total que puede alcanzar la red. El nodo *sink* debe poseer bastantes recursos para la comunicación y procesamiento. El nodo *sink* realiza funciones de concentrador, enrutador y conmutador. En esta topología es fácil monitorizar un error en algún nodo sensor inalámbrico o en el nodo *sink*, además, ante un fallo, en un nodo sensor inalámbrico la WSN sigue funcionando, a menos que el fallo sea el nodo *sink*. En las WSNs la mayor parte de transmisión de datos fluye en topología estrella. [79]. Esta es una de las topologías que soporta el estándar IEEE 802.15.4 para el diseño de una red.

En la topología malla, los nodos sensores inalámbricos se conectan mediante múltiples rutas donde los datos se envían salto a salto al nodo *sink*, característica que la hace tolerante a fallos [49]. La confiabilidad de esta topología es debido a que las rutas de comunicación de los nodos sensores inalámbricos son redundantes. Si un nodo sensor inalámbrico tiene un fallo, la red de forma automática calcula una nueva ruta de transmisión, pero este tipo de topología tiene el inconveniente de que aumenta la latencia en la red debido a la cantidad de saltos que deben efectuar los datos antes de llegar al nodo *sink*. En esta distribución los nodos sensores inalámbricos además de capturar y difundir sus propios datos, se utilizan como nodos retransmisores, en otras palabras, colaboran en la transmisión de datos de un nodo sensor inalámbrico a el nodo *sink* [80]. En este tipo de topología, el cálculo o diseño

de las rutas de comunicación lo realiza el enrutador. Cuando un nodo sensor inalámbrico envía datos a el nodo *sink*, el enrutador se encarga de organizar el camino a seguir para cumplir con dicho propósito. La topología malla tiene la flexibilidad de organizarse de forma completa o parcial. Completa es cuando todos los nodos sensores inalámbricos están conectados directamente entre ellos. Parcial es cuando algunos nodos sensores inalámbricos están organizados en topología malla completa y otros se enlazan de forma indirecta entre ellos. Esta última forma tiene desventajas, ya que se minimiza coste a cambio de confiabilidad.

En las WSNs los nodos sensores inalámbricos deben tener la capacidad de ser movibles, de estar protegidos y de ser interferidos, por lo tanto, se puede decir que su posición es aleatoria. De aquí que la auto-organización de los nodos sensores inalámbricos es un punto clave en las WSNs para cumplir con la solidez de la red, por lo que se desarrolló la topología de malla inteligente. En esta topología la comunicación se realiza cuando un nodo sensor inalámbrico monitorea a los nodos sensores inalámbricos adyacentes para determinar la intensidad de la señal y, en base a ésta, realiza la sincronización con el nodo sensor inalámbrico conveniente para enviarle la solicitud de vinculación. Posteriormente, el nodo sensor inalámbrico, elegido para el vínculo de comunicación, envía una solicitud al nodo *sink* para que este asigne los recursos de red para realizar la comunicación. Al nodo sensor inalámbrico se le pueden asignar dos o más rutas de transmisión para mejorar la confiabilidad de la WSN [81]. La formación de los caminos de transmisión se realiza mediante algoritmos de enrutamiento controlados por árboles y tablas, por lo que los enrutadores y los receptores de radio del nodo *sink* siempre permanecen encendidos [65].

La topología clúster o árbol es una organización algo compleja ya que cada nodo sensor inalámbrico debe mantener una única ruta de comunicación con el nodo *sink* o estación base. La ruta se puede soportar sobre otros nodos sensores inalámbricos para transmitir los datos. Este tipo de topología permite que se formen redes *peer-to-peer* utilizando enrutamiento de saltos múltiples con poca sobrecarga de enrutamiento [65]. Está auto-organizada y permite redundancia de transmisión de datos en la red, lo que hace tener un alto grado de soporte a fallos y realizar reparación de forma automática. Esta topología puede ser significativamente grande lo que permite abarcar físicamente grandes áreas. También, se utiliza para aumentar el campo de monitorización de las WSNs. Está es una topología frecuentemente empleada en sistemas de alumbrado público, ya que puede tener una o varias ramas.

La topología híbrida es una combinación de la topología estrella y la topología malla. En esta topología los nodos sensores inalámbricos que se encuentran adyacentes al nodo *sink* se enlazan en topología estrella, y los nodos sensores inalámbricos que no se encuentran dentro del rango de transmisión directo del nodo *sink* se enlazan en topología malla [49]. Esta provee simultáneamente los beneficios de las topologías estrella y malla que son respetivamente, el bajo consumo de energía y la tolerancia a fallos, lo que hace que esta topología sea muy utilizada en las WSNs [49]. Las topologías dinámicas deben cumplir con requisitos como la limitación de recursos en los nodos sensores inalámbricos y la distribución de los mismos en la WSN [44]. Las redes ad-hoc deben tener nodos sensores inalámbricos idénticos o con igualdad de recursos y condiciones, además de no existir un nodo *sink* [3], lo que permite la libre movilidad de los nodos sensores inalámbricos, para implementar una red temporal sin la necesidad de infraestructura externa.

2.1.5. **Nodo inalámbrico**

La evolución en la comunicación inalámbrica ha hecho posible tener un entorno inteligente a nuestro alrededor. El entorno inteligente depende de forma obligatoria de datos sensados del mundo real. Los datos sensados provienen de la colección de nodos sensores inalámbricos ubicados de manera distribuida y específica para implementar una WSN [51,74,75]. La función del nodo sensor inalámbrico, es el monitorizar datos de su entorno y transmitirlos. En otras palabras los nodos sensores inalámbricos recogen el evento físico del área de monitorización y lo transmiten a otros nodos sensores mediante un salto directo o multsaltos [46,73]. Un nodo sensor inalámbrico es un dispositivo diminuto que incluye diferentes módulos o subsistemas para realizar cada uno funciones importantes. Por lo general, los módulos son: de sensado o monitorización de datos, de procesamiento computacional de los datos, de comunicación inalámbrica o transceptor y una fuente de energía eléctrica [65,73,82-84]. El desarrollo de los nodos sensores inalámbricos históricamente se ha enfocado en disminuir el tamaño, el coste y el consumo de energía [74]. En la Figura 2.5 se muestra la estructura de un nodo sensor inalámbrico.

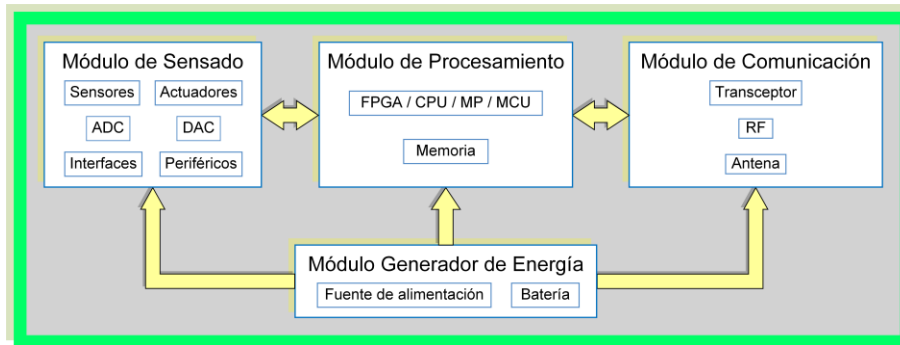


Figura 2.5 Arquitectura general del HW de un nodo sensor inalámbrico.

Esta investigación está especialmente enfocada a los nodos de sensores inalámbricos inteligentes, específicamente, en el desarrollo de nodos sensores inalámbricos que posean dispositivos embebidos para realizar procesamiento computacional, pero con recursos mínimos de HW para ahorrar coste y consumo de energía.

El módulo de sensado o monitorización recopila los datos que caracterizan a los diferentes fenómenos a evaluar. Se utilizan sensores que dependen de la aplicación del nodo, por ejemplo, de iluminación, de temperatura, de humedad, de acústica, de campo magnético, etc. Por lo general, los fenómenos son medidos con transductores que forman señales eléctricas analógicas, que se convierten a valores digitales por medio de convertidores de señal analógica a digital, y después ser procesados [65,73,82-84].

Históricamente el procesamiento computacional de los nodos de sensores inalámbricos se ha realizado generalmente a través de MCUs y memorias [83] y existen nodos sensores inalámbricos con capacidad de memoria limitada o insuficiente que dificulta el procesamiento de la información [85]. El módulo de procesamiento computacional se encarga de procesar (con una CPU) y/o almacenar (en memoria) los datos sensados y controla la ejecución de los demás módulos. Es claro, que la capacidad de procesamiento del nodo sensor inalámbrico está limitada o depende de las características de la CPU [65,73,82-84].

El módulo de comunicación inalámbrica o transceptor realiza la transmisión y recepción de información del nodo sensor con los demás nodos. El canal de comunicación para transmitir los datos por la red puede ser en radiofrecuencia (RF), infrarroja, acústica, o óptica. Este proceso es el que consume la mayor cantidad de energía en el nodo sensor inalámbrico. Es primordial que se disminuya el consumo de energía realizando una comunicación independiente del procesamiento de datos [65,73,82-84]. La estabilidad del

nodo sensor inalámbrico es fundamental ya que el fallo de uno causa inestabilidad en la WSN [44].

El módulo generador de energía eléctrica provee el potencial eléctrico a los demás módulos. Es el módulo crítico de los nodos sensores inalámbricos ya que de éste depende el tiempo de vida del nodo sensor inalámbrico [65,73,82-84].

En síntesis, es inherente entender a las WSNs en función de los nodos sensores inalámbricos. El conjunto de recursos que tienen los nodos sensores inalámbricos permite la globalización de aplicaciones. En la actualidad gracias a la miniaturización de los componentes electrónicos ha permitido disminuir el coste y el consumo de energía tanto en los MCUs como en los módulos de RF. Lo anterior posibilita la integración de nodos sensores inalámbricos en una pequeña PCB, haciéndolos económicos y con un funcionamiento óptimo.

2.2. Iluminación inteligente

La iluminación o alumbrado inteligente se considera como la capacidad de adaptar la intensidad de la luz eléctrica a las condiciones del mundo real [86]. El objetivo de la iluminación inteligente es el control electrónico de lámparas o farolas, para lo cual se utilizan varios dispositivos y/o adaptadores que monitorizan la luz ambiental y miden el consumo de energía para realizar el control de la iluminación, logrando así, mejorar la eficiencia en el consumo de energía [16]. En la actualidad, los sistemas de iluminación inteligente integran varios dispositivos de medición e interfaces de conectividad en un nodo sensor inteligente para el control de la iluminación [87]. Incorporando características como el ajuste de la atenuación y operaciones para optimizar el funcionamiento de las luminarias, permiten reducir el consumo de energía y el mantenimiento, alargando la vida útil de las luminarias y por consecuencia minimizando el coste de todo el sistema de iluminación [16]. Los sistemas de SSL adaptan automáticamente la cantidad de iluminación a las necesidades reales, por ejemplo, atenuando el alumbrado público dependiendo de la cantidad de tráfico en una carretera y de las condiciones de luz natural. Además, debe contar con capacidades de comunicación bidireccional.

Los sistemas de SSL basados en redes inalámbricas permiten controlar y monitorear las características funcionales de las luminarias, como son el consumo de energía, la temperatura, los parámetros eléctricos, entre otros. La WSN también debe recolectar datos, como el tráfico de automóviles y personas, la contaminación atmosférica, la precipitación

pluvial, inundaciones, por lo que los nodos sensores inalámbricos colocados en la red del alumbrado público, deben tener esas capacidades de sensado [86]. Además, los nodos sensores inalámbricos que se encuentran en las farolas, deben poseer diferentes interfaces de comunicación.

El aumento en aplicaciones de SSL cada vez es mayor, con base a la eficacia en su funcionamiento que varias investigaciones e implementaciones han demostrado [88]. Conjuntamente, las lámparas en los sistemas inteligentes de iluminación no se consideran entes aislados, ya que los nodos sensores inalámbricos instalados en ellas generan enlaces multifuncionales capaces de transmitir datos [86]. En diversos desarrollos [89-92], se propusieron diferentes formas de controlar dinámica y remotamente las farolas, basados en centros de control que administran el intercambio de datos entre las farolas. Las WSNs se utilizan en [93] para cambiar el paradigma de las redes del alumbrado público estándar en un sistema de SSL. Esto permite el desarrollo de aplicaciones, como las realizadas en [94-96], de iluminación inteligente para IoT. En el tema de iluminación inteligente también existen investigaciones enfocadas en diseñar controladores de lámparas [97-99] y al control automático de la iluminación [100,101] mediante HW. En esta literatura se puede identificar una carencia de implementación de controladores de iluminación en HW “puro” que liberen de ese procesamiento computacional en los nodos. También se observa que el SSL debe tener la capacidad de generar eficiencia energética, monitoreo en tiempo real y soportar comunicación bidireccional con el fin de solucionar problemas ambientales y de coste de electricidad.

Las diferentes perspectivas que delinear las investigaciones en SSL se visualizan en la Figura 2.6. Los factores que interviene son: la comunicación, el control, la ciberseguridad y la aplicación [7].



Figura 2.6 Componentes del SSL.

La comunicación se puede realizar de tres formas: inalámbricas, cableadas e híbridas. La comunicación inalámbrica se ha clasificado en comunicación de Nivel 1 y comunicación de Nivel 2. El Nivel 1 se utiliza para comunicar una red de alumbrado público que se encuentra a corta distancia, en contraste el Nivel 2 realiza comunicación entre redes de alumbrado público que se encuentran a corta distancia con redes de alumbrado público que se encuentran a largas distancias [7].

La ciberseguridad se enfoca en evaluar la vulnerabilidad del funcionamiento de la iluminación respecto al control no autorizado o hasta la cancelación del servicio [102]. El control se refiere a las estrategias de control que se aplican a la iluminación inteligente para su funcionamiento y pudieran clasificarse en atenuación adaptativa y autónoma. La aplicación es la operación o función específica en la que se utilizan los datos sensados por los nodos [7].

2.2.1. **Protocolos utilizados en el control de iluminación**

La mayoría del alumbrado público en el mundo no cuentan con estrategias o métodos de control de iluminación debido a la falta de equipamiento para realizar comunicación. Posiblemente cuenten con tecnologías de temporizadores ambientales que no cumplen con las necesidades de atenuación de la iluminación. En contraste los sistemas de SSL deben ser eficientes, confiables y armonizados, por lo que agregar a las farolas una interfaz para el control de la intensidad de iluminación es una oportunidad de investigación y desarrollo [7].

Existen diferentes protocolos utilizados habitualmente para el control de la iluminación entre los que se puede mencionar X-10, LONWorks, 0-10 V, DMX 512 y BACNet.

El protocolo X-10 se utiliza habitualmente para controlar iluminación en interiores, emplea modulación de 120 KHz para transmitir información a través de líneas eléctricas de 60 Hz. Algunos controladores con este protocolo no son compatibles con dispositivos de baja potencia debido a la corriente de fuga del dispositivo. Relativamente es lento debido a que se requiere de aproximadamente tres cuartos de segundo para transmitir la dirección del dispositivo y el comando [103,104].

LONWorks es un protocolo propietario que requiere de la CPU Neuron y se utiliza frecuentemente para el control de luminarias por línea eléctrica [105]. En esta tecnología el tráfico de datos es extremadamente irregular [106]. El estándar LonWorks se emplea normalmente enfocado a la automatización de edificios que requieren control de espacios

específicos y la gestión de datos se realiza de forma centralizada y con bajo ancho de banda [107].

En el protocolo 0-10V la señal depende de un voltaje de corriente continua entre 0 y 10 volts. Este protocolo se utiliza de manera ordinaria en el control de iluminación. La interfaz de control se define en el Anexo E de la especificación IEC60929 [22]. Se requiere un cable por canal para controlar cada una de las luminarias, además de un cable “común” para la señal de retorno, por lo que para implementar un sistema de múltiples farolas es necesario de múltiples cables y conectores multinúcleo, lo que hace costoso e inviable un sistema de este tipo [108,109].

El protocolo simple DMX 512 es utilizado habitualmente para el control de iluminación. Permite la transmisión de datos entre la consola de iluminación y los focos, para emitir destellos luminosos o ajustar su atenuación. Es un estándar de transmisión de datos digitales en serie asincrónica que se utiliza generalmente para controlar iluminación. La mala calidad interactiva muestra que es solo un protocolo que se utiliza para el control digital en la iluminación de escenarios artísticos [110]. Tiene desventajas de realizar transmisión de datos unidireccional, no transmite código de dirección, solo transmite información de bucles de control y luminancia, la transmisión no es confiable, no está diseñado para conocer los parámetros eléctricos y el estado de las luminarias [111].

El protocolo BACNet está diseñado para automatizar procesos en edificios. Uno de los procesos en los que se utiliza de forma habitual es el control de iluminación. Es un sistema propietario complejo que requiere de una gran cantidad de memoria para su adecuado funcionamiento [107]. La instalación del cableado debe de cubrir especificaciones estrictas, que lo hace un sistema de alto coste [112].

2.2.2. Protocolo DALI

A diferencia de los protocolos presentados previamente, una de las interfaces principales creadas para los sistemas de iluminación es DALI [8], desarrollada especialmente para controlar digitalmente, luminarias y balastos regulables electrónicos y fluorescentes. Realiza un control robusto y fiable sobre la iluminación de las lámparas. En Europa se utiliza para solucionar problemas de energía, principalmente para fines comerciales e industriales. Es parte de la especificación IEC 60929 [22] y modificado por el estándar IEC 62386 [113] de balastos electrónicos. Los balastos tipo DALI pueden funcionar con niveles de potencia más bajos que los balastos magnéticos estándar [23],

además, soporta luminarias y balastos regulables de diferentes fabricantes [114]. Algunas de las principales ventajas de utilizar el protocolo DALI son:

- Seguridad en la transición de datos.
- Monitorización de los fallos eléctricos o los fallos en las lámparas.
- Notificación del estado de la lámpara.
- Regulación del nivel de potencia de la lámpara.
- Se pueden establecer tolerancias de funcionamiento de las lámparas con valores máximos o predeterminados.
- Optimización del consumo de energía.
- Capacidad de enviar información sobre las características eléctricas de la lámpara. (voltaje, corriente y potencia).
- Control individual o grupal, mediante direccionamiento con Identificadores (IDs).
- Sistema con más funciones y menor coste en comparación con los sistemas de 1–10 V.

Estas son solo algunas de las características que hacen que DALI se considere como la interfaz ideal para realizar el control en sistemas de SSL y, por tanto, se ha seleccionado en este trabajo de investigación. El protocolo DALI es la forma de comunicación digital ideal, simplificada y adaptada a las necesidades de la tecnología de iluminación actual, y soporta el remplazo de luminarias y balastos regulables de diferentes fabricantes [114].

2.3. FPGAs

En esta sección se indican las características de las FPGAs para implementar circuitos lógicos, digitales y unidades de control. Las FPGAs son un tipo especial de CIs o chips que se pueden reconfigurar. Una FPGA es en sí misma, un sistema digital compuesto por bloques de reconfiguración básicos [28]. En la Figura 2.7 se muestra una organización simplificada de una FPGA, la cual está formada por Bloques Lógicos Configurables (Configurable Logic Block (CLB), en inglés), basados en LUT (*Look-Up Table*), Bloques de Puntos de Interconexión Programables (Programmable Interconnections Points (PIPs), en inglés), Bloques de Entrada/Salida (Input/Output Blocks (IOBs), en inglés) y Bloques de Memorias de Acceso Aleatorio (Block Random Access Memory (BRAM), en inglés) [115].

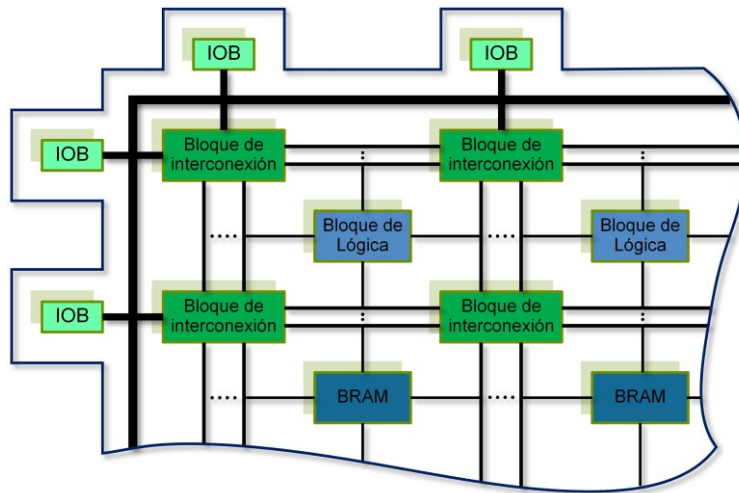


Figura 2.7 Organización simplificada de las FPGAs.

Las hojas de datos de las FPGAs a diferencia de los CIs no detallan función o propósito específico. Lo que se indica son características como el número de puertas lógicas, la cantidad de memoria disponible y cómo reconfigurar la FPGA [31]. Las FPGAs son una tecnología utilizada para diseñar, depurar e implementar soluciones únicas en HW, sin tener que desarrollar dispositivos de silicio personalizados [29]. Una de las ventajas de las FPGAs es que permiten el diseño libre de sistemas digitales y la modificación de los mismos, disminuyendo el ciclo de diseño incluso después de que el producto final se ha implementado. Este dispositivo tiene una característica muy poderosa y atractiva, que es el permitir la creación de diseños personalizados de un circuito digital en HW, por medio de un Lenguaje Descriptivo de HW (Hardware Description Language (HDL), en inglés). Un distintivo muy valioso de las FPGAs es que permiten al desarrollador de sistemas describir o cambiar el diseño del HW en la FPGA a su voluntad, lo que genera otra virtud en las FPGAs, pues son útiles para el desarrollo de productos y la creación de prototipos. La programación se realiza mediante un fichero de configuración, denominado “fichero de bits” y una vez cargado en la FPGA, este se comportará como el circuito digital que se diseñó [31], con la virtud de que, si en el diseño una característica cambia o se descubre un error, se puede reconfigurar la FPGA para implementar un nuevo producto. Al proceso de “programar” la FPGA se le denomina “configurar” [29].

2.3.1. Evolución de las FPGAs

La empresa Xilinx se considera que fue la que desarrolló la primera FPGA en el año 1984, a la que se denominó Matriz de Celdas Lógicas (Logic Cell Array (LCA), en inglés). Cuatro años después la empresa Actel popularizó el término “FPGA”. La XC2064 de Xilinx se considera como la primera FPGA, la cual contenía 64 bloques lógicos, cada uno con dos LUTs de tres entradas y un registro [116]. La capacidad se ha elevado vertiginosamente a lo largo de los años hasta tener millones de puertas y aumentado la velocidad de la frecuencia de funcionamiento [117]. Anteriormente a las FPGAs existían dispositivos, como la Matriz Lógica Programable (Programmable Array Logic (PAL), en inglés), que fusionaban dos bloques, uno de puertas AND y otro de puertas OR, por lo que se implementaban las funciones lógicas como suma de productos, pero ninguno se acerca a la complejidad ni a la versatilidad que ofrecen las FPGAs. Otra diferencia es que estos dispositivos solo se podían configurar una sola vez, ya que se basaban en la tecnología de las Memorias Programables de Solo Lectura (Programmable Read-Only Memory (PROM), en inglés) [118].

La evolución que han tenido las FPGAs a lo largo del tiempo, ha hecho que en la actualidad existan FPGAs de bajo coste que contienen los elementos lógicos necesarios para cumplir adecuadamente con diferentes tipos de aplicaciones [31]. Al surgir la FPGA se diferenció de otros dispositivos lógicos programables en dos características importantes, el enrutamiento reconfigurable para interconectar los recursos lógicos necesarios dentro de un dispositivo y las LUTs programables con registro asociado para implementar las funciones lógicas [29].

Un estudio presentado en [117] divide la evolución de las FPGAs en tres etapas, denominadas: invención, expansión y acumulación. Inicialmente, los Dispositivos Lógicos Programables (Programmable Logic Devices (PLD), en inglés) desarrollados en 1978 por la empresa Monolithic Memories Inc. (MMI), remplazaron las puertas con tecnología Lógica Transistor-Transistor” (Transistor-Transistor Logic (TTL), en inglés) y eran configurables solo una vez, lo que llevó a tratar de desarrollar PLDs reconfigurables basados en tecnologías EEPROM o EPROM [29]. Posteriormente, las PALs configuradas por EPROM quedaron relegados debido a que disponían de una estructura lógica de dos niveles [119,120]. Esto no se pudo solucionar al desarrollar PALs más grandes, ya que no eran prácticos ni en área ni en rendimiento. Para la empresa Altera una solución fue desarrollar

el Dispositivo Lógico Programable Complejo (Complex Programmable Logic Device (CPLD,) en inglés) a inicios de los años 80s, el cual estaba compuesto por varios bloques tipo PAL con conexiones pequeñas entre ellos. La innovación de las FPGAs fue eliminar la matriz (array, en inglés) de las puestas ANDs que proporcionaba la capacidad de programación y distribuir alrededor de la matriz las celdas de memoria de configuración para controlar la funcionalidad y la interconexión. Aunque las FPGAs basadas en Memoria Estática de Acceso Aleatorio (Static Random Access Memory (SRAM) en inglés) eran reconfigurables, requerían una *Flash PROM* (FPROM) externa para almacenar la configuración, de forma que, al aplicarle la tensión de alimentación, se transfiere la información de configuración de la FPROM a la SRAM de la FPGA. La arquitectura de las FPGAs hace que su capacidad y su rendimiento no estén limitados por el crecimiento cuadrático y la disposición de la interconexión de la matriz de ANDs.

La arquitectura de las FPGAs difiere a la de una memoria, ya que el diseño y la fabricación son muy diferentes, por lo que con las FPGAs surgen modelos de implementación lógica completamente nuevos basados en unidades de función personalizada [117]. Todo lo anterior marcó el comienzo de la primera era de las FPGAs, la era de la invención.

A inicios de los años 80's, la primera arquitectura de las FPGAs de Xilinx basada en LUTs de cuatro entradas se consideró *coarse-grained*, pero no todas las entradas se utilizaban, por lo que no se tenía optimización de área, es por esto que se da inicio al desarrollo de arquitecturas con funciones fijas más detalladas para evitar desaprovechar Celdas Lógicas (Logic Cell (LC), en inglés) [121]. El desarrollo en la arquitectura de las FPGAs se enfocó en hacer eficiente la interconexión bidimensional del chip, ya que los enlaces largos de las PALs generaban interconexiones lentas, por lo cual se desarrollaron enlaces cortos para conectar bloques adyacentes y unirlos mediante reconfiguración. Teniendo en cuenta que esto inicialmente se realizaba con transistores que dirigían las señales a través de los segmentos de interconexión, el cambio de modelo hizo más eficiente la interconexión, ya que evitó tener enlaces sin uso, de forma que se compactó el área de interconexión y propició el desarrollo de las FPGAs [117].

La época de la expansión se enfocó en la automatización del diseño. Además, con la ventaja de que en 1992, la empresa Xilinx desarrolló la FPGA XC4010 de 10,000 puertas y en 1999 fabricó la Virtex XCV1000 de un millón de puertas [117], lo que facilitó la ejecución de las etapas necesarias en el proceso de diseño, que son: la síntesis [122], la colocación y el enrutamiento (placement and routing o place and route, en inglés) [123-

127]. Automatizar estas etapas disminuyó en gran medida el esfuerzo a los diseñadores de HW. Además, la elección de LUTs como LC facilitó la interconexión dentro de las FPGAs, ya que una LUT se pudo considerar como una memoria y las memorias se colocaron de una forma más eficiente en el chip de silicio [117]. Los enlaces internos en las FPGAs contienen búferes, multiplexores de enrutamiento y celdas de memoria, por lo que gran parte del coste lo genera la interconexión [128]. Igualmente, la lógica basada en LUTs redujo la cantidad de interconexiones requeridas para implementar una función. Esta evolución hizo predecir que, en un corto periodo de tiempo, se podría disponer de FPGAs que cumplieran con los requisitos de un Circuito Integrado para Aplicaciones Específicas (Application-Specific Integrated Circuit (ASIC), en inglés) con la ventaja de que fuera de un coste mínimo [117].

En lo que se denominó la edad de la acumulación, los desarrolladores de FPGAs se enfocaron en cumplir con dos líneas de mercado, que las designaron “gama baja” y “gama alta”. La gama baja se centró en la eficiencia y el bajo coste, produciendo FPGAs con alto rendimiento pero menor capacidad, como las Spartan de Xilinx, las Cyclone de Altera y las EC/ECP de Lattice [117]. La gama alta se concentró en suministrar FPGAs de gran capacidad y producir librerías de *soft logic* IP para diversas funciones o hasta Microprocesadores (MPs) [129]. El objetivo principal en ambos casos fue reducir el área y el consumo de energía, además de aumentar el rendimiento, la sobrecarga de reconfiguración y facilitar el desarrollo en el diseño [130]. Como resultado se obtuvieron FPGAs con bloques dedicados para mejorar el rendimiento en lugar de contar con millones de puertas, incluyendo multiplicadores, BRAM, múltiples MPs, como Power-PC, bloques Manejadores Digitales de Reloj (Digital Clock Manager (DCM), en inglés), transceptores síncronos con una velocidad de gigahercios y cifrado de flujo de bits para proteger la IP del diseño, además del conjunto de LUTs, flip-flops, IOBs y las interconexiones o enrutamiento reconfigurable. En consecuencia, las herramientas de diseño e implementación también se optimizaron para cumplir con la gran variedad de objetivos y usos. Por otro lado, no todos los diseños requieren de ciertas funciones determinadas, como un MP Power-PC, memorias o multiplicadores, que aumentan el coste en el diseño de cualquier sistema, por lo que los fabricantes desarrollaron FPGAs específicas dependiendo de la industria o uso que se le fuera a dar. Por ejemplo, para la industria de las comunicaciones las FPGAs incorporaron transceptores de entradas/salidas (E/S) de alta velocidad, miles de multiplicadores dedicados de alto rendimiento, la capacidad de crear amplias rutas de datos y *pipelines* para conmutar grandes cantidades de datos sin sacrificar el rendimiento [117]. La desaceleración económica originó a los fabricantes de FPGAs a reducir de gran manera los costes

[131,132]. En contraste, la evolución de dispositivos para diferentes aplicaciones, generó el desarrollo de Sistemas en Chip (System-on-a-Chip (SoC), en inglés), que en las FPGAs son un SoC completo programable, que contienen memoria, MPs, interfaces analógicas, una Red en Chip (Network-on-Chip (NoC), en inglés) y un bloque lógico reconfigurable, como por ejemplo, la familia Zynq de Xilinx y SoC FPGA de Altera [117].

Las ventajas de coste, capacidad y velocidad que actualmente poseen las FPGAs se fueron obteniendo lo largo de su desarrollo, así como también, la forma de descripción les dio una mayor eficacia. Como se puede observar, en el desarrollo de las FPGAs los principales desafíos fueron minimizar el coste inicial y obtener una arquitectura con un diseño simple para obtener una reconfiguración instantánea, además de reducir los fallos, características que con el paso de los años siguen siendo una de las principales ventajas de las FPGAs [117]. Lo anterior ha generado la existencia de empresas fabricantes que ofertan de forma general tres opciones de FPGAs diferentes denominadas: gama baja, gama alta y SoC. La gama baja proporciona la solución más rentable (coste-efectividad) para una interfaz digital general. Esta gama solo contiene elementos básicos (lógica, memorias y relojes) y es muy utilizada para productos de consumo, creación de prototipos de bajo coste y educación. La gama alta proporciona potencia computacional de alta densidad e interfaces digitales de súper alta velocidad, mediante un elemento lógico especialmente diseñado como bloque de Procesamiento Digital de Señal (Digital Signal Processor (DSP), en inglés). Estas FPGAs están diseñadas para procesar una gran cantidad de datos digitales, utilizados en HPC, redes de transporte óptico y radio definida por SW.

Los SoC basados en FPGAs tienen integrados una CPU y muchas interfaces. Por ejemplo, el MCU ARM *Core* es muy común de encontrar en SoC basados en FPGAs. El objetivo de estos sistemas es cumplir con las necesidades de tener en el dispositivo la flexibilidad del SW y velocidad del HW [133]. Además, actualmente embebida en el CLB existe predefinido una cadena de propagación rápida de los acarreo (high-performance look-ahead carry, en inglés) que permite a las FPGAs implementar sumadores de muy alto rendimiento [29]. Actualmente existen diferentes modelos o familias de FPGAs lo que hace que estas varíen en coste [133]. En la Tabla 2.2 se muestran diferentes tipos de FPGAs dependiendo del fabricante. Para este trabajo se denota la importancia de las FPGAs de gama baja, de las cuales, la compañía Lattice Semiconductor se ha especializado en su fabricación y desarrollo.

Tabla 2.2 Diferentes tipos de FPGAs.

Fabricante	Gama Baja	Gama Alta	SoC
Lattice	iCE40 / MachXO	ECP / Lattice	
Xilinx	Spartan-6	Virtex	Zynq
Altera	Max 10	Stratix / Arria	Stratix SoC
Microsemi	IGLOO 2		Smart Fusion 2
Achronix		Speedster 22i	

El desarrollo de las nuevas funciones reconfigurables de las FPGAs está relacionado con la evolución de las nuevas generaciones y familias de las FPGAs, logrando reducir el coste y mejorando el rendimiento [29], entre las que se pueden enumerar las siguientes:

- Sumadores y contadores de alta velocidad.
- Memorias de Acceso Aleatorio (Random Access Memory (RAM), en inglés) distribuidas o LUTRAMs.
- BRAM
- Bloque de MP Computadora con Conjunto de Instrucciones Reducido (Reduced Instruction Set Computer (RISC), en inglés) basado en Power-PC
- Un SoC completo como bloque en la FPGA [29].

Hoy en día, el desarrollo de las FPGAs tiene una combinación de recursos que son escalables y flexibles que se enfocan a las necesidades de los nodos sensores inalámbricos como es el tamaño, el peso, el bajo consumo de potencia y el bajo coste, lo que hace factible su utilización en el desarrollo de dispositivos para el IoT. Aunque, existen FPGAs que se basan en la tecnología SRAM y requieren una memoria no volátil externa para mantener su patrón de configuración, su evolución ha permitido una disminución en el consumo de energía. Por ejemplo, la familia de FPGAs iCE40 de Lattice HX [134] puede funcionar por debajo de 10 mA en funcionamiento activo y sobre 35 μ A en funcionamiento inactivo. No obstante, en este tipo de FPGAs la intensidad, que consume de la tensión de alimentación en el instante del *power-on*, puede ser mayor que la del funcionamiento activo, debido al estado inicial desconocido de las celdas SRAM. Sin embargo, la familia de FPGAs iCE tiene un máximo de 1,2 mA de corriente de entrada, que es muy efectivo para aplicaciones alimentadas por batería [135].

En este contexto, es claro que las FPGAs son una alternativa de implementación muy adecuada para la próxima generación de nodos sensores inalámbricos para IoT, ya que brindan ventajas competitivas únicas, que son: flexibilidad (lógica de reconfiguración), versatilidad de los recursos de HW, escalabilidad (combinación de múltiples recursos lógicos con capacidades de reconfiguración), alto rendimiento (procesamiento paralelo),

bajo consumo y seguridad [135]. Por ejemplo, en [136] se menciona que las FPGAs de la familias ECP5 e iCE40 UltraPlus de Lattice satisfacen las necesidades de rendimiento y bajo consumo de potencia hasta para acelerar redes neuronales, esto en relación a que el consumo de potencia de las FPGAs ECP5 es menor a 1 W y que de las FPGAs iCE40 UltraPlus está en el rango de los mW.

2.3.2. Descripción general de la arquitectura de las FPGAs

El objetivo principal de las FPGAs es la implementación de lógica programable para crear nuevos dispositivos en HW. Las FPGAs se construyen alrededor de una serie de bloques lógicos programables integrados a través de un gran número de interconexiones configurables. A la matriz que forman las FPGAs se le denomina estructura lógica programable y en los márgenes contienen IOBs configurables o programables, diseñados para interconectar las señales de la estructura con el mundo exterior [29].

La Figura 2.8 muestra la arquitectura general de una FPGA. Esta consiste en una matriz de bloques lógicos programables interconectados con Puntos de Interconexión Programables (Programmable Interconnection Point (PIPs), en inglés) [117].

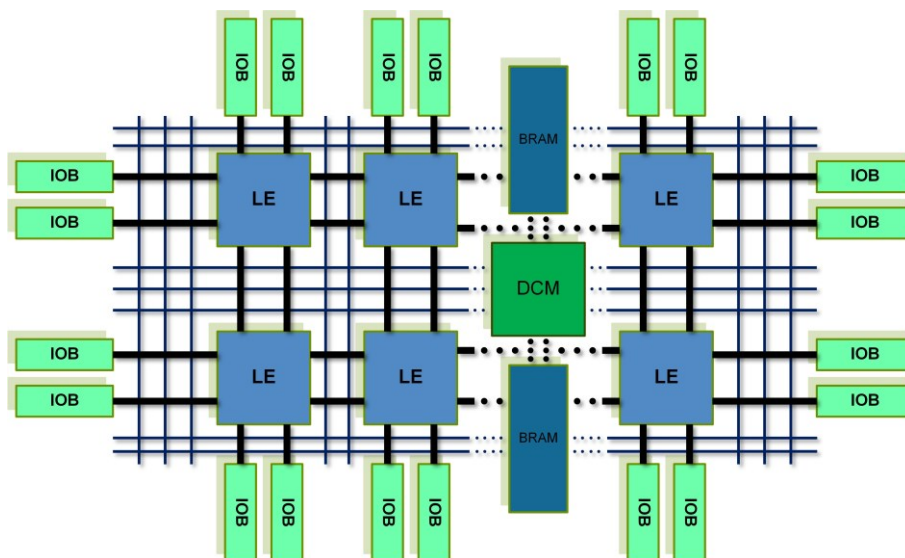


Figura 2.8 Arquitectura general de las FPGAs

Uno de los elementos básicos de una FPGA para la implementación de funciones lógicas es el Elemento Lógico (Logic Element (LE), en inglés) o conocido también como

CLB (en Xilinx), el cual está formado por una LUT, un flip-flop y un multiplexor [31], tal y como se muestra en la Figura 2.9. En [29] a la combinación de un LUT, un *carry chain* y un registro se le denomina LC. En la actualidad, cada LE implementa una o más funciones lógicas mediante una LUT configurable. Los LEs implementan desde las funciones más simples hasta las más complejas dentro de una FPGA [31].

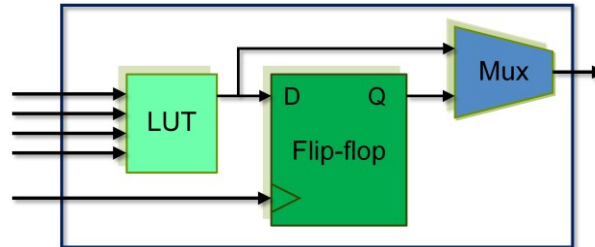


Figura 2.9 Elemento Lógico (LE) básico configurable.

Los diferentes fabricantes de FPGAs, tienen sus propios diseños de LEs y son muy similares al mostrado en la Figura 2.9. Las entradas a un LE están conectadas a la LUT y la salida de la LUT a un flip-flop. La salida del LE se selecciona mediante un multiplexor (MUX). La LUT y el MUX son los principales bloques configurables en el LE. Fundamentalmente, la LUT es una pequeña SRAM, generalmente de cuatro entradas y una salida, por lo que puede generar cualquier función booleana de cuatro entradas [31]. La evolución de las FPGAs ha permitido que en cada LUT exista un registro integrado asociado, y la cantidad de recursos de la lógica programable hace factible realizar diseños que mantienen velocidades altas de funcionamiento. Existen diferentes familias de FPGAs, y en cualquiera de ellas todas comparten una arquitectura común, independientemente de la cantidad de lógica programable que contenga [29].

Actualmente, dependiendo del fabricante de la FPGA, existen diferencias en el tamaño del empaquetado de los CIs, la velocidad de funcionamiento, el rango de temperatura soportado, los niveles de voltaje y la capacidad para soportar señales de E/S. Lo anterior, ayuda a minimizar los costes y el espacio en las PCBs al momento de diseñar algún sistema [29].

2.3.3. Tipos de tecnologías utilizadas

Los fabricantes se asisten de diferentes tecnologías para desarrollar las FPGAs, entre las que cabe destacar la tecnología SRAM por su rápida velocidad de acceso y su

reprogramación ilimitada y dinámica. La SRAM almacena la información de configuración (el estado de los puntos de interconexión) de la FPGA, pero al ser una memoria volátil se debe añadir al FPGA una FEPROM externa, que almacene el “fichero de bits”. Cuando se aplica la tensión de alimentación al sistema, la FPGA automáticamente transfiere el contenido de la FEPROM a su SRAM interna. Otras tecnologías son las Flash y las de Antifusible, que no son volátiles, por lo que pueden almacenar la configuración sin necesidad de alimentación de energía eléctrica. Por tanto, tiene la ventaja del “encendido instantáneo” sin necesidad de recargar el “fichero de bits” en la FPGA en el “Power On”, además de que consumen menos energía eléctrica [31].

Lattice Semiconductor es un proveedor líder en la evolución de FPGAs, y ha desarrollado FPGAs con tecnologías no volátil y basadas en Flash, de 90nm y 130nm. Mejoró significativamente el diseño de la PAL al hacer que los chips fueran programables electrónicamente. En 1983 desarrolló la Matriz Lógica Genérica (Generic Array Logic (GAL), en inglés) al combinar la tecnología EEPROM y CMOS para crear un dispositivo lógico de alta velocidad y bajo consumo de energía [118].

Los tres tipos de tecnologías de configuración en referencia a los fabricantes de FPGAs [31] se muestran en la Tabla 2.3, en la que se observa que Lattice Semiconductor tiene la ventaja de desarrollar FPGAs con ambas tecnologías SRAM y Flash.

Tabla 2.3 Diferentes tipos de tecnología de configuración.

Fabricante	SRAM	Flash	Antifusible
Achronix	√		
Altera	√		
Lattice	√	√	
Microsemi		√	√
Xilinx	√		

2.3.4. Comparación de FPGAs y otros dispositivos

De manera general se pueden comparar las FPGAs con otros dispositivos, tomando en cuenta los recursos para diseñar e implementar sistemas digitales. Los dispositivos se pueden clasificar en cuatro grupos: el elemento discreto, el ASIC, la FPGA y el MCU. Implementar un sistema digital utilizando elementos discretos tiene las desventajas de que el espacio físico puede ser limitado, las conexiones pueden resultar prohibitivas y el diseño será estático una vez realizado. La implementación realizada en ASICs hace que el diseño de la topología quede fijo y necesite bastante tiempo para su fabricación [28].

En contraste, la ventaja más importante de la FPGA es la reconfiguración del diseño, con la facilidad de hacerlo al modificar el código del HDL correspondiente, además de la facilidad de crear y verificar los prototipos. También, las FPGAs se consideran un entorno de diseño libre y flexible, lo que permite la disminución en el consumo de energía, en comparación con los MCU que disponen de un conjunto único de instrucciones para ejecutar una acción. Asimismo, la implementación paralela es inherente en las FPGAs, a diferencia de un MCU que es un dispositivo secuencial. Por lo tanto, en las FPGAs las operaciones se realizan más rápidamente y hasta en órdenes de magnitud más elevados en comparación con MCUs. Por esto y más, las FPGAs son una buena alternativa para implementar sistemas digitales [28]. Otra diferencia entre las FPGAs y las CPUs es la ventaja que brinda la FPGA al reconfigurar el HW, y en contraste la CPU tiene una configuración permanente para realizar tareas predefinidas “de forma consecutiva” y de acuerdo con sus propias funciones, a través de SW [31].

Las FPGAs tienen una cantidad determinada de elementos lógicos, para realizar funciones e interconexiones entre ellas, los cuales el diseñador controla, por lo que las tareas que realiza la FPGA no son predefinidas de fábrica. Las diferencias descritas en este apartado entre FPGAs y CPUs se resumen en la Tabla 2.4. Por ejemplo, para ejecutar una tarea específica dentro de la FPGA es necesario codificarla en un HDL que se ejecuta de forma concurrentemente. La capacidad que tienen las FPGAs de ejecutar procesamiento paralelo es una de las características principales y que las hace superiores en muchas aplicaciones a las CPUs [31].

Tabla 2.4 Comparación entre FPGAs y CPUs.

Característica	FPGAs	CPUs
Coste	Alto	Bajo
Estructura de HW	Flexible	Fija
Ejecución	Concurrente	Secuencial
Programación	HDL	Lenguaje ensamblador y de alto nivel
Tiempo de desarrollo	Largo	Corto
Consumo de energía	Eficiente	Menos eficiente

Las FPGAs también tienen diferencias con los ASICs, entre las que se pueden mencionar el rendimiento, la flexibilidad de diseño, el coste y las herramientas de descripción. A continuación se indican las ventajas de las FPGAs [118].

- Reconfiguración y rendimiento:

- ✓ Contienen HW especial: BRAM, módulos CLK, MACs Ethernet, memorias y E/S de alta velocidad, lo que permite construir hasta una CPU embebida, que se puede utilizar para obtener un mejor rendimiento.
- ✓ Las de gama alta contienen bucles de bloqueo de fase, Señal Diferencial de Bajo Voltaje (Low-Voltage Differential Signaling (LVDS), en inglés), bloques de multiplicador-acumulador (MAC) por HW para DSPs y núcleos de MP.
- ✓ Existen en la actualidad FPGAs con ADCs embebidos.
- Rápida reprogramación:
 - ✓ La capacidad de reprogramación permite que un nuevo diseño se pueda cargar de forma remota e instantáneamente.
- Mínimo coste:
 - ✓ No tienen costes de Ingeniería No Recurrente (Non-Recurring Engineering (NRE), en inglés) y herramientas de bajo coste o gratuitas.
 - ✓ La fabricación de desarrollos de sistemas por HW a pequeña escala genera un coste muy bajo.
- Mínimo tiempo para comercialización:
 - ✓ Una gran ventaja es la reutilización de la misma FPGA para verificar la implementación de diferentes diseños o prototipos.
 - ✓ Están “listas” para utilizarse, solo necesitan de una descripción en un HDL y cargar el “fichero de bits”.
 - ✓ Realizan una síntesis sencilla en comparación con los ASICs.

Las diferencias más importantes entre las FPGAs y los ASICs se muestran en la Tabla 2.5.

Tabla 2.5 Diferencias entre FPGAs y ASICs.

Característica	FPGA	ASIC
Coste inicial	Muy bajo	Caro
Alto coste de volumen	Caro	Más barato
Corrección de errores	Fácil	Difícil
Tiempo de comercialización	Bueno	Malo
Consumo de potencia	Más	Menos
Método de diseño	Fácil	Complicado

En resumen, la capacidad de reconfiguración y el soportar procesamiento concurrentemente permiten que en las FPGAs se desarrollen muchos y diferentes sistemas digitales. Por consiguiente, la funcionalidad depende del diseño de la estructura del HW para manejar la tarea deseada [31].

Del recuento realizado en este apartado, se puede deducir que el tiempo de procesamiento computacional necesario para ejecutar una tarea por HW en la FPGA puede ser mucho menor al necesario para realizar la misma tarea en MCUs basados en SW. En consecuencia, esto reduce significativamente el consumo de energía en la implementación de sistemas digitales.

2.3.5. ***Glue logic* e interfaz personalizada**

En electrónica digital es necesario el uso de elementos de acoplamiento para unir dos o más dispositivos. A estos dispositivos también se los conoce como interfaces, las cuales son capaces de controlar o comunicar unidades con diferentes protocolos de comunicación, que pueden ser un caso complejo para implementar circuitos lógicos.

Glue logic es una forma especial de describir un circuito digital que permite que diferentes tipos de chips o circuitos lógicos funcionen juntos actuando como una interfaz entre ellos. En otras palabras, estos sistemas unen múltiples módulos [137]. Al momento de surgir las FPGAs se consideraron convencionalmente como chips *glue logic* utilizados para conectar componentes para formar sistemas complejos [138,139]. La evolución y la creciente complejidad de las FPGAs permitió que ahora sean un componente muy importante para la solución y funcionamiento de sistemas complejos [138]. Diseñadores de Núcleos de Propiedad Intelectual (Intellectual Property (IP) Core, en inglés) utilizan diferentes protocolos e interfaces de comunicación (incluso propietarios), lo que complica su integración y reutilización debido a problemas de compatibilidad. Según Rodríguez Andina et al. en [140] indican que es necesario agregar *glue logic* a los diseños, lo que crea problemas relacionados con el rendimiento del IP Core y, a su vez, de los SoC. En contraparte mencionan que la lógica de control para ADC y DAC externos se puede implementar utilizando lógica distribuida dentro de las FPGAs, eliminando la necesidad de chips adicionales que implementen *glue logic* y retrasos asociados con las interconexiones externas que limitan la frecuencia de muestreo y causan problemas de sincronización, lo que tiene un impacto negativo en el rendimiento. Hacen énfasis en que, las FPGA se utilizan para facilitar la descripción e implementación de *glue logic* y crear de manera rápida prototipos, en virtud de la capacidad y la cantidad de recursos de HW con los que cuentan las FPGAs. Esto permite también la implementación de sistemas completos y complejos en un solo chip, lo que ha globalizado enormemente sus dominios de aplicación. Por todo lo anterior, vale la pena utilizar FPGAs para resolver el control de aplicaciones,

implementando *glue logic* y evitan la necesidad de llenar la PCB con muchos y diversos dispositivos discretos requeridos solo para propósitos de interconexión o interfaz.

Como se puede observar, las FPGAs se han ubicado como el componente central en los sistemas digitales para implementar *glue logic*. Las FPGAs se utilizan como interfaz entre CPUs y dispositivos con una variedad de E/S y también con otros dispositivos para mapeo de memoria. Por lo general, muchas aplicaciones requieren interfaces personalizadas que se pueden implementar completamente en las FPGAs, ya que existen con un coste muy bajo con la ventaja de tener internamente redes lógicas programables combinacionales y registros, además de bastantes puertos de E/S [29].

Las FPGAs otorgan la facilidad de desarrollar varias interfaces de diversos estándares y niveles de rendimiento, y conectarlas de forma conjunta en un sistema de control o conmutador también implementado en una FPGA. Asimismo, las FPGAs se pueden utilizar de forma ideal como una interfaz para recibir y procesar flujos de datos en tiempo real y comunicarse con algún otro dispositivo que requiera altos niveles de velocidad y entonces las CPUs controlen el flujo de datos a otro nivel de velocidad. En diseños digitales actuales el acceso a memoria es muy importante, para lo cual, las FPGAs tienen embebidas diferentes estructuras que permiten implementar memorias como registros discretos, registros de desplazamiento, RAM distribuida o BRAM, y al mismo tiempo permiten el acceso externo a memorias DDR [29].

2.4. Máquina de estados finita

Una ventaja clave de los FPGA es que el diseñador puede desarrollar sistemas que pueden ofrecer niveles muy altos de rendimiento gracias a las características específicas del HW, logrando mayor velocidad y funcionalidad de procesamiento, lo que permite mapear de manera eficiente las rutas de datos de circuitos de control, de controladores, de interfaces y de periféricos, generalmente descritos mediante FSMs [138]. Las FSMs son un poderoso elemento de diseño, que permiten implementar en HW un cierto algoritmo. Si un diseño lógico puede ser expresado como un conjunto de pasos lógicos determinísticos, el algoritmo puede ser descrito mediante una FSM. Las FSMs se pueden describir fácilmente, por lo que es una práctica de modelado muy común, cómoda y factible que permite al diseñador crear y definir un comportamiento específico y de forma personalizada [141,142]. Se utilizan para generar soluciones a sistemas digitales complejos [143]. Las FSMs permiten describir circuitos de control, ocupándose de generar señales de configuración y muestreo, siguiendo

las secuencias requeridas y cumpliendo los requisitos de tiempo especificados por el diseñador [140]

La descripción de la funcionalidad de una FSM es la representación del comportamiento del circuito. Un diagrama de estados es una forma gráfica dirigida de describir la funcionalidad de una FSM, en el que cada estado (o vértice) dentro del sistema se denota como un círculo y un identificador. Las transiciones entre los estados se indican mediante flechas con las condiciones de entrada que causan las transiciones. El estado en el que se encuentra la FSM se denomina “estado actual”, y el estado destino de una transición se denomina “estado siguiente” [144]. La Figura 2.10 muestra el diagrama de estados de una FSM.

Dependiendo de cómo se generen los valores de salida se distinguen dos tipos de FSMs. Una es cuando la salida solo depende del estado actual de la FSM, y se le denomina Moore. La otra depende tanto del estado actual como de las entradas actuales del sistema y se le denomina Mealy [141,144,145].

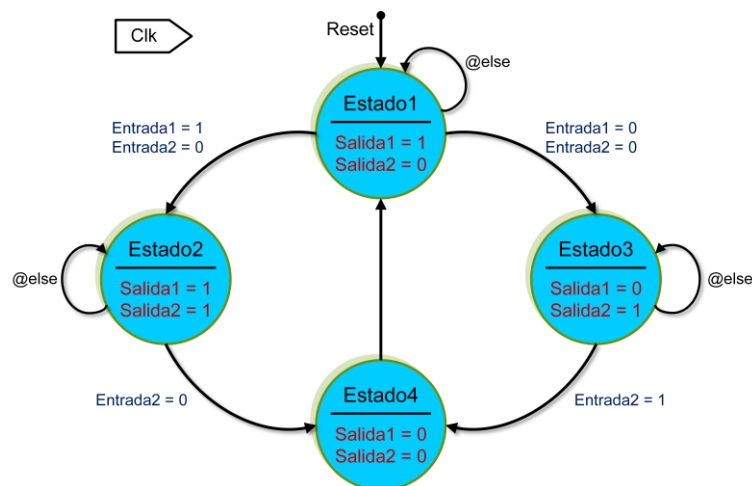


Figura 2.10 Ejemplo de un diagrama de estados de una FSM

En la Figura 2.11 se muestra el diagrama de bloques de una FSM, en el que se indican los tres bloques principales: la lógica del estado siguiente, el registro o memoria de estado y la lógica de salida. El bloque de lógica del estado siguiente es un sistema combinacional que genera la señal “estado siguiente”, la cual indica el estado a la que evolucionara la FSM, y cuyo valor se determina a partir del estado actual en el que se encuentra el sistema y del valor de las entradas. El registro de estado es el circuito que mantiene el estado actual de la FSM, que se actualiza con el estado siguiente cuando se produce un flanco de subida de la señal de reloj (CLK). El bloque de lógica de salida es un sistema combinacional que calcula

los valores de las salidas del sistema que siempre dependen del estado actual de la FSM, y según el tipo de FSM, de las entradas del sistema [141,144,145].

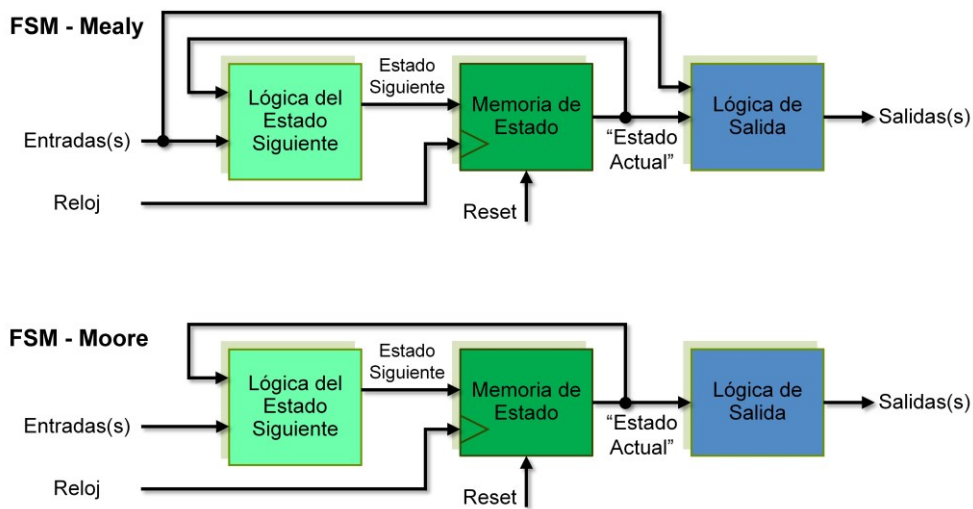


Figura 2.11 Componentes de una FSM.

Construir las funciones de un sistema digital requiere de la descripción del comportamiento de una FSM. Un diagrama de Máquina de Estados Algorítmica (Algorithmic State Machine (ASM), en inglés), también denominado Esquema-Gráfico de Algoritmos (Graph-Schemes of Algorithm (GSA), en inglés) se utiliza para representar una FSM, en el diagrama ASM las entradas se denotan como “condiciones lógicas” y las funciones de salida como “micro-operaciones” [145].

Un diagrama ASM es un grafo conectado y dirigido que tiene un conjunto finito de estados. Típicamente, tiene cuatro elementos diferentes denominados “inicio”, “fin”, “operador” y “condicional”. Un “operador” puede contener una colección de micro-operaciones, que se producen y ejecutan durante un ciclo de operación de la FSM, a la cual se le denomina FSM con ruta de datos (FSM with Data Path (FSMD), en inglés). Un “condicional” contiene un elemento de decisión con dos únicos valores booleanos diferentes (verdadero o falso) [145]. Un diagrama ASM tiene la ventaja, de que puede especificar al mismo tiempo el control y el flujo de la información [146]. En la Figura 2.12 se visualiza el ejemplo de un diagrama ASM.

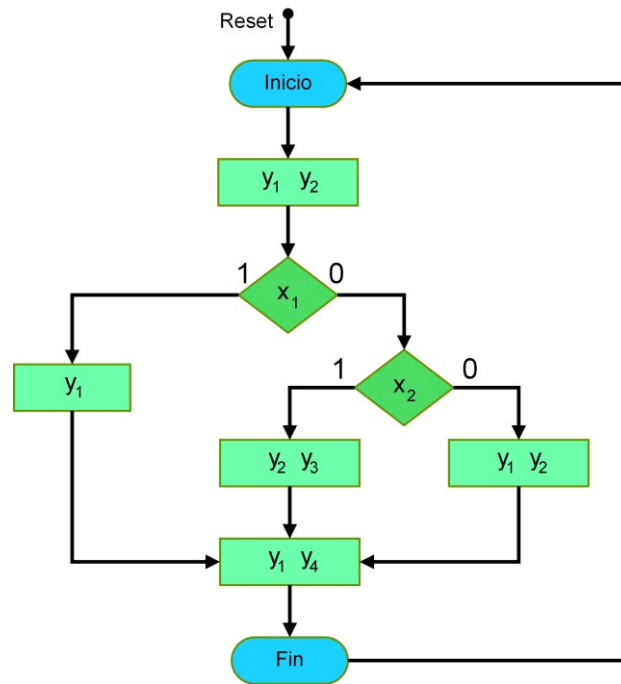


Figura 2.12 Diagrama de un ASM.

Una FSMD se constituye de una FSM y de un sistema secuencial regular. La FSM se encarga de la ruta de control, examinando el estado actual del sistema y el valor de las entradas o los comandos externos, a partir de lo cual, genera señales de control para especificar el funcionamiento del sistema secuencial regular, a lo que se le conocen como ruta de datos. Las FSMD se utilizan para implementar algoritmos representados por la metodología de transferencia de registros (register-transfer (RT), en inglés), donde las operaciones del sistema se describen de forma secuencial para la manipulación y la transferencia de los datos entre una colección de registros. La FSMD consta de dos tipos de sistemas secuenciales, ambos controlados por la misma señal de reloj, por lo tanto, es un sistema síncrono [147].

En base a estos métodos de descripción de HW, se realizó el diseño del *bridge* DALI para su implementación en la FPGA embebida en el nodo sensor inalámbrico utilizado en WSN para aplicaciones de alumbrado público.

2.5. Protocolo de comunicación en serie

Dentro de los requerimientos de los dispositivos embebidos, está la necesidad de interactuar e intercambiar datos entre sí y con otros dispositivos, por lo que la comunicación en serie es una forma de realizarlo. La comunicación entre el *bridge* DALI y el *host* del

nodo sensor inalámbrico se realizó utilizando el protocolo serie UART, el cual es un método de comunicación en serie que permite que dos componentes diferentes o dispositivos se comuniquen entre sí, sin la necesidad de tener una línea de reloj (CLK).

La comunicación en serie se utiliza para transferir un bit a la vez a través de un medio determinado. El protocolo de la comunicación en serie se puede observar en la Figura 2.13. Existe también la comunicación en paralelo que puede transferir una gran cantidad de datos a la vez, pero con el inconveniente de que cada bit requiere de su propia línea o canal separado de los demás para realizar la comunicación, lo que genera el disponer de más recursos. Es por esto que el método de comunicación en serie se utilizó en el dispositivo embebido del proyecto en cuestión, ya que, a diferencia de la comunicación en paralelo, solo requiere una línea para facilitar el intercambio de datos [148,149].

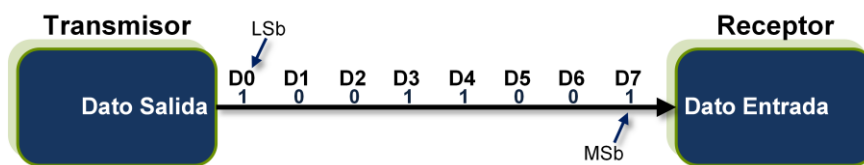


Figura 2.13 Protocolo de la comunicación en serie.

Otros métodos de comunicación en serie populares son el Estándar Recomendado 232 (Recommended Standard 232 (RS232), en inglés), el SPI, el Circuito Inter-Integrado (Inter-Integrated Circuit (I2C), en inglés). Gracias a los recientes avances en la tecnología, la comunicación en serie se está volviendo más barata, rápida y confiable [148,149].

UART es un protocolo asíncrono que no requiere de un CLK para sincronizar la comunicación entre los dispositivos que se lleva a cabo la transferencia de datos. Por lo tanto, los datos se transfieren sin la necesidad de una línea adicional para un CLK externo, de este modo solo se requieren las líneas de transmisión y recepción. Un paquete de datos o mensaje UART consta de cuatro componentes. En primer lugar, se tiene el bit de inicio, que indica el comienzo de la transmisión de los datos, suele ser un pulso bajo o cero lógico. A continuación, se transmite el mensaje real en formato de 8 bits, empezando por el bit menos significativo y finalizando con el bit más significativo. Después se transmite el bit de paridad que utilizara el receptor para comprobar errores. El bit de paridad solo se utiliza para verificar y validar si los datos son correctos o tienen algún error, y no para la corrección real. El último bit del mensaje es el bit de paro, que tiene como propósito indicar que ha finalizado la transmisión, generalmente con un pulso alto o uno lógico. El formato no está limitado a un solo bit de paro, esto dependerá la configuración utilizada por el desarrollador

del dispositivo [148,149]. El formato del mensaje de la transmisión asíncrona se muestra en la Figura 2.14.



Figura 2.14 Formato de la transmisión asíncrona.

Si bien la comunicación con la UART no requiere un CLK de alguna forma debe existir una velocidad de transmisión para el entendimiento entre los componentes, que se comunican mediante este método, lo que genera el concepto de velocidad de baudios, que especifica la velocidad a la que se transfieren los datos entre dispositivos, o más apropiadamente, la cantidad de bits transferidos por segundo, lo cual es necesario por no existir una línea de CLK. Una consideración importante es que solo una velocidad de baudios debe ser establecida para ambos componentes durante todo el proceso de intercambio de datos, por lo cual es importante identificar la velocidad de baudios del dispositivo destino. Las velocidades de baudios comunes son 1200, 2400, 4800, 9600, 38400, 19200, 57600 y 115200 [148,149].

En el capítulo 4 se explicará el análisis que se realizó para la descripción de los componentes necesarios para el diseño del *bridge* DALI y su implementación en la FPGA embebida en el nodo sensor inalámbrico utilizado en WSNs para aplicaciones de alumbrado público.

Capítulo 3. Trabajos relacionados

En este capítulo se presenta una recapitulación de trabajos relacionados con el desarrollo de nodos de sensores inalámbricos en sistemas de alumbrado público y nodos de sensores que utilizan FPGA. En especial se describirán brevemente aquellos trabajos que incluyan la implementación del protocolo DALI o que utilicen una FPGA como parte esencial del nodo inalámbrico. Los trabajos se han clasificado en tres partes dada la extensión del capítulo. Esta clasificación no es excluyente en el sentido de que algunos trabajos podrían estar en más de un apartado, pero se ha incluido en el que se considera que realiza una mayor aportación o es más interesante. Como se ha comentado a lo largo del capítulo de Introducción, hasta hace poco tiempo, no se utilizaban FPGAs en nodos de sensores inalámbricos por lo que el número de trabajos publicados es pequeño en comparación con el uso de FPGAs en otros tipos de sistemas electrónicos.

3.1. Trabajos que incluyen nodos inalámbricos para sistemas de alumbrado público

La innovación de los sistemas de SSL, históricamente se han soportado en las WSN y por consecuencia en los nodos sensores inalámbricos. En los trabajos que se relacionarán en este apartado cabe destacar el elevado procesamiento computacional que realiza el MCU de los nodos, por lo que nos planeamos tener como objetivo el reducir esta carga computacional en el trabajo de tesis. Un proyecto donde incorporan nodos de sensores inalámbricos en las farolas se presentó en [150], con el fin de monitorearlas y controlarlas en forma inalámbrica, donde la WSN soporta la transmisión de datos entre las farolas y el centro de control. Con el objetivo de tener un consumo eficiente de la energía, en [151] se presentó el control para un sistema de SSL donde cada farola está equipada con dispositivos inalámbricos, y mediante un proceso dinámico de sensado de las ubicaciones de peatones,

se logra encenderlas y apagarlas. Como se puede observar el control de los sistemas de SSL se enfoca en métodos que determina el movimiento de personas, el tráfico de automóviles, el estado del tiempo y la condición ambiental de un lugar.

En el estudio realizado se observa que el desarrollo acelerado que tiene actualmente la tecnología de las WSNs, permite realizar la adquisición y el tratamiento de datos, con la ventaja de la autoconfiguración y la facilidad de despliegue. Esto permite que los nodos sensores inalámbricos realicen funciones de emisor – receptor, además de ofrecer servicios de enrutamiento entre nodos y el intercambio de datos relativos a los sensores de cada nodo. Por ejemplo, en [152] se enfocan en el desarrollo de algoritmos de enrutamiento de mensajes en la red de iluminación, tolerante a fallos para que los mensajes alcancen siempre su destino.

En [153] se presenta un sistema de alumbrado público inalámbrico que utiliza dispositivos basados en ZigBee y alimentados por paneles solares. La WSN recoge la información relevante asociada a la gestión y mantenimiento del sistema, especialmente el estado de las farolas, con el fin de garantizar el control óptimo del mismo y tomar las medidas oportunas en caso de avería. El sistema consta de un grupo de estaciones independientes de medición (en cada farola) y una estación base, con las que se mide la intensidad de la luz del día para activar o apagar las lámparas. La estación de medición tiene sensores de presencia, de insolación y de averías.

En [33] se propuso sistema inalámbrico para el control de alumbrado público donde la información se transmite punto a punto mediante emisores y receptores ZigBee que forman una WSN para comprobar el estado de las luminarias y prevenir fallas. Este sistema tiene como objetivo que el funcionamiento de cada luminaria sea independiente. La información sensada en cada nodo se procesa antes de ser enviada, para encender y apagar las luminarias solo cuando sea necesario, y posteriormente se envía a una estación base que controla todas las luminarias que forman la red.

En [154] se presentó el desarrollo preliminar de un sistema de SSL focalizado en el medio ambiente de Malasia. El sistema utiliza Waspote (Libelium) como un nodo sensor, una puerta de enlace Meshlium y un ordenador como servidor, con el fin de integrarse todo en una WSN. El nodo monitorea información del estado de funcionamiento, la cantidad de uso de energía y el tiempo de atenuación de la luminaria, utilizando la técnica Modulación por Ancho de Pulso (Pulse Width Modulation (PWM), en inglés). El prototipo del sistema se realizó en un laboratorio con cinco nodos Waspote, con sensores infrarrojos para detectar movimiento y sensores resistivos para detectar el nivel de iluminación.

Un sistema para controlar el encendido y apagado de la iluminación se presenta en [155], en el cual se evalúan en forma grupal los datos de las farolas y se transmiten sobre una WSN en topología árbol. El sistema también tiene la capacidad de monitorizar el estado y comunicar los fallos o alertar sobre el mantenimiento en las farolas. En [11] se propone un sistema de SSL con base a una WSN, en el cual se agrega a cada farola un nodo sensor inalámbrico con el fin de monitorizar el tráfico vehicular. Los datos obtenidos por los sensores inalámbricos se comparten con los sensores adjuntos para controlar la iluminación.

En [156] presentan el prototipo de un sistema de control de iluminación denominado “E + grid”, con el fin de adaptar la iluminación a condiciones ambientales reales, a las necesidades del usuario y en relación con el tráfico. El sistema modula la temporalización del encendido y apagado, además de los niveles de atenuación de luminarias led, utilizando sensores de movimiento infrarrojos, colocados en cada luminaria, que miden la velocidad y establecen la dirección del movimiento, para compartir mediante comunicación inalámbrica con sus vecinos el escenario de tráfico detectado.

En [157] a través de nodos sensores inalámbricos que forman WSN, recopilan información del tráfico de vehículos y la transmiten a un centro de control para optimizar la iluminación mediante atenuación.

En específico para la administración de lámparas led en los sistemas de control de iluminación, se han diseñado nodos inalámbricos en [33,158] con el objetivo de reducir el consumo de potencia y los costes generados por el consumo de luz mediante la atenuación de la intensidad de la iluminación y la comprobación el estado de las lámparas para realizar el mantenimiento oportuno en caso de fallos. En estos casos, para lo comunicación inalámbrica los nodos utilizan el módulo comercial XBee de RF Digi MaxStream. Por otro lado, en [159] se intenta mejorar el módulo Wasmote a través de la conexión de sensores y añadiendo un sistema de control para lámparas led, con el fin de solucionar problemas de iluminación, además de minimizar el coste de inversión en comparación con los sistemas cableados.

En [6] se propone un esquema adaptativo para iluminación en base a detección de tráfico de personas y vehículos mediante sensores, el algoritmo se validó utilizando las herramientas de simulación SUMO y OMNeT++.

Las plataformas de evaluación basadas en Raspberry-Pi se utilizan habitualmente para controlar iluminación, como por ejemplo en [160], donde controlan farolas de una zona aislada. Se utiliza una red ZigBee en la comunicación entre las farolas y una comunicación

Worldwide Interoperability for Microwave Access (WiMAX) para transmitir información a un centro de control remoto.

En [158], se presenta un sistema inalámbrico para el control y monitorización de alumbrado público basado en ZigBee™ donde se programa el 8-bit AVR MCU para controlar todas las interfaces inalámbricas de transmisión y recepción de datos.

Otro sistema inalámbrico de alumbrado público reportado en [161] intenta soportar aplicaciones de IoT para transporte inteligente. Se utilizó un conjunto de sensores para la medición de las condiciones ambientales (iluminación, lluvia, temperatura y humedad), y cámaras de video para la detección de vehículos, donde se realiza un controlador por SW con el fin de establecer comunicación con la puerta de enlace para transmitir y recibir tramas, además de procesar la información, utilizando el transceptor CC1125 y el MCU de la familiar MPS430 (MSP430F5438A), ambos de Texas Instruments (TI). El controlador de comunicaciones se desarrolló en el lenguaje de programación “C”. Al ser realizado por SW, tiene el inconveniente que en la programación de la máquina de estados se necesitan retardos (delay, en inglés) necesarios para la sincronización adecuada en el envío de las tramas. Del análisis de los mecanismos que emplean para las interfaces de comunicación en los sistemas de iluminación mencionados, estas interfaces son dependientes de una colección de librerías, código fuente, diferentes herramientas y aplicaciones determinadas por el dispositivo donde se ejecuta el programa, lo que limita su funcionamiento.

3.2. Trabajos relacionados con sistemas de iluminación inteligente que utilizan DALI

En este apartado se describen sistemas, tanto cableados como inalámbricos, que utilizan DALI como control de las luminarias. Un ejemplo de sistema cableado se presenta en [111], donde además, se analizan varios métodos de control de iluminación, entre los cuales están el control de Modulación por Ancho de Pulsos (Pulse Width Modulation (PWM), en inglés) y el protocolo DALI. Determinando como resultado que el protocolo DALI es el más conveniente para desarrollar un sistema de control de iluminación. Como caso de análisis, controlan la iluminación mediante un Controlador Lógico Programable (Programmable Logic Controller (PLC), en inglés), utilizando un DALI *soft-gateway* para realizar la conversión del protocolo Modbus del PLC al protocolo DALI, para el control de lámparas led.

En una serie de trabajos basados en comunicación inalámbrica de tipo ZigBee se utiliza DALI para controlar las luminarias. Por ejemplo en [162], se presenta un sistema para administrar el alumbrado público por medio de WSNs y utilizando el protocolo de control de iluminación para balastos DALI. Para el enlace inalámbrico se utilizó el módulo Digi XBee-Pro 868 RF con una placa de desarrollo del MCU Arduino y para el control de las luminarias se utilizaron balastos de lámpara HID con interfaz DALI, lo que permitió la comunicación half-dúplex para conocer algunos parámetros de las luminarias. Los balastos utilizados fueron OSRAM Powertronic PTo DALI 70/200–240 3DIM con Philips SON 70W/220 I E27 1CT. También, en [163] se propone un módulo inteligente simple para ser conectado en el sistema de alumbrado público para medir y controlar los parámetros de la lámpara. El módulo se instala en la lámpara para recopilar los datos relativos a la red eléctrica y a los parámetros propios de la luminaria, utilizando el MCU Texas ARM TM4C123GH6PM. La comunicación inalámbrica está basada en 802.15.4/Zigbee, y para la comunicación entre el módulo y la luminaria utilizan un transceptor DALI. Otro caso se presenta en [164], donde un ordenador personal (Personal Computer (PC), en inglés) envía y recibe comandos para controlar digitalmente lámparas ultravioleta utilizando un módulo inalámbrico ZigBee y una MCU dsPIC (digital signal (ds) Programmable Integrated Circuit (PIC), en inglés) como *bridge* para el protocolo DALI; el sistema tiene el inconveniente de que requiere estar conectado a un ordenador personal.

Varios trabajos publicados optan por utilizar sistemas basados en plataformas que utilizan Linux. Por ejemplo, en el prototipo mostrado en [39], utilizan una plataforma de desarrollo BeagleBone Black, que contiene un MCU ARM, para atenuar la iluminación. La aplicación que se ejecuta en el MCU invierte los valores de las señales de los optoacopladores y realiza la codificación / decodificación Manchester, con la desventaja de que el prototipo del sistema utiliza diferentes componentes que no están integrados dentro de la misma placa de desarrollo. Un prototipo similar se describe en [165], donde controlan la iluminación utilizando una plataforma de desarrollo BeagleBone Black (MCU ARM Cortex-A8), la cual requiere ejecutar un sistema operativo basado en Unix y utilizar extensas librerías para su funcionamiento. En este caso, el SW de control del protocolo DALI se ha implementado como un módulo controlador, cargado en el kernel Linux del MCU. En [41] presentan un prototipo donde utilizan el módulo *kernel* de *Linux* sobre *UBUNTU 18.04.2 LTS* en la plataforma de desarrollo Raspberry Pi3 para controlar la interfaz DALI. Las operaciones de procesamiento y comunicación relacionadas con DALI se realizan por SW (en lenguajes “C” y Python). Sin embargo, tienen la necesidad de utilizar

el “PREEMPT_RT real-time patch” para mitigar problemas de tiempo real en cuanto a la precisión del tiempo de bit requerido por DALI y los tiempos de bajo y alto antes y después de las transiciones para realizar correctamente la codificación Manchester. Además, otra desventaja es que el sistema que proponen utiliza un Driver DALI comercial y una placa para el aislamiento electrónico entre la Raspberry Pi3 y la interfaz DALI que por ende no están integrados dentro del mismo nodo, lo que aumenta los elementos, tamaño y el coste, haciéndolo poco práctico para su utilización de forma individual en farolas.

Por otro lado, existen fabricantes que han implementado controles de iluminación tipo DALI, por ejemplo, el módulo de evaluación TPS62260LED-338 de Texas Instruments [166] el cual tiene el MCU MSP430F2131 que se utiliza para realizar la comunicación. El módulo también cuenta con un dispositivo adaptador para ajustar los niveles de los voltajes entre el MCU y la interfaz DALI. Por la forma en que está desarrollado, las funciones implementadas por SW tienen dependencia de recursos de los *temporizadores* (timers, en inglés) hardware de la MCU para tratar de cumplir adecuadamente con el funcionamiento de DALI. De igual forma, existen compañías que han desarrollado módulos comerciales como el DALI Interface RS232 PS/S [167], aunque por la incompatibilidad de interfaces y su tamaño físico, no existe posibilidad de integrarlo dentro de un nodo sensor inalámbrico o de alguna plataforma de desarrollo.

3.3. Trabajos que describen nodos inalámbricos que contienen FPGAs

En lo referente a las FPGAs, que es la propuesta de investigación del presente trabajo de tesis, tienen ventajas inherentes como flexibilidad y capacidad para ejecución en paralelo, lo que permite el procesamiento en tiempo real y computación de Alto Rendimiento. También, tienen características propias que permiten disminuir la latencia y el consumo de energía estática y dinámica. Los algoritmos de control de iluminación urbana que en ocasiones utilizan redes neuronales, ocupan una gran cantidad de cálculos computacionales, lo que hace difícil su implementación en un dispositivo electrónico de bajo coste, de bajo consumo de energía y con un número reducido de recursos HW, requisitos indispensables de un nodo sensor inalámbrico. Algunos trabajos que aprovechan estas características de las FPGAs se presentan a continuación.

En el estudio realizado en [168] exponen que las FPGAs de bajo consumo de potencia mejoran el cálculo de varios tipos de algoritmos en términos de velocidad y consumo de energía, en comparación con las MCUs de los nodos de sensores comerciales. Al realizar por HW las diferentes tareas, se puede generar un mejor equilibrio entre el rendimiento y el consumo de energía, como sucede, por ejemplo, en los protocolos y algoritmos de comunicación que tradicionalmente se programan por SW y que consumen mucho tiempo y batería, en comparación con la implementación por HW. Sobre el mismo enfoque, en [169-172] se menciona que el HW reconfigurable proporciona mejoras de rendimiento, flexibilidad, escalabilidad, seguridad y la disminución del consumo de energía, haciendo que esta tecnología sea excelente para abordar los desafíos en el diseño de nodos sensores inalámbricos para IoT.

En [173] se presenta una infraestructura de WSN que pueda soportar nodos sensores con reconfiguraciones remotas del HW, con el objetivo de actualizar las necesidades de las aplicaciones. La infraestructura de WSN se prueba en una aplicación de monitoreo ambiental para prevenir incendios forestales en tiempo real. También se diseña un nodo con capacidad para soportar reprogramación de SW y reconfiguración del HW denominado “RH-mote”, con una FPGA como dispositivo principal. Para la reconfiguración utilizan la herramienta de diseño “Xilinx PlanAhead”. La FPGA integra un procesador *soft core*, además de los componentes para el control de las interfaces con los sensores, con el conector JTAG y con la memoria Flash.

En [21], realizan una optimización de los recursos lógicos de la FPGA para mapear modelos neuronales, disminuyendo la latencia en los nodos de IoT. En [2], utilizan una FPGA para lograr altos niveles de seguridad y bajo consumo de potencia en protocolos de comunicaciones de WSNs. Justifican la necesidad de utilizar una FPGA debido a que los procesos de seguridad requieren una gran cantidad de procesamiento, difícil de asumir por CPUs de bajo coste. Acelerando las operaciones complejas de criptografía, liberan al MCU de estas operaciones y permite mejorar el rendimiento de los dispositivos IoT. Estas ventajas son difíciles de obtener si las operaciones se realizan con “software libraries”.

En [174] se desarrolló el estándar IEEE 802.15.4 implementado en un nodo sensor inalámbrico para aplicaciones de IoT, que incluyen en el mismo nodo una FPGA y un MCU. El fin es aumentar el rendimiento del sistema, descargando al MCU de diversas tareas que se realizan por SW y utilizando la FPGA como solución de HW/SW *Codesign* para nodos de WSN.

En [175] efectúan un paradigma basado en componentes personalizados de *soft core* implementados en una FPGA. Proponen esta solución para el diseño de un nodo inalámbrico que deba tener en cuenta necesidades críticas, como el consumo de energía, la tolerancia a fallos y la seguridad. En [176] se propone el diseño e implementación en una FPGA de un procesador *soft core* con el objetivo de ahorrar energía en el nodo sensor inalámbrico, reduciendo la carga de trabajo de la CPU del nodo.

Otros trabajos utilizan FPGAs en nodos sensores inalámbricos para aplicaciones específicas. Por ejemplo, en [177] se diseña e implementa, con la ayuda de una FPGA, un nodo sensor inalámbrico con algunos coprocesadores integrados personalizados para hacer que las operaciones de transferencia de datos e intercambio de claves sean seguras y rápidas. La implementación en HW de estos algoritmos hace que sus tiempos de ejecución sean más cortos, lo que permite reducir significativamente el tiempo necesario para el intercambio de claves y acelerar el cifrado de datos, logrando una buena compensación entre el uso de recursos de la FPGA, el bajo consumo de energía y el rendimiento.

Un trabajo donde se realiza la estimación de ubicación de personas utilizando FPGAs se presenta en [178]. Se busca minimizar el error en la distancia estimada entre un nodo fijo y un nodo móvil, en función del indicador de intensidad de la señal recibida. Se utiliza una red neuronal artificial implementada en una FPGA Virtex7. Destacan como conclusión que las FPGAs son una excelente tecnología para implementar en HW la red neuronal artificial.

En varios trabajos, se utiliza una FPGA para procesamiento de sonido y transmisión de datos de voz, por la facilidad de procesamiento en tiempo real y posibilidad de reconfiguración. Por ejemplo, en [179] se describe un nodo sensor inalámbrico reconfigurable para la transmisión de datos de voz. En [180] se presenta un sistema de reconocimiento de silbidos de pájaros mediante algoritmos implementados en una FPGA Spartan 3E y el *MicroBlaze soft processor*, para aplicaciones de WSN. Los resultados revelan que la implementación por HW produce una mejora significativa en comparación con lo realizado por SW. Concluyen que el reconocimiento de silbidos de los pájaros se pudo realizar en el nodo del sensor, gracias a la alta flexibilidad y al alto poder computacional que permitió realizar HW/SW *Codesign* en la FPGA.

En [181] se propone una arquitectura SoC basada en bloques IP para construir un nodo de sensor inalámbrico en una placa de desarrollo con un FPGA Altera DE2-70, y medir temperatura por medio de un termómetro digital. El nodo tiene tres módulos: el de adquisición de datos, el de procesamiento de datos y el de transmisión de datos. Del mismo modo en [182] se presenta el diseño de un nodo sensor de visión basado en una FPGA y

una MCU, para obtener reconfiguración dinámica y capacidad de procesamiento de imágenes.

Por último, es importante hacer mención que existen antecedentes en el grupo de investigación IEI, desde hace tiempo, sobre el desarrollo de sistemas de adquisición de datos para uso en sistemas inalámbricos que utilizan dispositivos lógicos programables de bajo consumo de potencia y una MCU, como por ejemplo en [183].

Capítulo 4. Arquitectura del *bridge* DALI

En este capítulo se ilustra la descripción del protocolo DALI, el diseño del *bridge* y de la interfaz DALI. Se tomaron en cuenta las funcionalidades y características del protocolo DALI para el control de luminarias. También, se analizó la comunicación entre el *host*, el *bridge* DALI y la interfaz DALI, con el objetivo de que la arquitectura del *bridge* DALI se realice mediante una descripción eficiente y portable de forma que se pudiese implementar en cualquier FPGA.

4.1. Protocolo DALI

La interfaz DALI utiliza un bus serie para la transmisión y recepción de datos con una estructura Maestro – Esclavo y necesita una fuente de alimentación de energía eléctrica con unas características concretas que se verá posteriormente. El maestro transmite 16 bits y el esclavo (luminaria) 8 bits. Es factible el uso de varios maestros, y una luminaria esclavo puede ser compartida por dos maestros. Para conectar luminarias a una línea de interfaz existen 64 canales o direcciones individuales [24].

El control digital de luminarias ha aumentado y con ello la necesidad de dispositivos que se “comuniquen” entre sí [25]. Una de las ventajas del protocolo DALI es que no necesita de circuitos para el control de la alimentación de energía eléctrica, en comparación con los sistemas analógicos, además de tener la capacidad de realizar direccionamiento individual de la luminaria a través de conmutación digital. Admite varias formas de conexión combinando dos o más topologías entre las de estrella, multipunto de bus con un solo maestro y *Daisy chain* [24].

La transmisión DALI utiliza la codificación Manchester. La codificación Manchester es un formato de señal digital donde el símbolo ‘1’ es representado por una transición

ascendente (estado actual “Bajo” – estado siguiente “Alto”) y el ‘0’ se representa por una transición descendente (estado actual “Alto” – estado siguiente “Bajo”). En la Figura 4.1 se puede observar el cambio de estado de la señal con respecto a los valores lógicos.

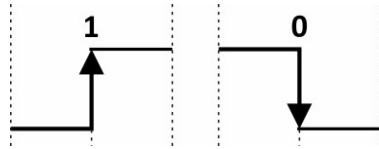


Figura 4.1 Cambio de estado de la codificación Manchester.

El paquete enviado por el dispositivo de control al receptor se denomina trama “Forward” y la respuesta, trama “Backward” [24]. En la Figura 4.2 se muestra que la trama “Forward” es de 19 bits en el siguiente orden: 1 bit de “Inicio” (I), 8 bits para la dirección, 8 bits para el comando DALI y 2 bits de “Paro”. Mientras tanto, la trama “Backward” es de 11 bits en el siguiente orden: 1 bit de “Inicio” (I), 8 bits de datos y 2 bits de “Paro”. En ambos casos se envía primero el bit más significativo.



Figura 4.2 Formato de la trama DALI.

En la transmisión DALI, el periodo de tiempo de ambos pulsos (alto y bajo) es la mitad del período de bit. El tiempo de un bit es de 833,33 μ s. La velocidad de transferencia es de 1200 bps con un error aceptable de $\pm 10\%$ [24]. En la Figura 4.3 se muestra una trama DALI en la que se indican los parámetros temporales relativos al periodo de bit y las transiciones de la señal con respecto a los valores lógicos.

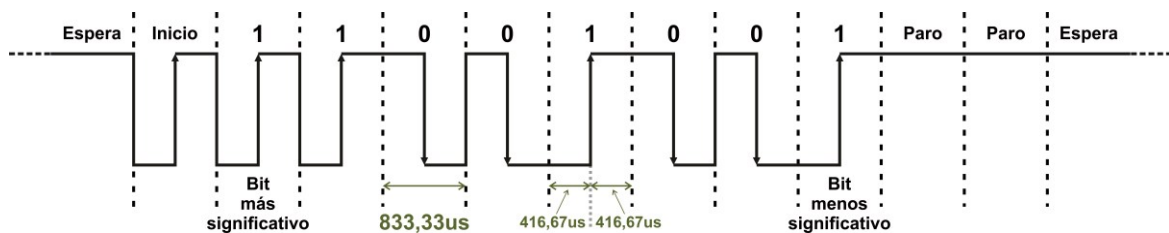


Figura 4.3 Transmisión DALI.

En relación con el tiempo de transmisión, el tiempo de medio bit es 416,67 μ s, permitiendo un rango de variación entre un máximo de 458,33 μ s y un mínimo de 374,99 μ s, del pulso a nivel alto y bajo. Una trama “Forward” tiene una duración de 38 medios bits, equivalente a 15,83 ms. Una trama “Backward” tiene una duración de 22 medios bits o 9,17 ms. El tiempo entre dos tramas “Forward” consecutivas es de al menos 22 medios bits o 9,17 ms. El tiempo entre una trama “Forward” y una trama “Backward” es mayor o igual a 7 tiempos de medio bit o 2,92 ms, y menor o igual a 22 tiempos de medio bit o 9,17 ms. El tiempo entre una trama “Backward” y una trama “Forward” consecutivas es de al menos 22 medios bits o 9,17 ms [24].

Por todo lo indicado anteriormente, ni la codificación/decodificación Manchester ni el formato de las tramas DALI pueden ser implementados mediante un puerto serie estándar de un MCU. Para poder implementar el *bridge* DALI se diseñó una interfaz específica que cumple las características anteriormente indicadas. La interfaz se podría implementar por SW, utilizando un puerto de Entrada/Salida de Propósito General (General Purpose Input/Output (GPIO), en inglés) y los *timers* necesarios para temporizar todos los tiempos necesarios para la transmisión y recepción de las tramas, pero, tal y como se indicó en la introducción, puede ocurrir que el MCU no tenga la suficiente capacidad de procesamiento ni recursos para garantizar el cumplimiento de la temporización de bit, recibir y transmitir las tramas, y realizar el resto de las funciones de un nodo sensor inalámbrico de una WSN. Por eso, surgió la necesidad de desarrollar un *bridge* que implemente el protocolo DALI y que controle la luminaria según los comandos que le envíe el MCU del nodo sensor inalámbrico de la WSN.

4.2. Diseño y descripción del *bridge* DALI

A continuación, se resaltan los aspectos más relevantes para realizar las descripciones de los componentes necesarios para diseñar el *bridge* DALI requerido por el nodo sensor inalámbrico para la interconexión entre el MCU y el conector de la luminaria (interfaz DALI). Además, se indican las características de la descripción para realizar la comunicación entre el *bridge* y el *host* del nodo sensor inalámbrico. Inicialmente se describen los objetivos del diseño, tomando en cuenta las consideraciones del HW y las funcionalidades del protocolo DALI para el control de luminarias, y obtener una descripción portable del *bridge* DALI para la implementación en FPGAs.

El objetivo del *bridge* DALI es recibir, interpretar y transmitir comandos DALI e información entre diferentes dispositivos. El *bridge* DALI tiene el propósito de decodificar y codificar los datos enviados entre el *host* y la interfaz DALI para el control de luminarias. La Figura 4.4, muestra el diagrama de bloques de la conexión del *bridge* DALI.



Figura 4.4 Diagrama de bloques de la conexión del *bridge* DALI.

El diseño del *bridge* DALI se basó en las siguientes especificaciones:

- Implementación en HW para su ejecución en tiempo real.
- Descripción portable desarrollada en VHDL-93, por lo que puede ser sintetizada con cualquier herramienta compatible con el estándar IEEE 1076-93. Por tanto, se podrá implementar en una FPGA de cualquier fabricante.
- Implementación realizada utilizando sólo las LUTs de la FPGA.
- Integración en FPGAs de bajo coste, bajo consumo de energía y con un número reducido de LUTs.
- Flexibilidad para soportar actualizaciones y reconfiguración.
- Desarrollo de una interfaz de comunicaciones tipo UART, para su fácil conexión a cualquier MCU, o como periférico de un nodo sensor inalámbrico.

Es importante destacar que también se tomaron en cuenta los aspectos necesarios para utilizar el *bridge* DALI en WSNs, consiguiendo una arquitectura integral y autónoma, con poder de escalabilidad y adaptable, lo que permite la comunicación directa con dispositivos como MCUs o plataformas de evaluación, sin depender de complementos externos. En resumen, el *bridge* DALI se encarga directamente de todas las comunicaciones, lo que libera al MCU de trabajo computacional, evitando la espera de eventos e interrupciones. Además, la descripción del *bridge* DALI es eficiente en el uso de los componentes lógicos de la FPGA, ya que deja libres componentes suficientes para implementar otros algoritmos de procesamiento adicionales que se consideren oportunos.

El *bridge* DALI consta de tres componentes principales: la UART, el procesador DALI y el Transmisor / Receptor Código Manchester (TRCM). El diagrama de bloques del *bridge* DALI se muestra en la Figura 4.5. El funcionamiento de cada componente se describió

mediante una FSM con Ruta de Datos Embebida (FSM with Embedded Data Path (FSMD), en inglés). El *bridge* DALI se ha diseñado como un sistema completamente síncrono, de forma que todos los componentes utilizan la misma señal de reloj y, a su vez, los modos de funcionamiento síncronos. Únicamente son asíncronas las entradas de inicialización (preset o clear, en inglés) de los flip-flops para inicializarlos al activarse la entrada de *reset* del *bridge* DALI. Los aspectos mencionados garantizan el funcionamiento a una frecuencia de operación mayor, ya que permite reducir los problemas de desviación (skew, en inglés) de la señal de reloj, y facilita el *place and route* del diseño en la FPGA. Los componentes se configuraron mediante *generic* para especificar la frecuencia de la señal de reloj, cuyo valor depende del circuito oscilador de la PCB donde se implemente el diseño. En base a este *generic* se determinaron los valores de los parámetros de los componentes y se establecieron las distintas temporizaciones que se utilizaron en el *bridge* DALI. A continuación, se describe la comunicación entre el *host* y el *bridge* DALI y los tres componentes de los que consta el *bridge* DALI: UART, procesador DALI y TRCM.

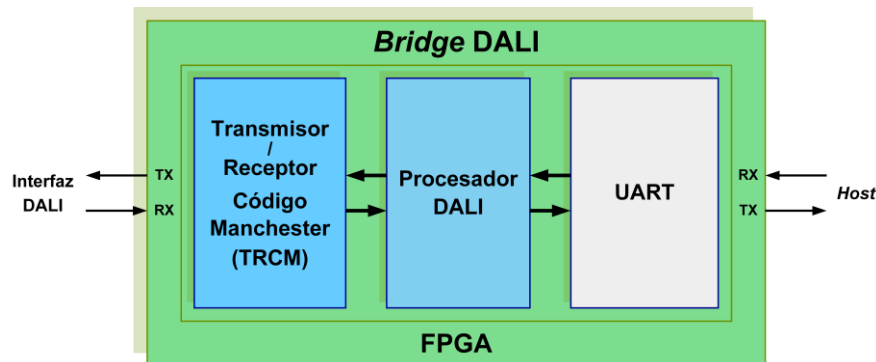


Figura 4.5 Diagrama de bloques del *bridge* DALI.

4.2.1. Comunicación entre el *host* y el *bridge* DALI

El *bridge* DALI puede formar parte del nodo sensor inalámbrico en la misma PCB o se puede implementar en alguna otra PCB como periférico, gracias a su descripción portable y al diseño realizado ad-hoc. En ambos casos la comunicación puede ser mediante un puerto serie asíncrono estándar. De esta forma, cualquier dispositivo puede actuar como *host*, por ejemplo, un MCU o un *desktop computer*. El puerto serie se configuró por defecto a una velocidad de 9600 bps, ya que la transferencia de datos es de bajo volumen. Además, es la

velocidad utilizada de forma predeterminada en muchos sistemas embebidos de comunicación serie. Por otra parte, se puede configurar la FPGA con cualquier otra velocidad estándar u otra velocidad de transferencia que se desee. El formato de configuración es 8 bits por carácter, un bit de paridad par y un bit de *stop*.

El *bridge* DALI permite en su descripción controlar las luminarias mediante la decodificación de los comandos, que se han definido y que envía el *host*. Para tal fin, el *host* envía comandos al *bridge* DALI que primero los decodifica y, según el tipo de comando, los ejecuta, enviando las tramas correspondientes a la luminaria y en el caso de tener una respuesta de la luminaria la envía de nuevo al *host*. La petición del *host* al *bridge* puede constar de uno a tres bytes dependiendo del tipo de comando que se indica en el primer byte enviado. Un ejemplo de la petición del *host* al *bridge* de tres bytes se indica en la Figura 4.6. El primer byte es el comando para el *bridge*, el segundo la dirección del dispositivo DALI y el tercero es el comando a ejecutar. Hay que tener en cuenta que, como la petición se envía a través de un enlace serie tipo UART en formato asíncrono, existe la posibilidad de que los bytes de una petición se envíen de forma continua o que pueda transcurrir un cierto tiempo entre cada byte.



Figura 4.6. Formato de la trama de peticiones del *host*.

Los datos de retorno desde la luminaria son enviados por el *bridge* al *host*, en el formato de datos que se muestra en la Figura 4.7. El formato está dividido en dos partes, el *acknowledgment* y la respuesta, cada parte de un tamaño de 1 byte. El *acknowledgment* indica si se devuelven datos válidos desde la luminaria, y, por lo tanto, contenidos en la respuesta.



Figura 4.7. Formato de la trama de respuesta al *host*.

El conjunto de comandos está dividido en dos tipos, uno para los comandos del *bridge* DALI y otro para las instrucciones de la interfaz DALI. Los comandos de la interfaz DALI

no son manipulados por el *bridge* DALI, solo los transfiere hacia el *host*. Los comandos que actualmente se ha implementado se indican en la Tabla 4.1.

Tabla 4.1 Comandos generales del *bridge* DALI y propios de la interfaz DALI.

Comando	Descripción
00	Echo
01	Reset del <i>bridge</i>
02	Retorna el estado del <i>bridge</i>
03	Reservado
04	Envía. Transmite la secuencia a la interfaz DALI de 16 bits
05	Envía/recibe. Transmite la secuencia a la interfaz DALI de 16 bits y espera como máximo 100 ms para un retorno de 8 bits desde la interfaz DALI.
06	Envío doble. Transmite una secuencia a la interfaz DALI de 16 bits dos veces con 10 ms de diferencia
07	Identificador del <i>bridge</i> DALI

Comandos del *bridge* DALI:

- El comando 00 (Echo) permite al *host* conocer si tiene un *bridge* conectado al puerto serie.
- El comando 01 se utiliza para inicializar los componentes del *bridge* DALI a un estado conocido.
- El comando 02 se utiliza para conocer el estado de la interfaz DALI. En los datos de retorno desde la luminaria, hay dos bits para conocer su estado. Por ejemplo, el bit de estado de sobrecarga (bit 2) determina la existencia de un cortocircuito en el bus DALI durante más de 254 ms, lo cual permite desactivar automáticamente la interfaz DALI. El otro bit (bit 1) indica si el *bridge* DALI ha recibido la respuesta después de la ejecución del comando 05, de forma que, si no se reciben los datos, este bit es afirmado ('1').

Por otra parte, el conjunto de comandos DALI (comandos 04 a 07) permiten controlar la iluminación, conocer el estado y configurar la luminaria. A continuación, se describen estos comandos:

- El comando 04 se utiliza para enviar comandos a la interfaz DALI para el control de iluminación, como, los que controlan la potencia de arco.
- El comando 05 se utiliza para enviar comandos a la interfaz DALI y conocer el estado de la luminaria, por ejemplo, para determinar si una luminaria funciona, si se encuentra encendida, y para verificar el nivel de iluminación.
- El comando 06 se utiliza para enviar comandos a la interfaz DALI para configurar la luminaria, por ejemplo, para encender, apagar o restablecer la luminaria.

- El comando 07 devuelve un identificador que se puede asociar al *bridge* DALI o al nodo sensor inalámbrico completo. Este identificador puede estar programado en la FPGA o se puede obtener de un selector exterior, que esté conectado a la FPGA e integrado en el nodo sensor inalámbrico.

Dependiendo del comando que se haya solicitado al *bridge*, éste devolverá una respuesta. Por ejemplo, cuando se ejecuta el comando (05) para consultar el estado de la luminaria, se recibe una respuesta DALI (1 byte) que permite verificar varios factores de la luminaria, cuya codificación de los bits se indica en la Tabla 4.2. Por ejemplo, cuando el bit menos significativo es cero ('0'), indica que la luminaria es OK; si el siguiente bit está activo ('1'), indica que la luminaria esta fundida; si el bit en la posición dos tiene el valor cero ('0'), indica que la luminaria está apagada, de lo contrario ('1') está encendida; y si el bit más significativo es igual a cero ('0'), indica que la alimentación de energía eléctrica esta OK, de lo contrario ('1') no existe alimentación de energía eléctrica.

Tabla 4.2 Codificación de la respuesta DALI.

Respuesta DALI				Decodificación
b(7)	b(2)	b(1)	b(0)	
			0	Luminaria es OK
		1		Luminaria fundida
	1			Luminaria encendida
1				No existe alimentación de energía eléctrica

4.2.2. UART

La representación a nivel de bloque de la Figura 4.8 corresponde a la UART. La función de la UART dentro del *bridge* DALI es realizar la comunicación entre el procesador DALI y el *host*, transmitiendo y recibiendo datos entre ambos dispositivos. Para cumplir con estos propósitos la UART tiene dos componentes, el transmisor y el receptor. En la Figura 4.8 los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho. También se puede observar la conexión que tiene la UART con el procesador DALI y con el *host*.

Los puertos de entrada de la UART son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento de todos los componentes.
- Reset. Es la señal que inicializa todos los componentes a un estado conocido.

- WR_UART. Es la señal de control de escritura que el procesador DALI activa cuando quiere iniciar una transmisión.
- Dato_In_UART(7:0). Es el bus de los datos en el que el procesador DALI pone los datos a transmitir.
- RD_Dato_UART. Es la señal de control de la lectura de la UART que el procesador DALI debe poner en '1' para leer el dato recibido.
- Clear_Error_UART. Es la señal que borra el error de paridad.
- Rx_En_UART. Es la señal de habilitación del receptor. Debe ponerse en '1' para permitir la recepción de los datos en la señal de RXD_UART.
- RXD_UART. Es la señal por donde se reciben en serie los datos.

Los puertos de salida de la UART son:

- TXD_UART. Es la señal por donde se envía la transmisión serie.
- Ready_Tra_UART. Es la señal que indica el estado del transmisor. Si se encuentra en '1', el procesador DALI puede escribir un dato nuevo.
- Dato_Rec_UART. Es la señal de estado que al estar en '1' indica que se ha recibido un dato.
- Error_Par_UART. Es la señal de error que indica una paridad incorrecta en el dato recibido, al tener un valor de '1' se ha producido un error de paridad.
- DO_UART(7:0). Es el bus de los datos del receptor.

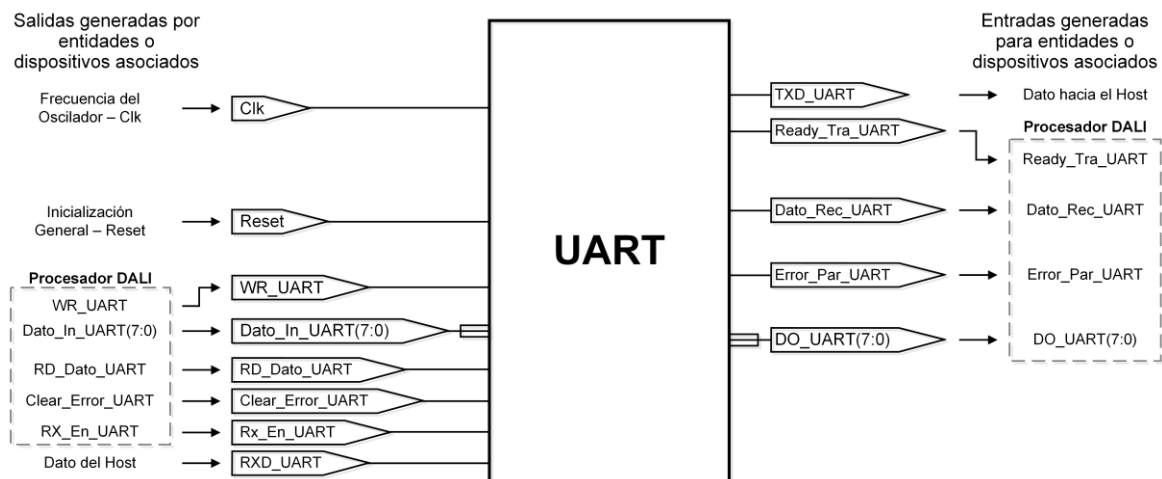


Figura 4.8 Representación a nivel de bloque de la UART.

El diagrama de bloques de la UART se puede observar en la Figura 4.9, en la cual, tanto para el receptor como para el transmisor, los puertos de entrada se muestran en el lado

izquierdo y los puertos de salida en el lado derecho. Además, se pueden observar las conexiones específicas que tienen el transmisor y receptor con el procesador DALI y con el *host* respectivamente. Se describió la UART de forma que se puede configurar cualquier velocidad de transmisión. Por defecto, se configuró a 9600 bits por segundo (104 μ s por bit). El dato que se transmite tiene un tamaño de 11 bits. La transmisión comienza por el bit menos significativo.

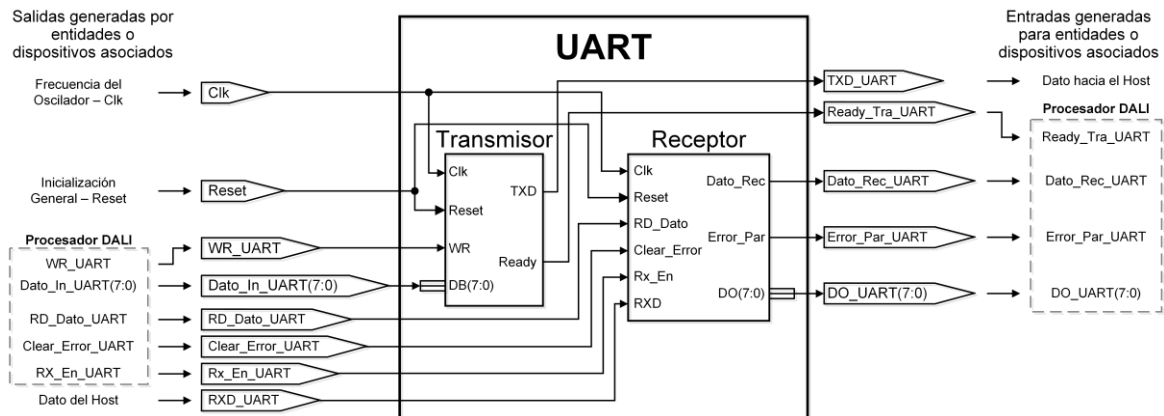


Figura 4.9 Diagrama de bloques de la UART.

El diagrama de bloque del transmisor de la UART se muestra en la Figura 4.10. El objetivo de este componente es transmitir datos entre el procesador DALI y el *host*.

Los puertos de entrada del transmisor de la UART son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento del transmisor.
- Reset. Es la señal que inicializa el transmisor a un estado conocido.
- WR. Es la señal de control de escritura que el procesador DALI activa para iniciar una transmisión. Es activa a nivel lógico alto.
- DB(7:0). Es el bus de los datos que el procesador DALI utiliza para colocar los datos a transmitir.

Los puertos de salida del transmisor de la UART son:

- TXD. Es la señal con la que se envía la transmisión serie.
- Ready. Es la señal que indica el estado del transmisor. Si se encuentra a '1', el procesador DALI puede escribir un nuevo dato.

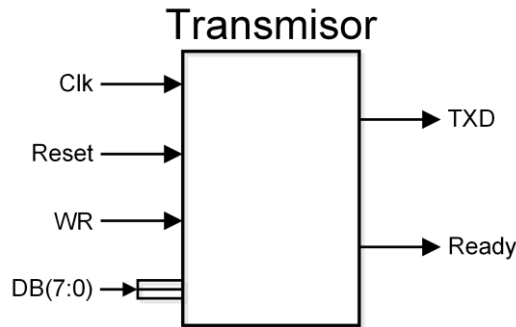


Figura 4.10 Diagrama de bloque del transmisor de la UART.

La ruta de datos del transmisor de la UART se muestra en la Figura 4.11, la cual está constituida por cuatro registros: un registro *buffer* de entrada (Reg_Buffer), un registro serializador (Reg_Serial), un registro generador de baudios (Gen_baudios) y un registro contador (Reg_counter). El registro *buffer* realiza la operación de carga paralela, ya que su función es almacenar el carácter a transmitir. El registro serializador realiza las operaciones de carga paralela y desplazamiento a la derecha para hacer la conversión de paralelo a serie. Este tiene un tamaño de 10 bits, ya que debe almacenar el bit de inicio ('0'), en la posición menos significativa, los 8 bits a transmitir y el bit de paridad par. El generador de baudios genera la señal *Tic* que sincroniza la transmisión del dato. El registro contador contabiliza los bits que se transmiten para poder determinar el final de la transmisión. La ruta de datos y la unidad de control del transmisor de la UART se describió mediante una FSM. En la Figura 4.11 los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho.

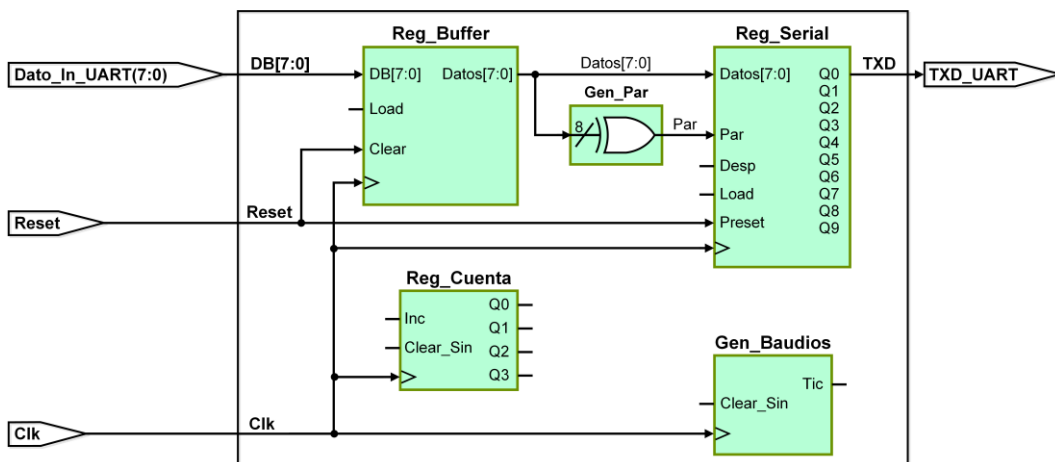


Figura 4.11 Ruta de datos de la FSM del transmisor de la UART.

El diagrama ASM de la unidad de control del transmisor se muestra en la Figura 4.12. El proceso de transmisión se inicia cuando la unidad de control detecta que la señal WR_UART está activada, entonces se carga el dato paralelo aplicado en Dato_In_UART en el registro *buffer*. Después se transfiere el contenido del registro *buffer* al registro serializador y se inicializa con cero el registro generador de baudios. Una vez almacenado el carácter a transmitir en el registro serializador, se habilita el conteo en el registro generador de baudios. Cada vez que el registro generador de baudios determine que se ha cumplido el tiempo de bit (*Tic* a '1'), se transmite un bit por la salida TXD, es decir, se realiza un desplazamiento a la derecha en el registro serializador introduciendo un '1' por la izquierda. Evidentemente se realizan posteriormente los desplazamientos necesarios para transmitir todos los bits, para ello, se utilizó el registro contador, que incrementa su cuenta cada vez que se transmite un bit. Al introducir un bit con valor de '1' en cada desplazamiento se genera el bit de *stop* automáticamente después de transmitir los 10 bits, previamente almacenados, por lo que se ahorra un bit en el registro serializador.

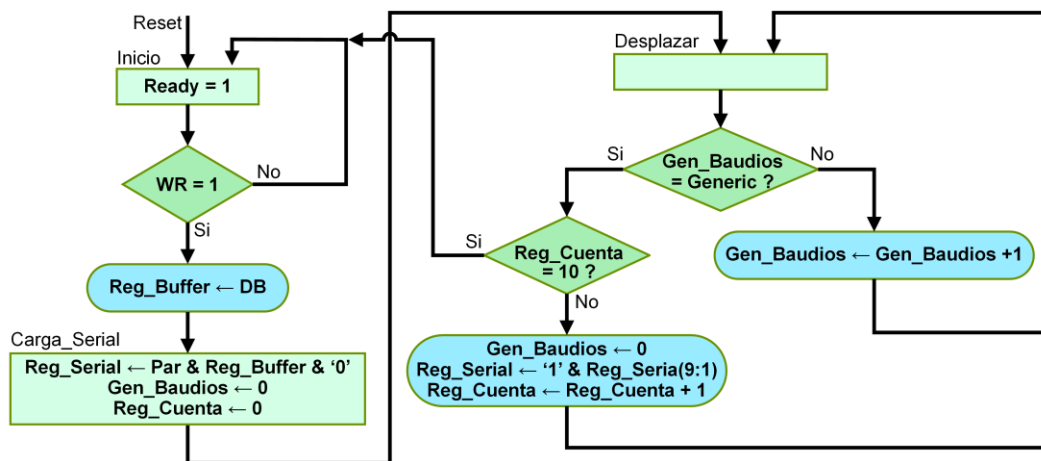


Figura 4.12 Unidad de control de la FSM del transmisor de la UART.

El diagrama de bloques del receptor de la UART se muestra en la Figura 4.13. El objetivo de este componente es recibir los datos que el *host* envía al *bridge* DALI.

Los puertos de entrada del receptor de la UART son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento del receptor.
- Reset. Es la señal que inicializa al receptor a un estado conocido.
- RD_Dato. Es la señal para habilitar la lectura del receptor de la UART, que es activada por el procesador DALI para leer el dato recibido. Es activa a nivel alto
- Clear_Error. Es la señal que borra el error de paridad.

- Rx_En. Es la señal de habilitación del receptor. Debe al ponerse a '1' para permitir la recepción de los datos recibidos por la entrada RXD.
- RXD. Es la entrada por donde se reciben en serie los datos que envía el *host*.

Los puertos de salida del receptor de la UART son:

- Dato_Rec. Es la salida de estado del receptor, que si está a '1' indica que se ha recibido un dato.
- Error_Par. Es la señal de error que indica una paridad incorrecta en el dato recibido, al tener un valor de '1' se ha producido un error de paridad.
- DO(7:0). Es el bus de los datos del receptor.

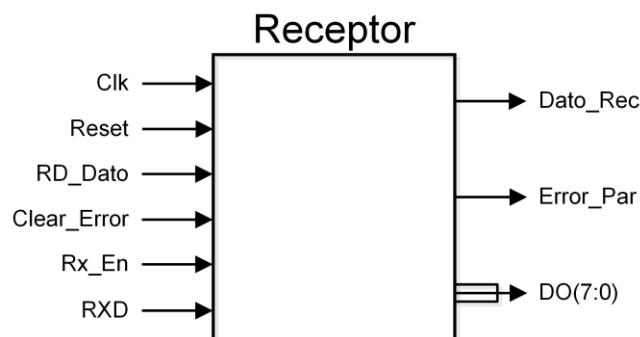


Figura 4.13 Diagrama de bloque del receptor de la UART.

La ruta de datos del receptor de la UART se muestra en la Figura 4.14. El receptor muestrea la entrada de recepción (RXD_UART) aproximadamente en la mitad del tiempo de bit para asegurar que la recepción sea correcta. Por ello, se realiza el muestreo a una frecuencia 16 veces superior a la de transmisión. La ruta de datos consta de un registro *buffer* de entrada (Reg_Buffer), un registro de datos (Reg_Datos), un registro divisor de frecuencia (Reg_Divide), un registro generador de baudios (Reg_Baudios), un registro contador (Reg_Cuenta) y un registro de estado (Reg_Estado). El registro *buffer* almacena el carácter recibido. El registro de datos realiza las operaciones de carga serie y desplazamiento a la derecha para hacer la conversión de serie a paralelo. El registro divisor de frecuencia genera una señal de tipo pulso (*Tic16*) con una frecuencia 16 veces superior a la de transmisión para garantizar el muestreo correcto de la entrada de recepción (RXD_UART), tal y como se indicó anteriormente. El generador de baudios, a partir de la señal anterior, activa su salida para indicar el momento de captura del bit recibido. El registro contador contabiliza los bits que se reciben para determinar el fin de la recepción del dato. El registro de estado, como se verá posteriormente, genera las salidas Dato_Rec y

Error_Par. La ruta de datos y la unidad de control del receptor de la UART se describió mediante una FSM. En la Figura 4.14 los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho.

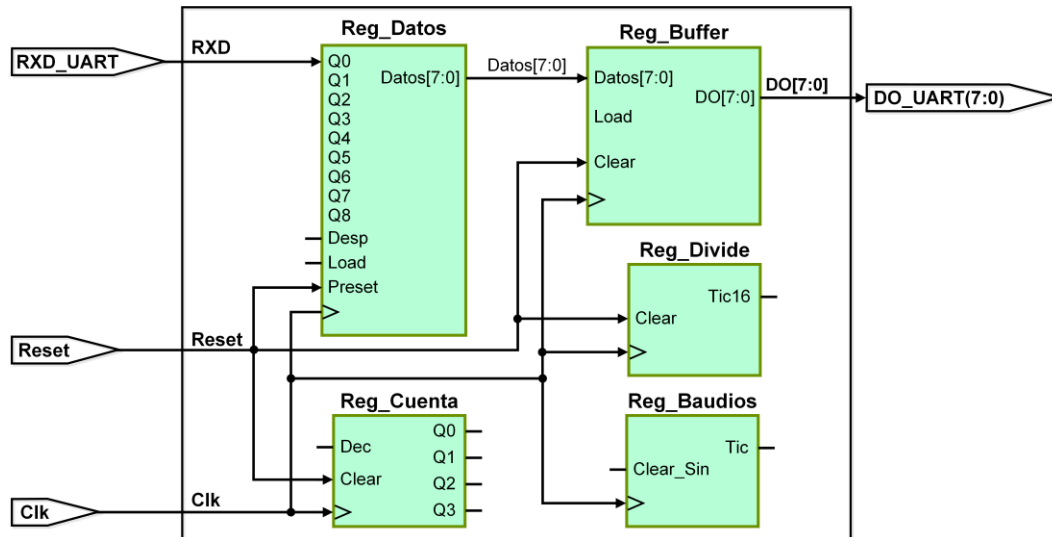


Figura 4.14 Ruta de datos de la FSM del receptor de la UART.

El diagrama ASM de la unidad de control del receptor de la UART se muestra en la Figura 4.15. En el estado Inicio el registro generador de baudios y el bit de paridad se inicializan a cero, y el registro contador se carga con el número de bits a recibir. El proceso de recepción se inicia cuando la unidad de control detecta que la entrada Rx_En está activada y la entrada RXD_UART tiene un valor cero (bit de inicio), entonces se pasa al estado Espera_1_2 para sincronizarse correctamente con el punto medio del bit de inicio. Una vez sincronizada la recepción se pasa al estado Desplaza, en el que se espera sucesivamente un tiempo de bit para ir muestreando los bits restantes en el punto medio de cada uno de ellos. El tiempo de bit se mide mediante el generador de baudios, por lo que cada vez que éste llega a su cuenta final (15) se almacena el bit recibido en la entrada RXD_UART, realizando un desplazamiento a la derecha en el registro de datos, y se decrementa el registro contador. Al mismo tiempo se actualiza el valor de la paridad par de los bits recibidos. Este proceso se repite hasta que el registro contador se pone a cero ('0'), indicando que se han recibido todos los bits de datos y el de paridad. Finalmente, la FSM pasa al estado "Stop" en el que se espera un tiempo de bit para recibir el bit de *stop* y, una vez recibido el carácter, se transfiere el dato recibido al registro *buffer* y se pasa al estado Comprobar en el que se actualiza el registro de estado.

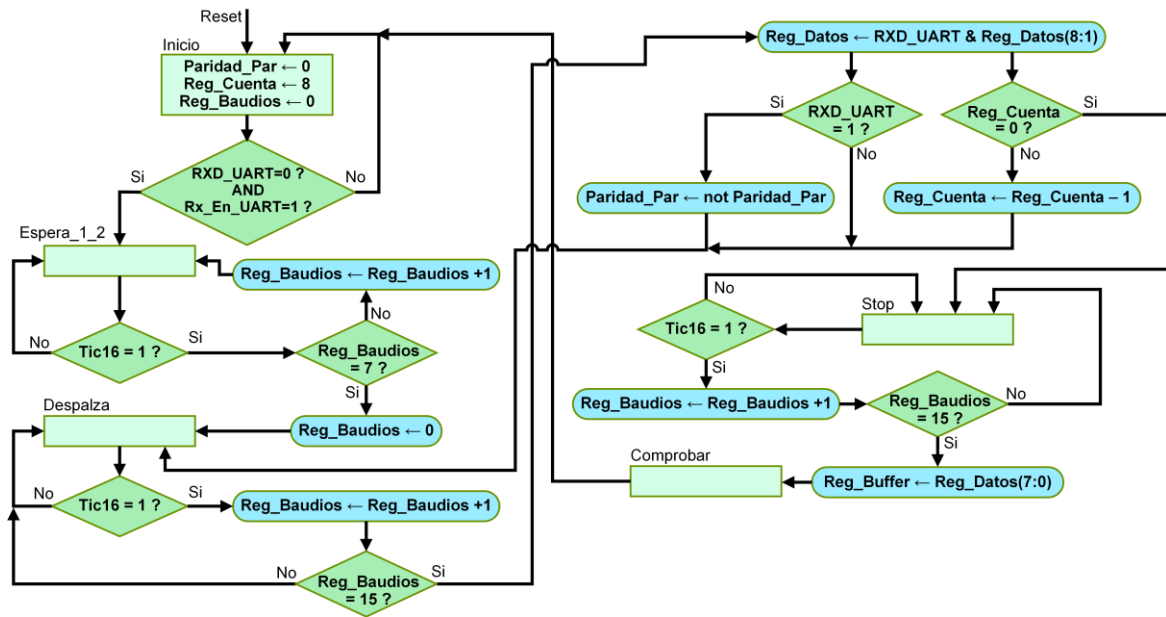


Figura 4.15 Unidad de control de la FSM del receptor de la UART.

Las salidas Dato_Rec_UART y Error_Par coinciden con los bits del registro de estado del receptor, cuya representación a nivel de bloque se muestra en la Figura 4.16. La salida Dato_Rec_UART se pone a '1', cuando el estado actual de la FSM es Comprobar, para indicar que se acaba de recibir un dato y, una vez activa, se mantiene hasta que el procesador DALI lee el dato activando la entrada de control RD_Dato_UART.

La salida Error_Par se activa si al recibirse un dato, se ha producido un error de paridad. Por tanto, se pone a '1' si la FSM está en el estado Comprobar y Paridad_Par está a '1'. Una vez activa, se mantiene hasta que se active la entrada Clear_Error_UART.

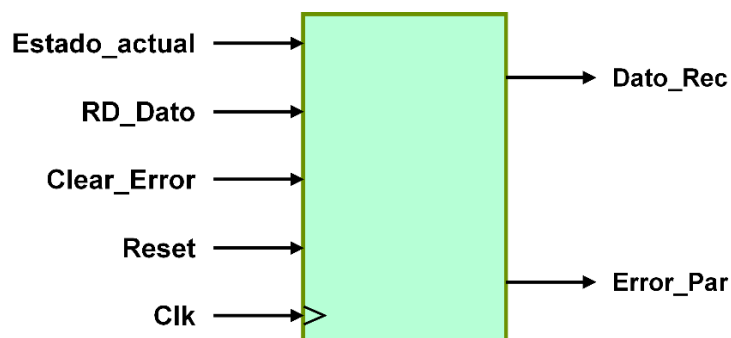


Figura 4.16 Representación a nivel de bloque del registro de estado.

4.2.3. Procesador DALI

El procesador DALI tiene como objetivo controlar el intercambio de información entre la UART y el TRCM. Realiza la decodificación de las instrucciones y comandos DALI, que recibe por parte del *host*, y codifica las respuestas de la interfaz DALI que devuelve al *host*. La representación a nivel de bloque del procesador DALI se muestra en la Figura 4.17, los puertos de entrada se encuentran en el lado izquierdo y los puertos de salida en el lado derecho. También se puede observar la conexión que tiene el procesador DALI con el TRCM y con la UART.

Los puertos de entrada del procesador DALI son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento del procesador DALI.
- Reset. Es la señal que inicializa el procesador DALI a un estado conocido.
- Ready_Tra_UART. Es la señal que indica el estado en que se encuentra el transmisor de la UART. Si la señal está a '1', el procesador DALI puede escribir un nuevo dato.
- Data_Rec_UART. Es la señal de estado que se activa para indicar que la UART ha recibido un dato. Es activa a nivel lógico alto.
- Error_Par_UART. Es la señal de control que indica un error de paridad en el receptor de la UART.
- DO_UART(7:0). Es el bus de los datos en el que se obtiene los datos recibidos por la UART.
- Ready_Tra_TRCM. Es la señal que indica el estado del transmisor del TRCM. Si está a '1', el procesador DALI puede escribir un nuevo dato en el TRCM.
- Dato_Rec_TRCM. Es la señal de estado que indica que se ha recibido un dato en el TRCM. Es activa a nivel lógico alto.
- DO_TRCM(15:0). Se conecta a la salida de datos del receptor del TRCM.

Los puertos de salida del procesador DALI son:

- WR_UART. Es la salida de control que controla la escritura del transmisor de la UART.
- RD_Dato_UART. Es la señal de control que permite leer el dato recibido en la UART. El procesador DALI la pone a '1' al activarse la salida Dato_Rec_UART, que indica que la UART ha recibido un dato.

- Clear_Error_UART. Es la señal de control que borra el error de paridad en el receptor de la UART al ser activada con '1'.
- Dato_In_UART(7:0). Es el bus donde se colocan los datos a transmitir por la UART.
- RX_En_UART. Es la señal de control con la cual el procesador DALI habilita el receptor de la UART. Es activa a nivel lógico alto.
- Reset_TRCM. Es la señal de control para inicializar el TRCM a un estado conocido. Es activa a nivel lógico alto.
- WR_TRCM. Es la señal que controla la escritura del TRCM, una vez que se activa, se inicia la transmisión del comando. Es activa a nivel lógico alto.
- Dato_In_TRCM(15:0). Es el bus de los datos (16 bits) en el cual el procesador DALI pone el comando a transmitir por la TRCM.
- Rx_En_TRCM. Es la señal de control con la cual el procesador DALI habilita el receptor de la TRCM. Es activa a nivel lógico alto.

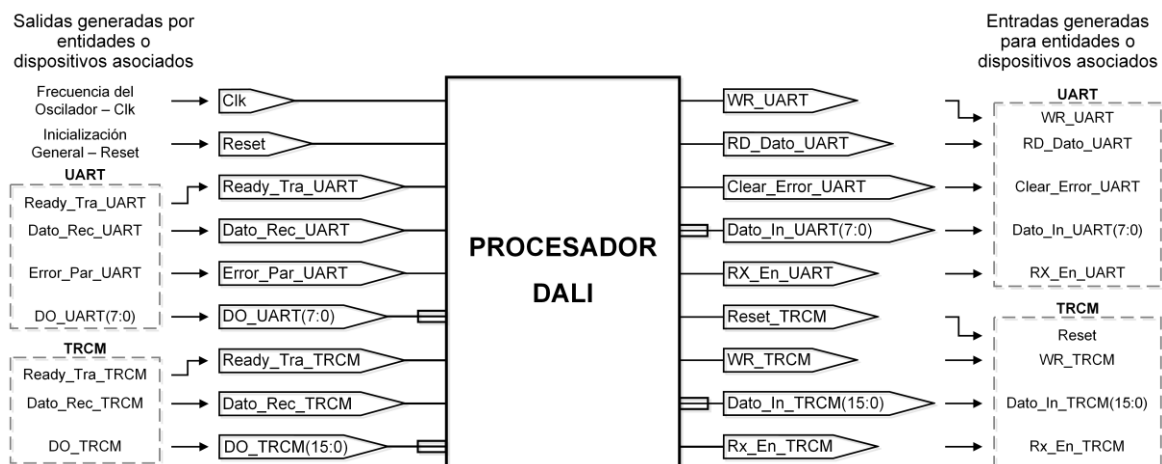


Figura 4.17 Representación a nivel de bloque del procesador DALI.

La ruta de datos y la unidad de control del procesador DALI se describieron mediante una FSM, cuyo diagrama de estados simplificado se muestra en la Figura 4.18. Se describe la ruta de datos y el funcionamiento de la FSM. En primer lugar, la ruta de datos consta de siete registros que son: Reg_Instrucción, Dato_UART, Forward_TRCM, Backward_TRCM, Bridge_acknowledge, DALI_acknowledge y Falso_acknowledge. Primeramente, se describe el propósito de cada registro. El registro Reg_Instrucción almacena el comando recibido en la UART. El registro Dato_UART almacena el dato que

se transmite a la UART. El registro Forward_TRCM tiene un tamaño de 16 bits y se utiliza para almacenar el dato que se envía al TRCM. El registro Backward_TRCM almacena los datos de retorno del TRCM. El registro Bridge_acknowledge almacena la respuesta de la ejecución de un comando del *bridge*. El registro DALI_acknowledge almacena la respuesta de la ejecución de un comando DALI.

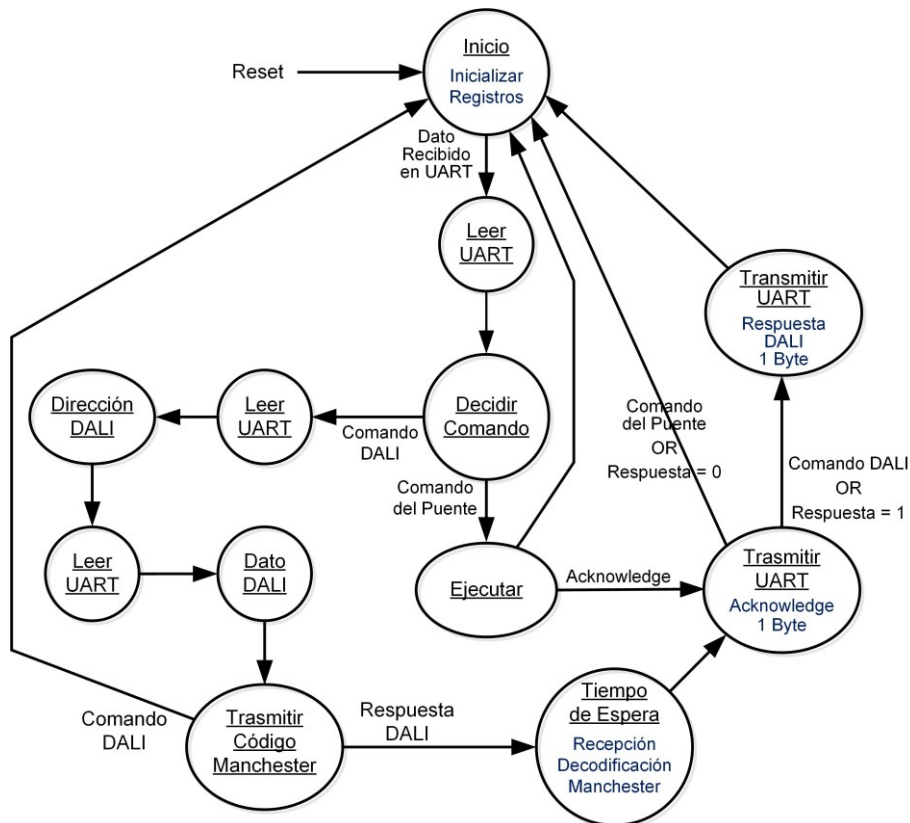


Figura 4.18 Diagrama de estados de la FSM del procesador DALI.

Por otra parte, el funcionamiento de la FSM del procesador DALI es el siguiente. En el estado “Inicio” de la FSM se inicializan todos los registros, las señales auxiliares y de salida. El proceso de la decodificación del comando se inicia cuando la unidad de control monitoriza que la señal Dato_Rec_UART está activa, instante en el que se lee por la entrada DO_UART(7:0) el dato proveniente de la UART y se almacena en el registro Reg_Instrucción. A continuación, se interpreta el comando para determinar si es un comando para el *bridge* o un comando propio de DALI. En caso de ser un comando para el *bridge* se ejecuta o si se solicita respuesta, los parámetros actuales del estado del *bridge* se cargan en el registro Bridge_acknowledge y, posteriormente, se transfieren a la salida en Dato_In_UART(7:0) para enviarlos a la UART. En ambos casos se vuelve al estado inicial.

Si en el estado “Decidir Comando” se determina que es un comando propio de DALI, se lee la UART (Leer UART) y se almacenan los datos en los ocho bits más significativos del registro Forward_TRCM (Dirección DALI), después, nuevamente se lee la UART (Leer UART) y se almacenan los datos en los ocho bits menos significativos del registro Forward_TRCM (Dato DALI). Después se transfieren los datos al puerto de salida, Dato_In_TRCM(15:0), y se envían al TRCM para que realice la codificación Manchester (Transmitir Código Manchester), y que envíe la trama a la interfaz DALI. Si el comando DALI solicita una respuesta, espera el tiempo determinado por el comando (Tiempo de Espera) y se carga el dato recibido en el registro Backward_TRCM. Posteriormente, se carga en el registro DALI_acknowledge (Trasmitir UART) la información que indica que los datos de respuesta de la luminaria son válidos. Después, se transfiere a la salida Dato_In_UART(7:0) (Trasmitir UART) y se escriben en la UART, para enviar la respuesta de la interfaz DALI al *host*. En caso de que el tiempo determinado para recibir una respuesta desde la interfaz DALI transcurra y no se reciba un dato, el registro Falso_acknowledge se transfiere a la salida Dato_In_UART(7:0) (Trasmitir UART) y se envía a la UART para notificar que no existió respuesta desde la interfaz DALI. En resumen, el procesador DALI recibe un comando de la UART lo decodifica y lo trasmite al TRCM. Si el comando solicita una respuesta después de su ejecución, el componente TRCM envía los datos de retorno al componente procesador DALI para su codificación y transferirlos a la UART.

4.2.4. Transmisor / Receptor Código Manchester

El TRCM tiene como función realizar el intercambio de datos entre el procesador DALI y la interfaz DALI, de tal forma que permite el cambio de señalización requerido entre las diferentes interfaces. Este bloque es el más complejo debido a la codificación/decodificación Manchester. La representación a nivel de bloque del TRCM se puede observar en la Figura 4.19. Tiene seis puertos de entrada y cuatro de salida, los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho. También se puede identificar la conexión que tiene el TRCM con el procesador DALI y con la interfaz DALI.

Los puertos de entrada del TRCM son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento de todos los componentes.
- Reset. Inicializa todos los componentes a un estado conocido.

- WR_TRCM. Es la señal de control de la escritura, que el procesador DALI activa para almacenar el comando en el transmisor e iniciar su envío a la interfaz DALI. Es activa a nivel lógico alto.
- Dato_In_TRCM(15:0). Es el bus de los datos en el que el procesador DALI pone el comando que se debe transmitir.
- Rx_En_TRCM. Es la señal de habilitación del receptor del TRCM. Es activa a nivel lógico alto.
- RXD_MCTR. Es la señal por donde entran los datos para la recepción serie provenientes de la interfaz DALI.

Los puertos de salida del TRCM son:

- TXD_TRCM. Es la señal por donde se realiza la transmisión serie (con la codificación del código Manchester) que se conecta a la interfaz DALI.
- Ready_Tra_TRCM. Es la señal que indica el estado del transmisor. Si está a nivel lógico alto, indica que se puede escribir un nuevo dato.
- Dato_Rec_TRCM. Es la señal de estado que se activa para indicar al procesador DALI que se ha recibido un dato desde la interfaz DALI. Es activa a nivel lógico alto.
- Dato_Out_MCTR(7:0). Es el bus de los datos de salida del receptor.

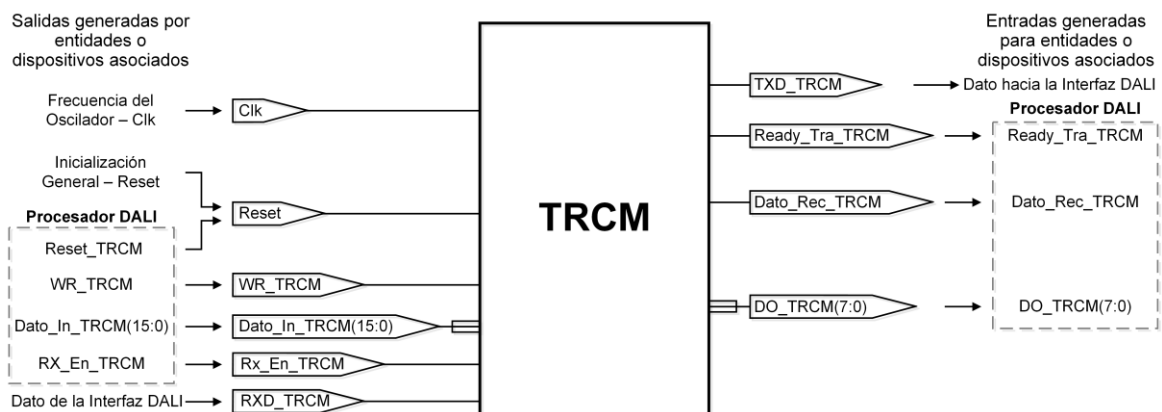


Figura 4.19. Representación a nivel de bloque del TRCM.

El TRCM transmite y recibe los datos usando la codificación Manchester. En la Figura 4.20 se muestra el diagrama de bloques del TRCM, en la que se puede apreciar que consta de dos componentes, denominados transmisor y receptor. Tanto para el transmisor como para el receptor, los puertos de entrada se muestran en el lado izquierdo y los puertos de

salida en el lado derecho. También se puede observar la conexión que tiene el TRCM con el procesador DALI y la interfaz DALI.

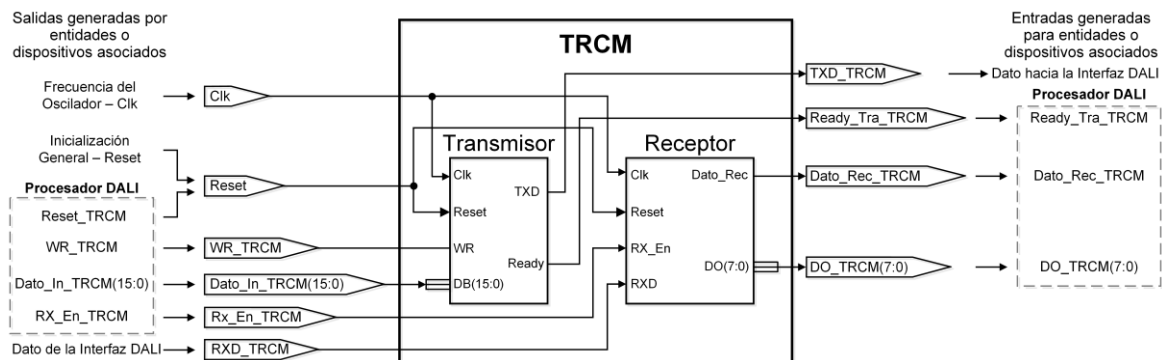


Figura 4.20. Diagrama de bloques de la arquitectura del TRCM.

La representación a nivel de bloque del transmisor del TRCM se muestra en la Figura 4.21. El objetivo de este componente es transmitir los comandos entre el procesador DALI y la interfaz DALI. Tiene cuatro puertos de entrada y dos de salida, los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho.

Los puertos de entrada del transmisor del TRCM son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento del transmisor.
- Reset. Es la señal que inicializa el transmisor a un estado conocido.
- WR. Es la señal de control de escritura que el procesador DALI activa para iniciar una transmisión. Es activa a nivel lógico alto.
- DB(15:0). Es el bus de datos que el procesador DALI utiliza para enviar los datos a transmitir.

Los puertos de salida del transmisor del TRCM son:

- TXD. Es la señal con la que se transmiten los datos y los comandos en serie a la interfaz DALI.
- Ready. Es la señal que indica el estado del transmisor. Si se encuentra a '1', el procesador DALI puede escribir un nuevo dato.

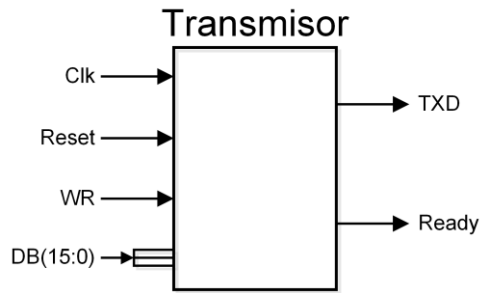


Figura 4.21 Diagrama de bloque del transmisor del TRCM.

Los comandos que se transmiten tienen un tamaño de 19 bits. La transmisión comienza por el bit más significativo. La ruta de datos del transmisor del TRCM se muestra en la Figura 4.22. Esta consta de un registro *buffer* de entrada (Reg_Buffer), un registro serializador (Reg_Serial), un registro generador de baudios (Gen_Baudios) y un registro contador (Reg_Cuenta). El registro *buffer* realiza la operación de carga paralela, y su función es almacenar el comando a transmitir hacia la interfaz DALI. El registro serializador tiene un tamaño de 17 bits, ya que contiene el bit de inicio ('1'), en la posición más significativa, y los 16 bits a transmitir. Este registro realiza las operaciones de carga paralela y desplazamiento a la izquierda para hacer la conversión de paralelo a serie. El generador de baudios genera una señal de tipo pulso (*Tic*) que se activa cada vez que se cumpla la temporización de medio bit, es decir, 416,67 μ s. El registro contador contabiliza los bits que se transmiten para concluir la transmisión. La ruta de datos y la unidad de control del transmisor se describieron mediante una FSM. En la Figura 4.22 los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho.

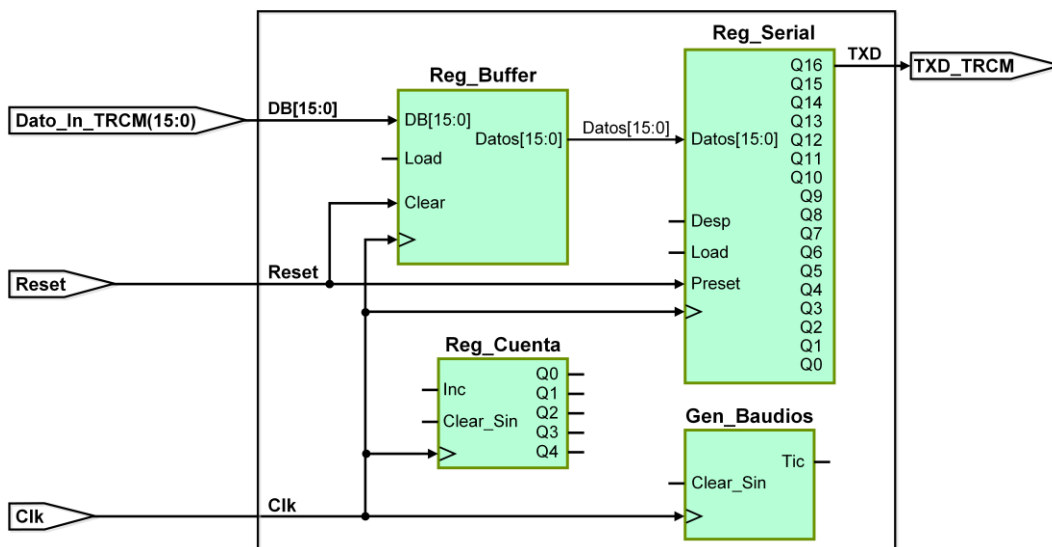


Figura 4.22 Ruta de datos de la FSM del transmisor del TRCM.

El diagrama ASM de la unidad de control del transmisor del TRCM se muestra en la Figura 4.23. En el estado “Inicio” se activa `Ready_Tra_TRCM` para indicar al procesador DALI que el TRCM no está transmitiendo y, por tanto, el procesador DALI puede escribir un comando para transmitirlo a la interfaz DALI. El proceso de transmisión comienza cuando la unidad de control detecta que la entrada de escritura, `WR_TRCM`, está activada. Si es así, se carga en paralelo el dato a transmitir en el registro `buffer`, y espera a que se desactive `WR_TRCM`. Una vez que la función de escritura fue desactivada, se pasa al estado “Carga_Serial” en el que se transfiere el contenido del registro `buffer` al registro serializador, y se inicializan a cero el registro generador de baudios y el registro contador. Los estados “Desplaza” y “Desplaza_2” realizan la transmisión de un bit en el código Manchester. Para ello, en el estado “Desplaza” se transmite por la salida `TXD_TRCM` el primer medio bit, que es el complemento del valor del bit de datos, y en el estado “Desplaza_2” el segundo medio bit, que es el valor del bit. De esta manera se genera la transición correspondiente según la codificación Manchester. En ambos estados se habilita el conteo del tiempo en el registro generador de baudios para contar la cantidad de pulsos equivalentes al tiempo de medio bit (416,67 μ s), y si éste llega a su cuenta final, se inicializa a cero. Del estado “Desplaza” se pasa al estado “Desplaza_2”, una vez transmitido el primer medio bit.

En el estado “Desplaza_2” se realiza un desplazamiento de un bit a la izquierda en el registro serie y se incrementa el registro contador, cada vez que el generador de baudios activa “*Tic*”. Esta secuencia se repite hasta que se hayan transmitido todos los bits de datos, por lo que se debe comprobar el valor del registro contador. Si éste ha llegado a 16, se pasa al estado “Stop”, para transmitir los dos bits de *stop*, de lo contrario se vuelve al estado “Desplaza”. En el estado “Stop” se pone la salida de transmisión a uno (‘1’) durante cuatro medios bits para terminar la transmisión de la trama “Forward”. Por ello, se habilita la cuenta del registro generador de baudios y, cada vez que llega a su cuenta final, se inicializa a cero, y se incrementa el registro contador. Este proceso se repite hasta que el registro contador tiene el valor 20, lo cual indica que ya se han transmitido todos los bits, por lo que se vuelve al estado “Inicio”, en el que se pone la salida de transmisión a uno (‘1’), que corresponde al estado “Espera”.

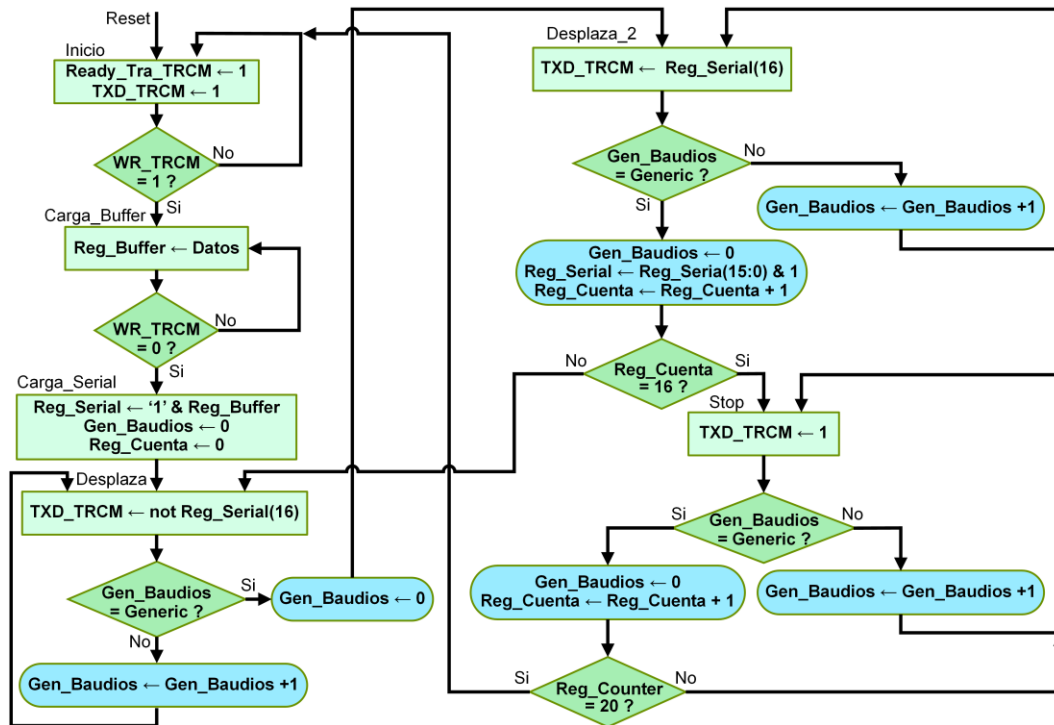


Figura 4.23 Unidad de control de la FSM del transmisor del TRCM.

El diagrama de bloque del receptor del TRCM se muestra en Figura 4.24. El objetivo de este componente es recibir los datos entre la interfaz DALI y el procesador DALI. Tiene cuatro puertos de entrada y dos de salida, los puertos de entrada se muestran en el lado izquierdo y los puertos de salida en el lado derecho.

Los puertos de entrada del receptor del TRCM son:

- CLK. Es la señal de reloj que sincroniza el funcionamiento del receptor.
- Reset. Es la señal que inicializa al receptor a un estado conocido.
- Rx_En. Es la señal de habilitación del receptor, de forma que al ponerse a '1' permite la recepción de los datos por la entrada RXD.
- RXD. Es la entrada por donde se reciben en serie los datos.

Los puertos de salida de receptor del TRCM son:

- Dato_Rec. Es la señal de estado del receptor, que, si se pone a '1', indica que se ha recibido un dato.
- DO(7:0). Es el bus de los datos del receptor.

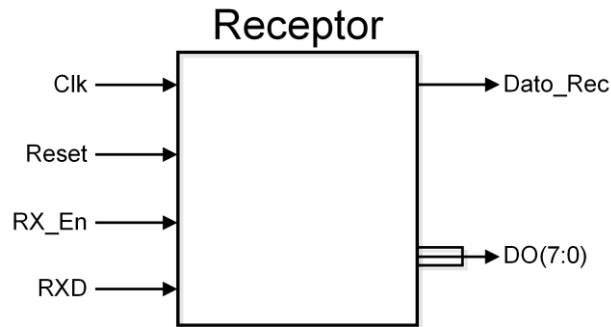


Figura 4.24 Diagrama de bloque del receptor de la TRCM.

El receptor recibe las tramas “Backward”, que constan de 11 bits, tal y como se muestra en la Figura 4.25. Para asegurar que la recepción sea correcta, el receptor muestrea los bits en el punto medio de cada primer medio bit. Haciéndolo así, se realiza al mismo tiempo la decodificación del dato Manchester recibido, ya que el valor de cada bit será el complemento del valor leído en la entrada de recepción. Para sincronizarse con el punto medio del pulso a nivel bajo del bit de inicio, se monitoriza la transición inicial y se espera un cuarto del tiempo de bit, es decir, $208,33 \mu\text{s}$. Una vez sincronizada la recepción, sólo se debe muestrear la entrada de recepción cada *tiempo de bit*. Para contar este tiempo con una precisión adecuada, se utilizó como base de tiempo un reloj con una frecuencia 32 veces superior a la de recepción.

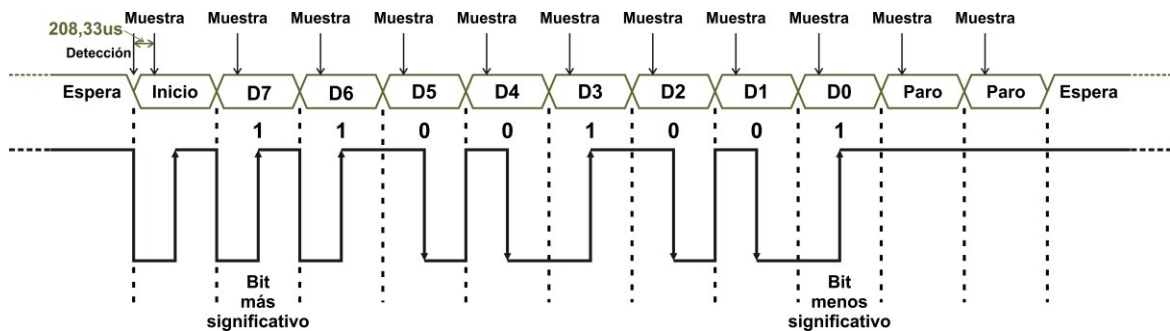


Figura 4.25 Muestreo de los datos de recepción en codificación Manchester.

La ruta de datos del receptor del TRCM se muestra en la Figura 4.26. Está constituida de un registro *buffer* de recepción (Reg_Buffer), un registro serie (Reg_Serial), un registro divisor de frecuencia (Reg_Divide), un registro de muestreo (Reg_Muestra) y un registro contador (Reg_Cuenta). El registro serie realiza las operaciones de carga serie y desplazamiento a la izquierda para generar la conversión de serie a paralelo. El registro *buffer* almacena el dato recibido, por lo que solo se realiza la operación de carga paralela.

El registro divisor de frecuencia genera la señal de reloj (*Tic32*) que se utiliza como base de tiempos para sincronizar la recepción. El registro de muestreo determina el instante de captura del bit recibido a partir de *Tic32*. El registro contador contabiliza los bits que se reciben.

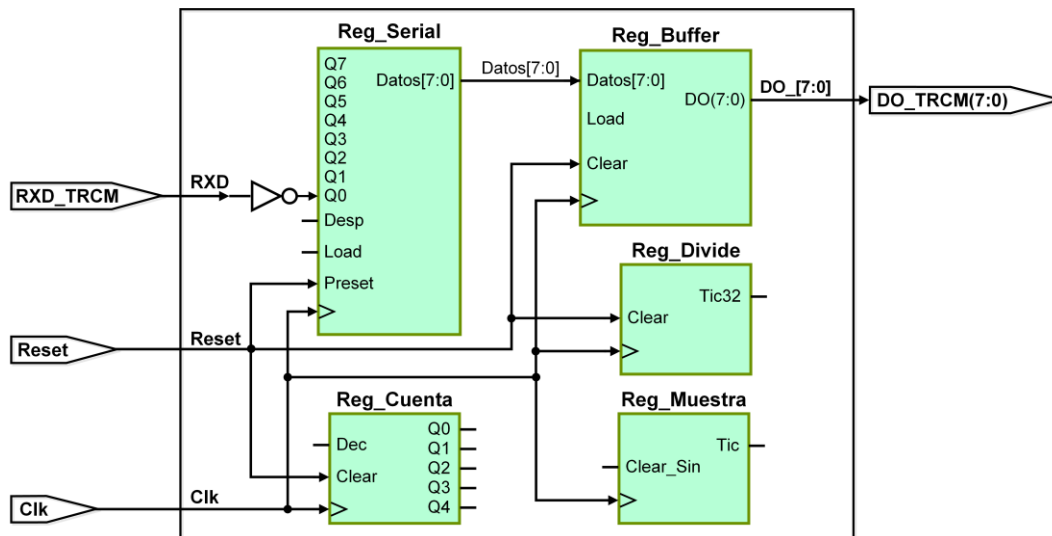


Figura 4.26 Ruta de datos de la FSM del receptor del TRCM.

La descripción del receptor se ha realizado mediante una FSM, cuyo diagrama ASM de la unidad de control se muestra en la Figura 4.27. En el estado “Inicio” se inicializa con cero el registro generador de baudios y el registro contador con el valor adecuado para monitorizar el final de la recepción de los bits de datos. En este estado se espera recibir el bit de inicio para comenzar con el proceso de recepción. Este se inicia si la entrada de habilitación de la recepción, *Rx_En_TRCM* está activa (‘1’) y la entrada de recepción de datos *RXD_TRCM* tiene el valor cero (‘0’ – bit de inicio). Si se cumple esta doble condición, se realiza el proceso indicado anteriormente, y que se puede ver en el diagrama ASM de la FSM. Se pasa al estado “Espera_1_4” en el que se espera un cuarto del tiempo de bit para sincronizarse con el bit de inicio. Para temporizar este tiempo, se utiliza el registro de muestreo, que se incrementa cada vez que se activa *Tic32*. Si el registro de muestreo alcanza la cuenta de siete se inicializa con el valor de cero y se pasa al estado “Desplaza” para recibir los ocho bits de datos. Como se indicó, se debe esperar un tiempo de bit y cargar en el registro serie el complemento del valor de la entrada de recepción. Para ello, cada vez que el registro de muestreo alcance el valor de treinta y uno, se realiza un desplazamiento a la izquierda en el registro serie cargando el complemento del valor de la

entrada de recepción y se decrementa el registro contador. Esta secuencia se repite en el estado “Desplaza” hasta que el registro contador alcance la cuenta de cero, que indica que sólo falta recibir dos bits de *stop*. Para ello, se utilizan los estados “Stop_1” y “Stop_2” en cada uno de los cuales se espera un tiempo de bit para validarlos, y se pasa al estado “Carga” en el que se transfiere el contenido del registro serie al registro *buffer* de recepción, y se activa la señal de salida Dato_rec_TRCM para indicar al procesador DALI que se acaba de recibir una trama “Backward” de la interfaz DALI.

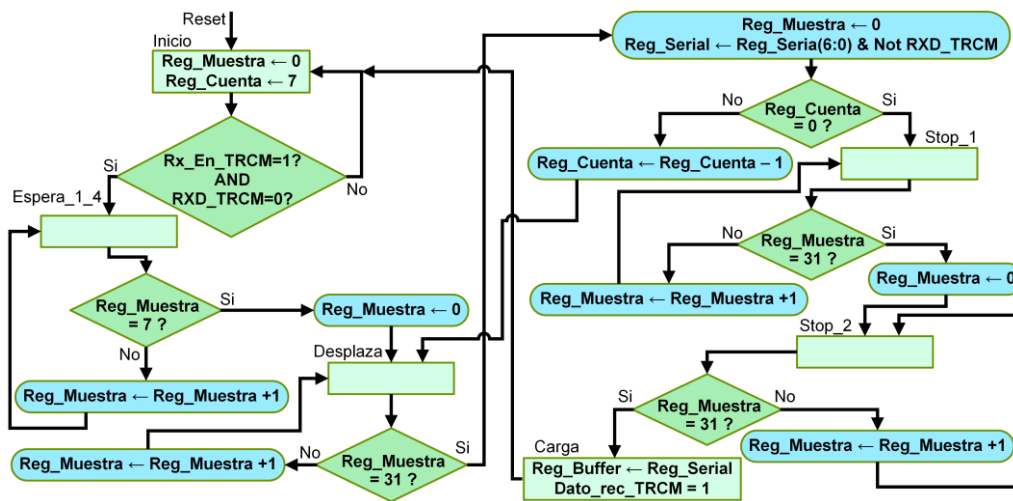


Figura 4.27 Unidad de control de la FSM del receptor del TRCM.

El *bridge* DALI soporta la comunicación con las luminarias mejorando el rendimiento del nodo sensor inalámbrico al liberar al MCU de trabajo y cálculos computacionales relacionados con la interfaz DALI. Además, desarrollar y describir una interfaz de comunicaciones tipo UART, facilita la conexión a cualquier dispositivo y con el nodo sensor inalámbrico. La descripción portable desarrollada en VHDL-93 puede ser sintetizada con cualquier herramienta compatible con el estándar IEEE 1076-93. Así mismo, tiene la flexibilidad para soportar actualizaciones y reconfiguración. Por tanto, el diseño y la descripción portable del *bridge* DALI implementado por HW, en concreto en una FPGA de bajo coste y bajo consumo de energía de cualquier fabricante, permite la ejecución en tiempo real del protocolo DALI.

Capítulo 5. Resultados

En este capítulo se muestran los resultados experimentales de la implementación del *bridge* DALI en la FPGA embebida en un nodo sensor inalámbrico para control de luminarias que utilizan protocolo DALI en aplicaciones de SSL. Además, se resaltan los aspectos relevantes de los diferentes elementos y dispositivos necesarios para crear el nodo sensor inalámbrico. También se presenta la síntesis del diseño, el tiempo de ejecución, la sincronización del *bridge* DALI y de las pruebas eléctricas de los componentes del nodo sensor inalámbrico. La síntesis fue realizada en la plataforma iCEcube2 de Lattice [184] y la implementación se realizó en una FPGA iCE40HX1K-VQ100 [134], obteniendo la cantidad de recursos lógicos y bloques de enrutamiento utilizados, la estimación del consumo de energía y la frecuencia máxima de funcionamiento.

5.1. Descripción del nodo sensor inalámbrico UCODALIB

En esta sección se muestran las consideraciones fundamentales del HW y las características electrónicas de los componentes para el desarrollo de un nodo sensor inalámbrico aplicado al control de iluminación. El diagrama de bloques del nodo inalámbrico se muestra en la Figura 5.1. Al nodo sensor inalámbrico se le denominó UCODALIB. En el diseño del nodo sensor inalámbrico se tiene como el elemento clave la FPGA que permitió el desarrollo del *bridge* DALI. En dicha figura, también se observan los bloques de la Memoria Flash Serie (Serial Flash Memory (SFM)), de la interfaz eléctrica DALI, del conector para el MCU, de la fuente de alimentación de energía eléctrica, del identificador del nodo sensor inalámbrico (ID), de la interfaz con el usuario y de los elementos auxiliares. Por otra parte, también se muestra el circuito oscilador y el conector *Joint Test Action Group* (JTAG, por sus siglas en inglés) usado en la programación de la FPGA. Uno de los objetivos del diseño del nodo sensor inalámbrico fue integrar todos los componentes necesarios dentro de una sola PCB. Así como también, sintetizar la

descripción del protocolo DALI como un *bridge*, para eliminar los problemas identificados con la implementación por SW en otro nodo de similares características implementado anteriormente y comentados en el capítulo anterior.

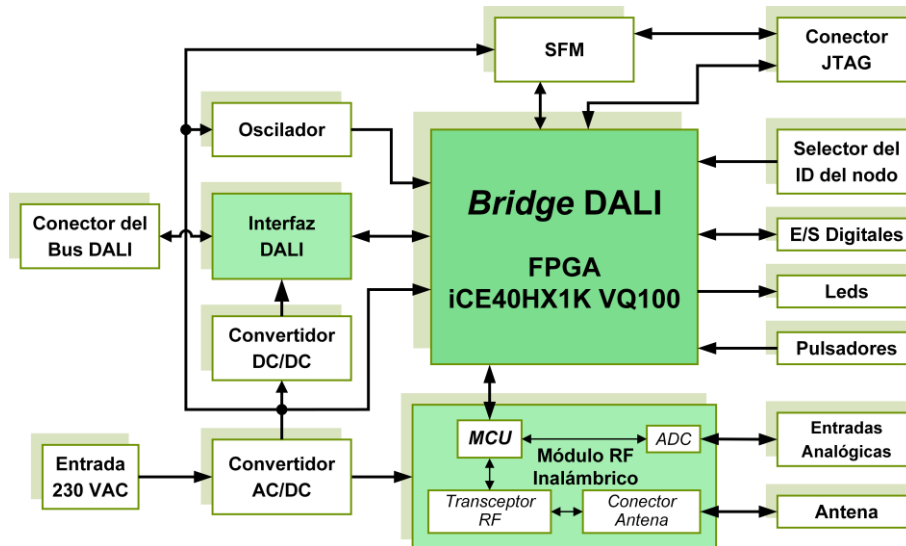


Figura 5.1 Diagrama de bloques del nodo sensor inalámbrico UCODALIB.

Una fotografía del nodo donde se ha implementado el *bridge* DALI se muestra en la Figura 5.2. El elemento para la conexión inalámbrica no está soldado sobre la misma PCB, sino que se ha incluido un conector de expansión genérico que permite conectar cualquier MCU, módulo RF o SoC.

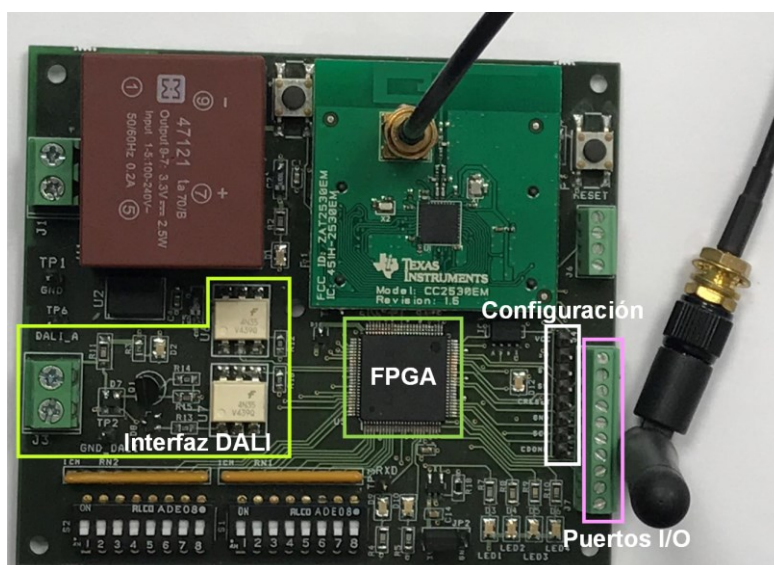


Figura 5.2 Fotografía del nodo sensor inalámbrico UCODALIB.

5.1.1. FPGA

La FPGA que se utilizó en el nodo sensor inalámbrico es la iCE40HX1K-VQ100 de la empresa Lattice Semiconductor. La FPGA tiene el propósito de implementar la descripción del protocolo DALI. Para la selección de la iCE40HX1K-VQ100 se consideró que es una FPGA de ultra bajo consumo de potencia, alto rendimiento y bajo coste. En su arquitectura tiene lógica programable de bajo coste basada en LUTs, aunque también cuenta con elementos específicos como Bloques Embebidos de Memoria RAM (Embedded Block RAM (EBR), en inglés), Memoria Configurable No Volátil (Non-volatile Configuration Memory (NVCM), en inglés) y Lazo de Seguimiento de Fase (Phase Locked Loops (PLLs), en inglés).

Las FPGAs iCE40 LP/HX se fabrican con tecnología CMOS de 40 nm de baja potencia. La arquitectura de la FPGA presenta diversas ventajas como: entradas con disparador Schmitt con una histéresis de 200 mV, esto permite conectar entradas que tienen un *slew rate* lento como, por ejemplo, las salidas de los optoacopladores que se usan en la interfaz DALI. Los chips de este tipo de FPGAs están fabricados en diferentes encapsulados, lo que permite disminuir el área en el diseño de las PCBs. Las especificaciones de la familia iCE40 LP/HX que se están usando en este trabajo son:

- La arquitectura flexible en la lógica para la descripción de HW.
- El consumo de ultra baja energía de 21 μ W en reposo.
- La memoria embebida y distribuida, tipo sysMEM EBR, para almacenar la configuración de la FPGA.
- Las E/S síncronas para el pre-diseño de registros en celdas DDR.
- Los búferes de E/S flexibles y de alto rendimiento.
- El administrador de reloj flexible en el chip.
- La configuración flexible de la SRAM en el dispositivo través de la interfaz SPI estándar

La herramienta de desarrollo iCEcube2 de Lattice Semiconductor se utiliza para sintetizar la descripción del diseño. La familia de estas FPGAs de tipo SRAM son reprogramables y admiten distintos métodos de configuración como: maestro SPI o esclavo SPI. Estas FPGAs también pueden ser configuradas desde una memoria flash EPROM con bus SPI o por un maestro externo como una CPU [134]. En el diseño del nodo sensor inalámbrico la configuración de la FPGA se realiza mediante la memoria flash EPROM, de

forma que la propia FPGA al aplicarle la tensión de alimentación, transfiere el contenido de la memoria flash EPROM a su SRAM interna de configuración.

Algunas características que se consideraron para la selección de la FPGA iCE40HX1K-VQ100 son la cantidad de 1,280 LUTs, el tamaño del encapsulado que está en proporción a los recursos lógicos y al coste, el número de las patillas de I/Os y otros parámetros claves, mostrados en la Tabla 5.1, que cumplen con las necesidades del proyecto. Además, los CLBs que constituyen el *Core* de la iCE40 LP/HX pueden programarse para realizar funciones lógicas y aritméticas.

Tabla 5.1 Información de la FPGA iCE40 -VQ100 de Lattice Semiconductor.

LCs (LUT + flip-flop)	1280
Bloques de memoria RAM4K	16
RAM4K	64Kb
PLLs	11
Máximo número de pines programables de I/O	95
Máximo número de pares de entradas diferenciales	11
Empaquetado	100 VQFP (14 mm x 14 mm, 0,5 mm) 100-Pin VQFP (paso de 0,5 mm)

La FPGA cuenta con Entradas y Salidas Programables (Programmable I/O (PIO), en inglés). Las celdas de las PIOs se encuentran alrededor del dispositivo, dispuestas en bancos. Las PIOs utilizan un búfer flexible de I/O, denominado búfer sysI/O, que se puede configurar con diferentes interfaces. Los cuatro bancos sysI/O de la FPGA son aprovechados en este diseño para conectar con los dispositivos bidireccionales que son: la interfaz DALI, el módulo de RF y el conector del puerto de I/Os. Además, los puertos se configuraron de entrada para los micro-interruptores y de salida para los leds (ver Figura 5.1).

La configuración de la FPGA se define mediante la carga de datos de una aplicación específica en las celdas de su memoria SRAM interna, lo que significa que el diseño es volátil. Sin embargo, la FPGA posee una memoria no volátil NVCM, pero con el inconveniente de que solo puede ser programada una sola vez. En el diseño del nodo sensor inalámbrico, se consideró lo anterior, y se incorporó una SFM externa para tener la ventaja de configurar la FPGA desde la memoria externa o desde una CPU. Estas dos formas de configuración requieren el conjunto de pines SPI de la FPGA.

5.1.2. Memoria flash EPROM de configuración

Debido a la naturaleza volátil de la FPGA, cuando no esté alimentada por energía eléctrica se perderá la configuración del *bridge* DALI, que está almacenada en su interior, ya que no se ha utilizado la NVCM, por lo que se utilizó una memoria externa para almacenar los datos de configuración permitiendo la reconfiguración de la FPGA. En este trabajo, se utilizó una SFM de 4 Mb que permite la auto-configuración de la FPGA (maestro SPI) a través de la interfaz SPI. Por lo tanto, permite reprogramar o realizar mejoras futuras que se requieran del diseño del *bridge* DALI. La elección de esta memoria externa se realizó a partir de las restricciones de compatibilidad que presenta la FPGA, que son:

- Rango de la tensión de alimentación entre 1,8 V y 3,3 V.
- Soporta el comando 0x0B para una rápida lectura de la memoria.
- Soporta los comandos 0xB9 y 0xAB para disminuir el consumo de energía eléctrica después de la configuración.
- Soporta la interfaz de configuración master SPI con la FPGA.

El circuito integrado elegido fue el M25P40VP de Micron que es un dispositivo de memoria flash serie con mecanismos avanzados de protección contra escritura a los que accede con un bus compatible con SPI de alta velocidad y sus características se especifican en [185]. Esta memoria cubre todos los requisitos de diseño, ya que admite comandos de alto rendimiento para frecuencias de reloj de hasta 75MHz. La programación de los datos de configuración en la SFM se realiza a través del puerto SPI disponible.

5.1.3. Oscilador

La FPGA necesita de un dispositivo oscilador que genere la señal de reloj, para lo cual se utilizó el oscilador de precisión LTC1799 de Linear Technology [186], con un rango de frecuencia programable que permite generar varias frecuencias de funcionamiento para tener versatilidad en el nodo sensor inalámbrico y, además, reducir el consumo de potencia según los requerimientos que se exijan.

En la Figura 5.3 se muestra el esquema eléctrico del circuito oscilador. La frecuencia de salida se puede configurar dentro del rango de 1 KHz a 33 MHz mediante la resistencia R18, conectada a la patilla de entrada SET, según la ecuación 5.1. La entrada DIV determina

el valor de la variable N, que actúa como divisor de la frecuencia base. El valor se determina mediante el puente JP2, de forma que se tienen los tres casos siguientes:

- Entrada DIV conectada a 3,3 V, N = 100
- Entrada DIV sin conectar, N = 10
- Entrada DIV conectada a GND, N = 1

Considerando una frecuencia de salida de 20 MHz y N igual a 1, se obtiene un valor de 5 KΩ para la resistencia R18, por lo que finalmente se ha seleccionado una resistencia de 4,99 KΩ y una tolerancia del 1%.

$$f_{osc} = 10MHz \cdot \left(\frac{10K}{N \cdot R18}\right), \quad N = \begin{cases} 100 \\ 10 \\ 1 \end{cases} \quad \text{Ecuación 5.1}$$

Por tanto, según se configure la entrada DIV mediante el puente JP2 se selecciona una de las tres frecuencias siguientes:

$$\text{Para } N = 1 \rightarrow f_{osc} = 10MHz \cdot \left(\frac{10K}{1 \cdot 5K}\right) = 20 \text{ MHz}$$

$$\text{Para } N = 10 \rightarrow f_{osc} = 10MHz \cdot \left(\frac{10K}{10 \cdot 5K}\right) = 2 \text{ MHz}$$

$$\text{Para } N = 100 \rightarrow f_{osc} = 10MHz \cdot \left(\frac{10K}{100 \cdot 5K}\right) = 200 \text{ KHz}$$

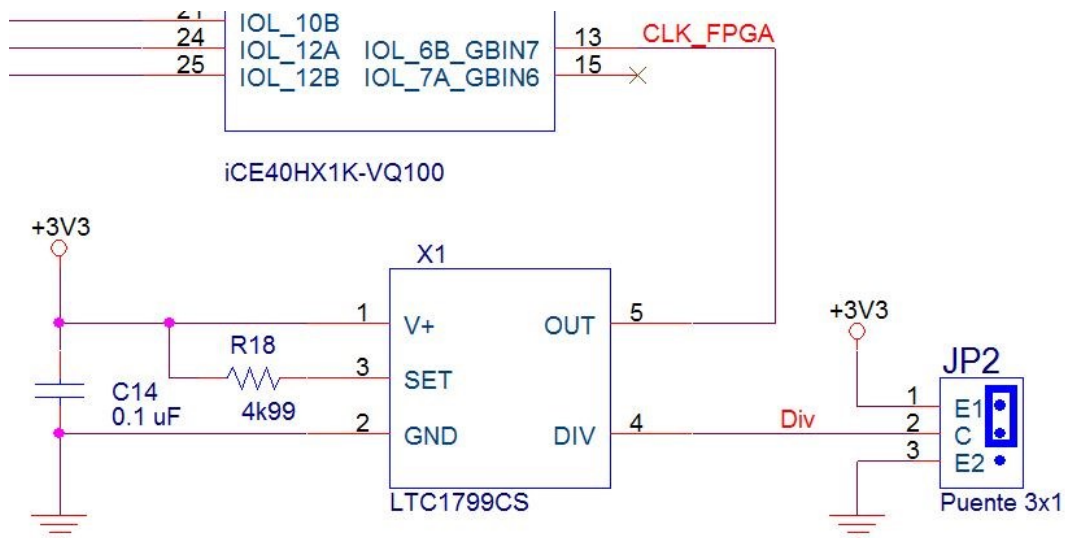


Figura 5.3 Esquema eléctrico de la configuración del oscilador de frecuencia.

5.1.4. Módulo SoC de radiofrecuencia

El nodo sensor inalámbrico requiere de un módulo SOC con enlace de RF para realizar la comunicación inalámbrica y la conexión con la FPGA que implementa el *bridge* DALI. El módulo SOC de RF seleccionado integra un transmisor/receptor y un MCU en un mismo dispositivo para realizar la conectividad inalámbrica. Las características de la velocidad de transmisión y el campo de cobertura dependen del protocolo de comunicación. En este proyecto se determinó utilizar el estándar 802.15.4/Zigbee para la comunicación inalámbrica entre los nodos sensores inalámbricos que formarán la WSN.

El módulo RF que se utilizó para realizar la comunicación inalámbrica en el nodo sensor inalámbrico es el CC2530EM [187], el cual se muestra en la Figura 5.4. Este es un SoC que permite implementar el estándar IEEE 802.15.4. El módulo consta de un chip CC2530F256 [187], dos cristales de cuarzo, de 32 MHz y 32 KHz; un acoplador para antena SMA (SubMiniature versión A, por sus siglas en inglés) y el circuito de ajuste de impedancia de RF [188]. El chip CC2530F256 tiene un MCU 8051, un coprocesador de seguridad AES, memoria flash programable de 256 KB y RAM de 8 KB y un transceptor RF compatible con el estándar IEEE 802.15.4 [187]. El MCU 8051 ejecuta una instrucción por periodo de reloj, porque posee tres diferentes buses de acceso a memorias, un núcleo de alto rendimiento y bajo consumo de energía para realizar búsqueda previa de código y, además, soporta depuración HW. Este módulo permite una solución completa y robusta para la comunicación inalámbrica en sistemas de iluminación.

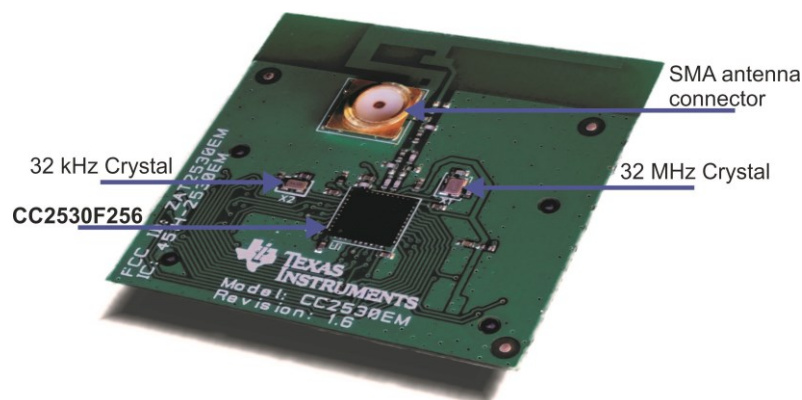


Figura 5.4 Módulo RF [188].

Las características con las que cuenta el módulo RF son:

- Permite el desarrollo del diseño de forma modular.

- Utilizar bandas de radio libres y sin necesidad de licencia.
- Instalación barata y simple.
- Desacoplamiento de RF.
- Soporta conexiones punto a punto y punto a multipunto.
- Soporta el direccionamiento de información de la red.
- Opera en la banda libre de ISM 2,4 GHz para conexiones inalámbricas.
- Soporta redes de baja tasa de transferencia de datos.
- Reduce los tiempos de espera en el envío y recepción de paquetes.
- Ciclo de trabajo reducido para una larga duración de la batería.
- Soporta topologías de red múltiples, por ejemplo, dinámica, estática, malla y estrella.
- Soporta un número elevado de nodos en una red.
- Soporta 128-bit Estándar Avanzado de Encriptado (Advanced Encryption Standard (AES), en inglés) de cifrado para realizar conexiones seguras.

El nodo sensor inalámbrico también cuenta con un conector de canales analógicos de entrada. Aprovechando las características del módulo de RF CC2530EM que cuenta con ADCs de 12 bits, con la posibilidad de que los convertidores funcionen con entrada diferencial, se habilitaron para su uso 3 canales de entrada. Para lo cual, en la PCB se añadió un conector de cuatro terminales de tornillo con ajuste de presión donde se pueden conectar de forma externa las señales analógicas.

El nodo sensor inalámbrico diseñado posee una ranura de expansión formada por dos conectores para la inserción del módulo RF. De esta manera, se evita que el dispositivo deba ser soldado directamente a la PCB y así poderlo extraer fácilmente en caso de rotura o sustitución. Además, admite la integración de un módulo RF que cumpla o mejore las características señaladas, para mejoras futuras.

5.1.5. Interfaz DALI

La función del bloque de la interfaz DALI es proporcionar una interfaz de comunicación con la luminaria a controlar, adaptando los niveles lógicos de voltaje de la FPGA a los del bus DALI, además de realizar el aislamiento del bus del resto de componentes de la placa. El bus DALI está localizado en la PCB del nodo sensor inalámbrico, en un conector de dos terminales de tipo tornillo. La diferencia de voltaje entre

ambas señales del bus determina los niveles lógicos, como se puede observar en la Figura 5.5. Para el nivel lógico alto, cuando la luminaria o el nodo inalámbrico funcionan como receptor, el rango de voltaje es entre 9,5 V y 22,5 V, y cuando funcionan como transmisor, el rango es de 11,5 V a 20,5 V. En contraste, para el nivel lógico bajo, cuando la luminaria o el nodo inalámbrico funcionan como receptor, los niveles de voltaje varían de -6,5 V a +6,5 V, y cuando funciona como transmisor van de -4,5 V a +4,5 V.

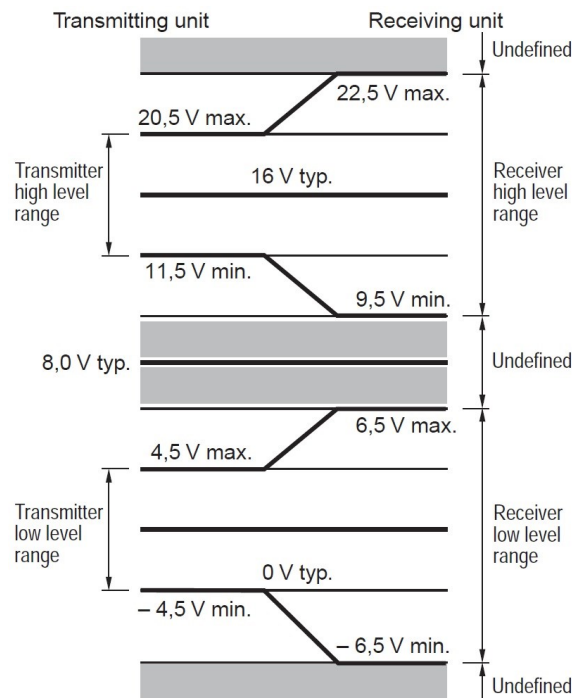


Figura 5.5 Niveles lógicos DALI [114].

La interfaz DALI acepta comunicación bidireccional y proporciona los niveles lógicos altos y bajos apropiados para el proceso de transmisión/recepción de tramas DALI, en el cual se emplearon optoacopladores para aislar las masas de la interfaz DALI y la del *bridge* y generar los niveles de tensión adecuados. Así como también, con el objetivo de proteger de cambios de polaridad de la interfaz DALI, se incluyó un puente de diodos Schottky para garantizar una baja caída de tensión directa y una velocidad de conmutación rápida, evitando la rotura del dispositivo.

En el nodo sensor inalámbrico se realizó un diseño propio de la interfaz DALI para el control de la luminaria, teniendo en consideración el soportar comunicación bidireccional además con la ventaja de tener integrada una fuente de alimentación de bus DALI y estar protegida contra cambios de polaridad. En la Figura 5.6 se muestra el esquema eléctrico de

la interfaz DALI. Se utilizaron dos optoacopladores con salidas de fototransistor 4N36 (U6 y U7) los cuales permitieron aislar los niveles de tensión tanto para la transmisión como para la recepción del control de la luminaria. El diseño permite el acoplamiento para que un nivel lógico bajo en el pin de la FPGA corresponda a un nivel lógico bajo en el bus DALI, y de la misma forma para un nivel lógico alto. De esta forma, la comunicación de los bits de las tramas se realiza de manera íntegra en la transmisión y en la recepción.

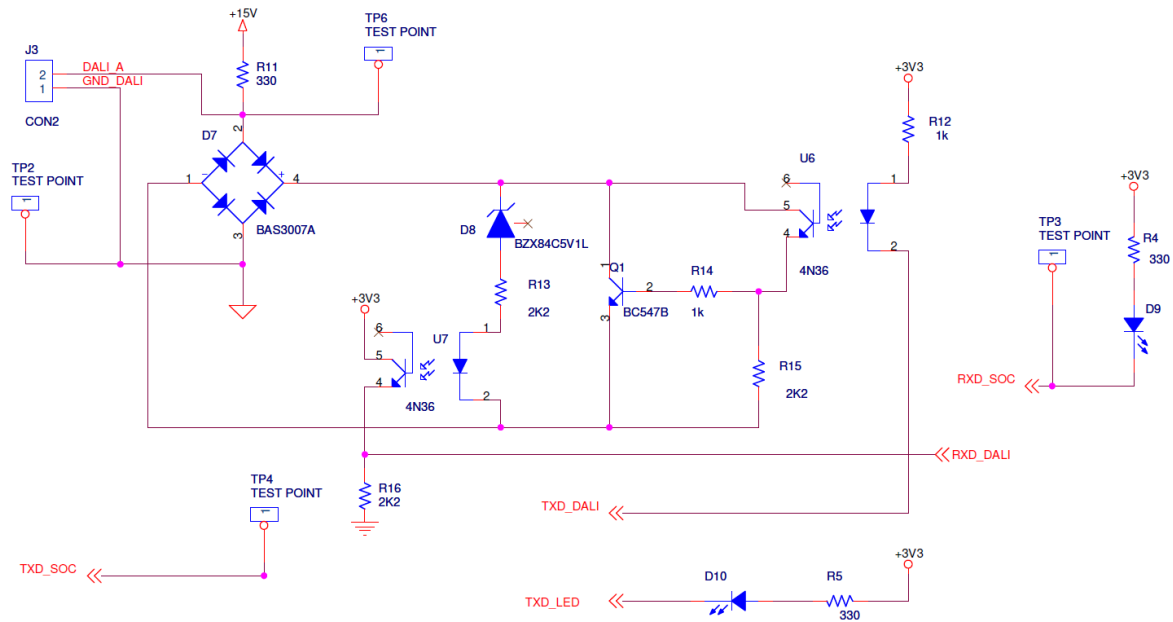


Figura 5.6 Esquema eléctrico de la interfaz DALI.

5.1.6. Fuente de alimentación de energía eléctrica

La mayoría de los componentes del nodo sensor inalámbrico necesitan una fuente de energía eléctrica de 3,3 V DC para funcionar, para lo cual se emplea el módulo FC47121 que integra una fuente de alimentación conmutada [189], cuyo esquema eléctrico se muestra en la Figura 5.7. Este convertidor genera a partir de la tensión de la red 230 VAC una tensión de salida de 3,3 V DC. El FC47121 está basado en topología *flyback*, por lo que es idóneo para aplicaciones de baja potencia, como una opción diferente a las tradicionales fuentes de energía eléctrica. Tiene las ventajas de contar con precisión $\pm 2\%$ del voltaje de salida al 100% de carga, 2,5 W de potencia y 65% de eficiencia [189].

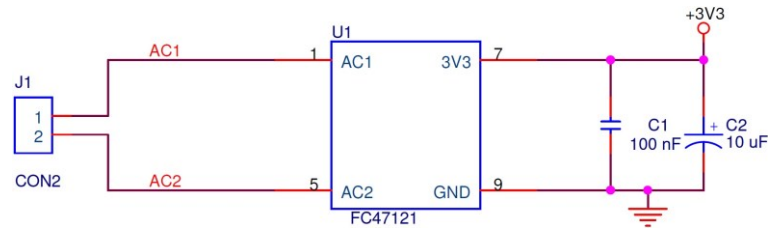


Figura 5.7 Esquema eléctrico para fuente de voltaje de 3,3 V.

Las FPGAs actuales necesitan varias tensiones de energía eléctrica, en concreto los IOBs utilizan una tensión de 3,3 V DC, y el funcionamiento de su *Core* interno requiere una tensión inferior, generalmente del orden de 1,2 V DC. Por otra parte, también necesitan otra tensión diferente para su programación, que es una tensión intermedia entre las dos anteriores. Estas dos tensiones de energía eléctrica propias de la FPGA se pueden obtener utilizando dos Reguladores Lineales (Low Dropout (LDO), en inglés), dado el bajo consumo de potencia del *bridge* DALI. Para la programación de la NVCM, si fuese necesario, es necesaria una tensión de 2,5 V. Para generarla se ha usado el circuito recomendado por el fabricante, dado su simplicidad. Como se puede observar en la Figura 5.8, simplemente se usa un diodo *Schottky* (SB05-03Q de SANYO) para obtener una tensión aproximadamente de 3 V a partir de la tensión principal de 3,3 V. Teniendo en cuenta que la intensidad es de 2,5 mA, la caída de tensión en el diodo corresponde a 0,28 V, por lo cual el voltaje de salida es: $V_{out} = 3,3 - 0,28 = 3,02$ V. El otro voltaje es de 1,2 V, que se utiliza para alimentar de energía eléctrica a los elementos lógicos de la FPGA, y se obtiene utilizando un regulador AP1122 de baja caída de voltaje, como se muestra en el esquema eléctrico de la Figura 5.9.

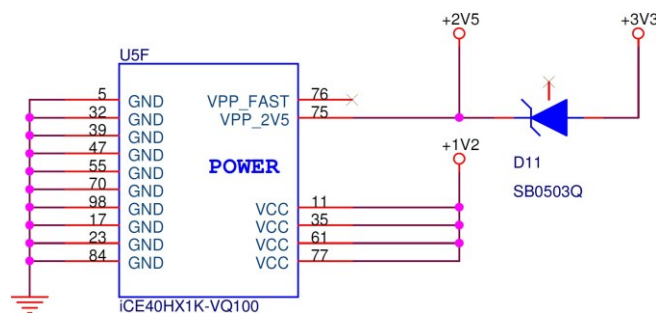


Figura 5.8 Esquema eléctrico para generar el voltaje auxiliar de la FPGA.

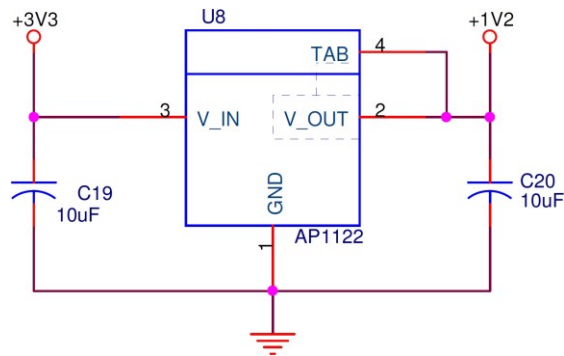


Figura 5.9 Esquema eléctrico para generar el voltaje de los elementos lógicos de la FPGA.

Por otro lado, se utilizó el módulo híbrido MEU1S0315ZC de la empresa Murata Power Solutions [190] para producir una tensión de energía eléctrica de 15 V DC para la interfaz DALI, y generar los niveles de voltaje adecuados para controlar la luminaria, como se muestra en la Figura 5.10. Este módulo es un convertidor DC/DC de 1W con aislamiento galvánico y una muy alta eficiencia, que garantiza el aislamiento necesario entre el segmento de control de la interfaz DALI y resto de los componentes del nodo sensor inalámbrico. El MEU1S0315ZC es ideal para alimentar el sistema de control de la interfaz DALI, consiguiendo un aislamiento galvánico de 1 kVDC, y permite obtener el nivel lógico alto requerido por la interfaz DALI, ya que el voltaje de salida es de 15 VDC con una eficiencia del 81% a partir de un voltaje de entrada de 3,3 V DC. Otra ventaja de este dispositivo es que su encapsulado ultra-miniatura, estilo ZIP, ha permitido reducir el área de la PCB, además del coste total en comparación con versiones anteriores del nodo sensor inalámbrico. Otra mejora ha sido el diseño de una PCB de 4 capas frente a las de 2 de las versiones anteriores. Esto ha permitido usar una capa para las alimentaciones y otra para las masas, por lo que se ha podido reducir el número de pistas y realizar un mejor desacoplo de la tensión de alimentación de los circuitos integrados. Por otra parte, se han podido hacer un plano de masa en las capas para cada una de las alimentaciones, indicadas anteriormente, por lo que todo ello ha permitido realizar un diseño del sistema más fiable e inmune al ruido eléctrico.

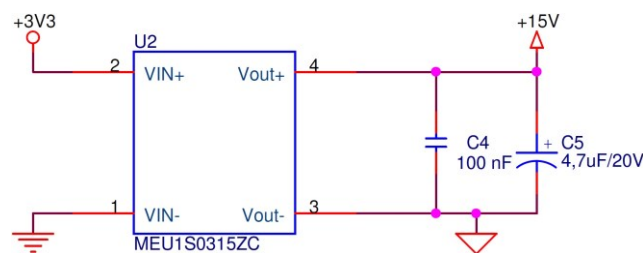


Figura 5.10 Esquema eléctrico para fuente de voltaje de 15 V.

5.1.7. Identificador del nodo, interfaz con el usuario y elementos auxiliares

En el diseño del nodo sensor inalámbrico se han incluido algunos elementos auxiliares que sirven de interfaz con el operador o usuario final.

Se incluyó un selector para el ID, que es el valor que se le devuelve al *host* cuando se ejecuta el comando 02 del *bridge* DALI. Para configurar y leer el ID del nodo de 16 bits se ha utilizado dos microinterruptores encapsulados de ocho líneas cada uno y dos multiplexores binarios de ocho a uno. Los multiplexores binarios se han incluido dentro de la FPGA y todo el proceso de control es compatible con las versiones anteriores, en las que se usaban dos multiplexores integrados 74VHCT151.

La PCB cuenta con leds para monitorear visualmente las comunicaciones y leds de propósito general. Dos leds permiten comprobar la existencia de las tensiones de 3,3 V y 15 V. Para observar el comportamiento de la interfaz DALI se han usado dos leds conectados a la señal de recepción y a la de transmisión. También se incluyó un led para identificar el estado de la FPGA, indicando el momento en el que se ha terminado la configuración. El nodo sensor inalámbrico también dispone de dos pulsadores de botón, uno para inicializar el sistema y otro de propósito general.

El nodo sensor inalámbrico tiene un puerto de E/S de 8 bits de propósito general implementado en la FPGA que pueden ser configurados de entrada o salida. Este puerto es accesible en la PCB a través de un conector de tipo tornillo.

La PCB incluye un conector JTAG que se utiliza para la configuración de la FPGA. Este cumple con el estándar IEEE 1149.1, y se adapta a las especificaciones del fabricante de la FPGA. Se incluyó este conector por ser un método muy utilizado para la configuración de las FPGAs.

5.2. Resultados de la implementación del *bridge* DALI en el nodo sensor inalámbrico

A continuación, se muestran los resultados de la síntesis del diseño del *bridge* DALI en términos de los recursos lógicos utilizados, la latencia, el consumo de potencia estimado y el tiempo de bit de la interfaz DALI. La síntesis fue realizada para la FPGA iCE40HX1K-VQ100 mediante la herramienta iCEcube2 de Lattice Semiconductor.

5.2.1. Recursos lógicos utilizados

La descripción en VHDL del diseño del *bridge* DALI es portable y se realizó buscando optimizar el consumo de recursos lógicos, de forma tal que aproximadamente más del 80 % de LCs y del 90 % de los recursos de enrutamiento quedan libres. En la Figura 5.11 se observa que el diseño consume aproximadamente el 11,48% de las LC combinacionales y 19,14% de las LC secuenciales. La cantidad de bloques de interconexión de la FPGA utilizados para el enrutamiento, como son los circuitos de conmutación, los *buffers* y los segmentos de interconexión, se muestran en la Figura 5.12. Por otra parte, no utiliza ninguno de los BRAM de la FPGA, por lo que se podrán utilizar para mejorar la funcionalidad del *bridge* DALI o para implementar otras interfaces.

```

Device Info:
-----
Device Family: iCE40
Device:       iCE40HX1K
Package:     VQ100

Design statistics:
-----
FFs: 245
LUTs: 387
RAMs: 0
IOBs: 34
GBs: 4
PLLs: 0

Logic Resource Utilization:
-----
Total Logic Cells: 392/1280
  Combinational Logic Cells: 147    out of 1280    11.4844%
  Sequential Logic Cells: 245      out of 1280    19.1406%
  Logic Tiles: 72                  out of 160     45%
Registers:
  Logic Registers: 245              out of 1280    19.1406%
  IO Registers: 0                   out of 560     0%
Block RAMs: 0                      out of 16      0%
Pins:
  Input Pins: 24                    out of 72      33.33333%
  Output Pins: 10                   out of 72      13.88889%
  InOut Pins: 0                    out of 72      0%
Global Buffers: 4                   out of 8       50%
PLLs: 0                             out of 0

```

Figura 5.11 Resultados de la síntesis en términos de la utilización de recursos lógicos.

Routing Resource Utilization:

Local line of tile	2606 out of	28666	9.09091%
Span 4	339 out of	6944	4.88191%
Span 12	46 out of	1440	3.19444%
Global network	4 out of	8	50%
Vertical Inter-LUT Connect	60 out of	1120	5.35714%

Figura 5.12 Resultados de recursos ubicados para el enrutamiento.

5.2.2. Latencia

Un parámetro muy importante en cualquier diseño de cualquier *bridge* es el tiempo de latencia. Es cierto que el flujo de datos a través del *bridge* DALI es muy pequeño y las velocidades de las interfaces de ambos lados son muy bajas, por lo que el tiempo de latencia no provoca un malfuncionamiento del *bridge*. Sin embargo, minimizar el tiempo de latencia es importante para no añadir retardos innecesarios. Para medir el tiempo de latencia se simuló la ejecución del comando DALI número 04 de la Tabla 4.1, funcionando a una frecuencia de 20 MHz. Se han realizado dos tipos de medidas de latencia del *bridge*. Por una parte, se ha medido la *latencia real* del *bridge* DALI, que es el tiempo que transcurre desde que termina de recibir el comando del *host* y comienza a enviarlo a la interfaz DALI. Este tiempo es muy pequeño, en concreto 300 ns, tal y como se puede apreciar en la Figura 5.13. Por otra parte, se midió la latencia, desde que el *host* envía el comando al *bridge* DALI y éste responde enviando la trama correspondiente a la interfaz DALI. Esta tiene un valor 3,38 ms, como se indica en la Figura 5.14.

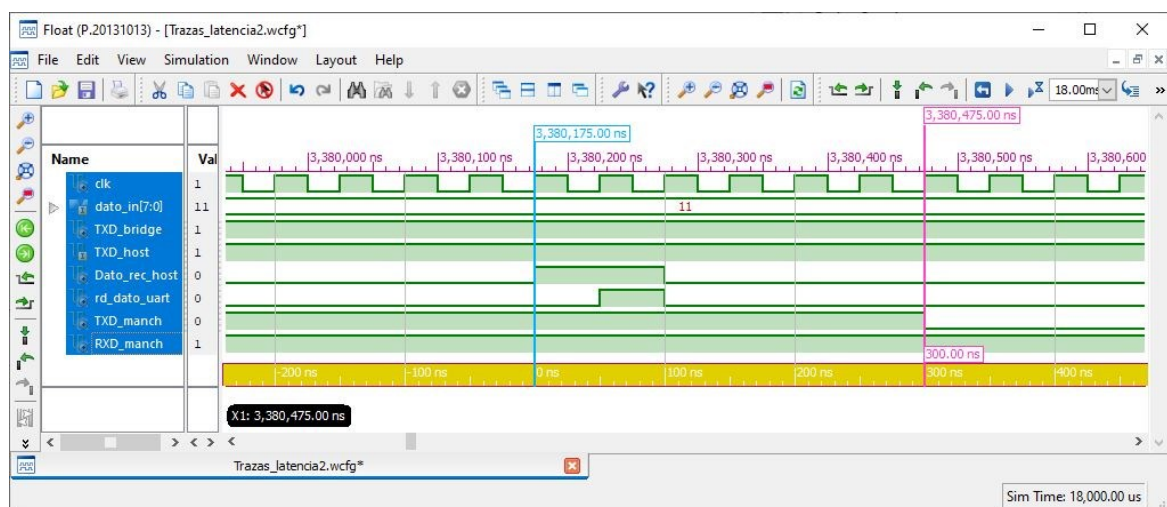


Figura 5.13 Tiempo de latencia real del *bridge* DALI

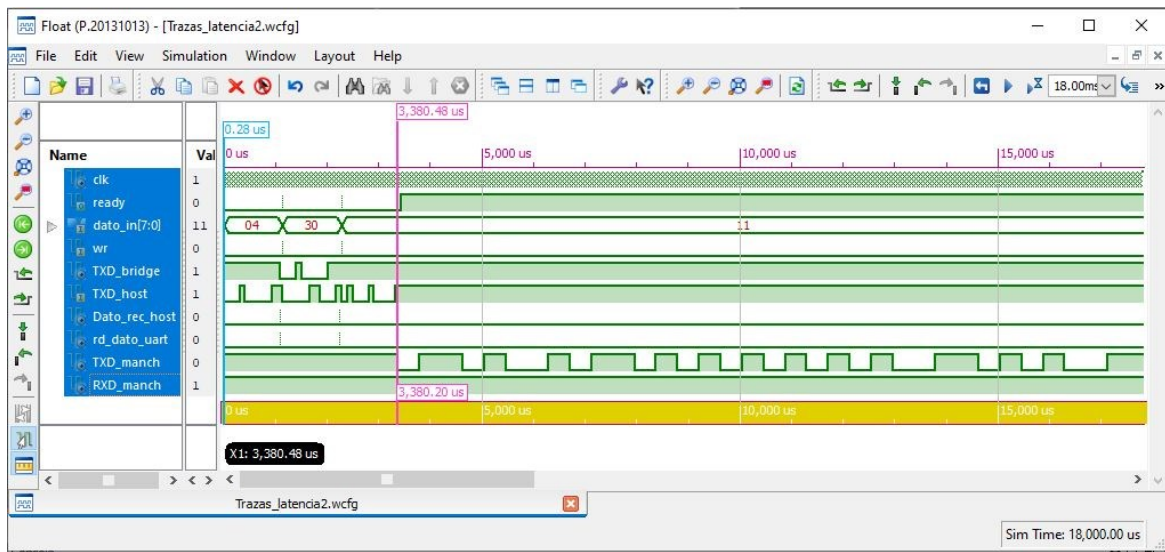


Figura 5.14 Tiempo de respuesta del *bridge* DALI incluyendo el tiempo de comunicación.

Cabe comentar que el último tiempo de latencia medido está determinado por la velocidad de transmisión, que en este caso es de 9600 bps, por lo que se puede reducir fácilmente, ya que el *bridge* DALI se puede configurar con las velocidades más altas permitidas por los MCUs. Con esto se puede afirmar que la ejecución de comandos en el *bridge* DALI es más rápida que en otros trabajos reportados, como por ejemplo en [161] donde las tramas se envían con un retraso de 0,1 segundo, o en el caso de [41] donde emplean *timers* de alta resolución para ejecutar tareas que requiere una resolución mayor a 1ms.

5.2.3. Frecuencia de operación

Se realizó el análisis de tiempos en la síntesis del *bridge* DALI para obtener la frecuencia máxima de la señal de reloj, que se muestra en la Figura 5.15. En cuanto a operaciones internas en la FPGA se observa que la implementación soporta una frecuencia de funcionamiento *Target* de 122,40 MHz, aproximadamente cinco veces la frecuencia del oscilador programable que contiene el nodo sensor inalámbrico. En el cálculo, el sintetizador tiene en cuenta la frecuencia de operación de todos los recursos lógicos individuales (celdas de I/O, LUTs, flip-flops y BRAM), las frecuencias de conmutación de los flip-flops de las celdas lógicas y los retardos de las celdas lógicas combinatoriales (LUT). El soportar esta frecuencia de funcionamiento permite que, en caso de necesitar

módulos adicionales e independientes que requieran operar a frecuencias mayores, se puedan implementar en la misma FPGA.

```
#####
1::Clock Frequency Summary
=====
Number of clocks: 1
Clock: Conjunto_FSM_Ctrl_UARTpc_RxTxManch|clk|Frequency: 168.74 MHz|Target: 122.40 MHz|
=====
End of Clock Frequency Summary
#####
```

Figura 5.15 Resultado de frecuencia de reloj.

5.2.4. Estimación de consumo de potencia

Un factor importante es determinar el consumo de potencia empleada por el *bridge* DALI. El desglose de la disipación de potencia estimada, en función de la frecuencia y la temperatura, se muestra en la Tabla 5.2. La estimación se realizó para una tensión de alimentación de energía eléctrica de los bancos de E/S de 3,3 V, ya que éstos se configuran con el estándar LVCMOS33, y una tensión de alimentación de energía eléctrica del *Core* de 1,2 V. Para los valores de frecuencia se han considerado tres casos: la frecuencia máxima de funcionamiento que determina la herramienta de síntesis (122,40 MHz), y dos frecuencias del oscilador configurable de precisión, que se ha utilizado en la implementación del nodo sensor inalámbrico (20 MHz y 2 MHz). De los datos de la tabla se deduce que el consumo de potencia en modo estático es mínimo, ya que en la condición más desfavorable (70 °C) es de sólo 1,41 mW. Por otra parte, se puede ver que el consumo de potencia en modo dinámico no varía con la temperatura, y lógicamente sólo depende de la frecuencia. Por tanto, el consumo total de potencia se mantiene muy estable en todo el rango de temperatura (0 a 70 °C), variando solamente 1,1 mW.

Tabla 5.2 Estimación de potencia.

Dominio de la Frecuencia (MHz)	Temperatura (°C)	Desglose dinámico de potencia (mW)		Consumo de potencia (mW)		
		Potencia en el Core	Potencia en E/S	Potencia estática	Potencia dinámica	Potencia total
122,40	0	12,46	3,54	0,32	16,00	16,32
	25	12,46	3,54	0,44	16,00	16,44
	70	12,46	3,54	1,41	16,00	17,41
20	0	3,39	0,96	0,32	4,35	4,67
	25	3,39	0,96	0,44	4,35	4,79
	70	3,39	0,96	1,41	4,35	5,76
2	0	0,34	0,10	0,32	0,44	0,76
	25	0,34	0,10	0,44	0,44	0,88
	70	0,34	0,10	1,41	0,44	1,84

Por lo que respecta a la variación del consumo de potencia respecto a la frecuencia, hay una diferencia relativa importante de 15,56 mW de promedio, entre la frecuencia máxima (122,40 MHz) y la mínima (2 MHz) a la que podría funcionar el diseño. Otra conclusión que se puede deducir es que, si solo se utilizan dos frecuencias del oscilador embebido en el nodo sensor inalámbrico, la diferencia es de únicamente 3,91 mW de promedio. Por tanto, si se necesita que la FPGA tuviera que realizar algún procesamiento para el que tuviera que funcionar a 20 MHz, el incremento del consumo de potencia total es realmente mínimo.

Este consumo de potencia tan bajo del *bridge* DALI reduce considerablemente el consumo total de energía eléctrica del nodo sensor inalámbrico comparado con otros nodos sensores inalámbricos.

5.2.5. Tiempo de bit

Otro punto importante de las especificaciones del protocolo DALI es que el *bridge* se ajuste a las tolerancias del tiempo de bit. Experimentalmente, se analizó la tasa de transferencia de bits, donde se obtuvo el tiempo medio por bit (T_e) de 416 μ s, y el tiempo de bit de 832 μ s, teniendo solo un 0,16 % de error, valor dentro del rango de tolerancia de error permitido (10%). En la norma se especifica que el tiempo de bit debe ser de 833,33 μ s y el tiempo medio por bit (T_e) debe ser de 416,67 μ s. Lo antes mencionado se muestra en la Figura 5.16. El resultado de la implementación del *bridge* DALI en la FPGA iCE40HX1K-VQ100 de Lattice Semiconductor embebida en el nodo sensor inalámbrico muestra que existe una sincronización en los datos de envío y recepción en codificación Manchester, y están en el rango de tiempo especificado en el estándar del protocolo DALI.

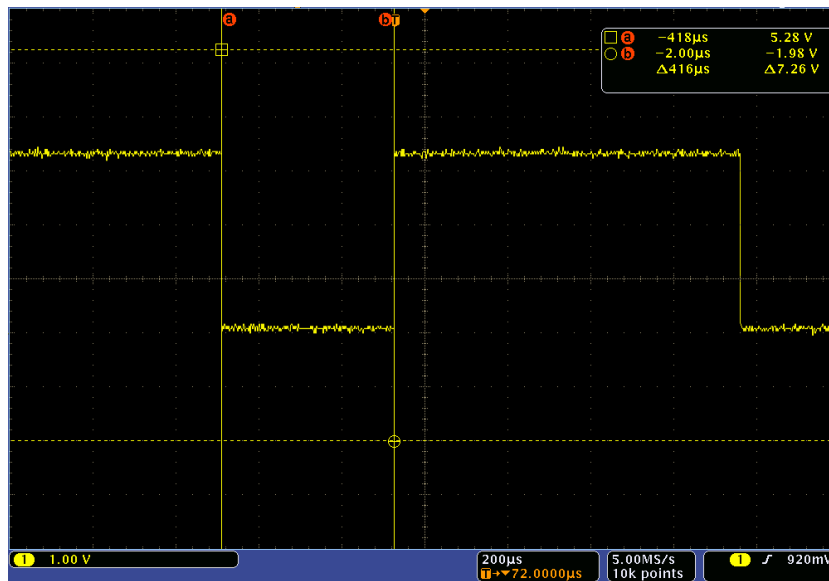


Figura 5.16 Medición del tiempo medio por bit (T_e).

Las pruebas del funcionamiento del *bridge* DALI, implementado en la FPGA embebida en el nodo sensor inalámbrico, se realizaron utilizando un enlace serie RS232 para interconectar el nodo sensor inalámbrico con una PC. Para facilitar la comunicación y transferencia de los comandos DALI entre la PC y nodo sensor inalámbrico, se utilizó un programa *HyperTerminal* a través del puerto serie COM para conectarse de forma remota. Una vez que los comandos son enviados por medio de la *HyperTerminal* a la UART, ésta recibe el dato y se lo transfiere al procesador DALI, que decodifica la instrucción y la envía al TRCM. A su vez, TRCM envía la trama de datos a la interfaz DALI, que son los bits que se obtienen a través del osciloscopio. Una prueba fue el envío de una trama “Forward” para su ejecución en el *bridge* DALI, la cual se ha verificado mediante el osciloscopio tal y como se puede observar en la Figura 5.17. De tal forma, el formato digital de la codificación Manchester cumple de forma síncrona con las transiciones ascendentes y descendentes.

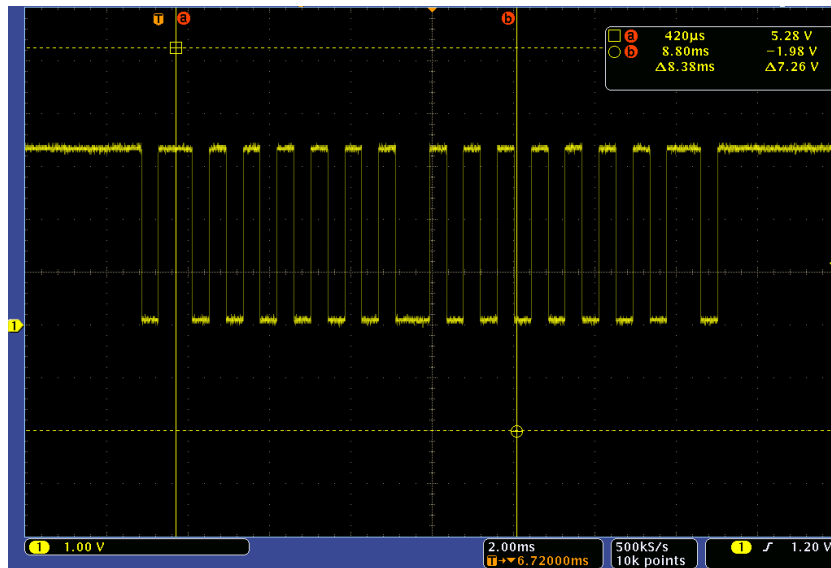


Figura 5.17 Resultado de una trama “Forward” ejecutada por el *bridge* DALI.

Se realizaron varias pruebas para comprobar que los comandos DALI enviados por el *host* se codifican correctamente. La Figura 5.18 muestra una de las múltiples pruebas realizadas. Se puede observar el resultado obtenido en el osciloscopio de las dos tramas de comunicación, la marcada con el número 1 es el comando enviado a través de la *HyperTerminal*, y la marcada con el número 2 es el resultado de la codificación realizada por el TRCM. La señal de la parte superior representa la trama recibida con codificación sin retorno al nivel de cero de tensión a una velocidad de 9600 bps, y la señal de la parte inferior representa la trama enviada por el *bridge* DALI a la interfaz DALI con codificación Manchester a una velocidad de 1200 bps. Las pruebas de transferencia de comandos DALI a la UART, al procesador DALI y al TRCM para identificar los resultados de la instrucción en la salida de este, permitió comando transferido es el mismo que se envía. El osciloscopio fue conectado tanto a la entrada de la UART como a la salida del TRCM, identificando que las dos tramas de comunicación son las correctas.

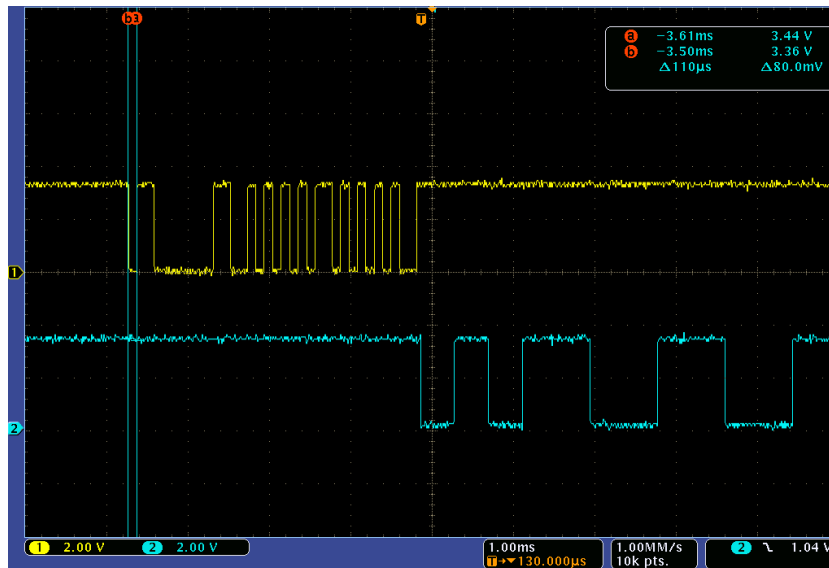


Figura 5.18 Decodificación del TRCM de un comando DALI.

5.3. Resultados de las pruebas electrónicas del nodo sensor inalámbrico

También se realizaron pruebas electrónicas para medir los niveles de voltaje de salida en la interfaz DALI, que es la que se conecta a la luminaria. Los resultados de los niveles de voltaje de salida de la interfaz DALI demuestran que son admisibles para conectarse directamente a la luminaria, ya que se encuentran dentro de las especificaciones del estándar del protocolo DALI. Con ayuda del osciloscopio se midieron los niveles de voltaje de salida del *bridge* DALI y de la interfaz DALI, tal como se observa en la Figura 5.19. La señal de color amarillo identifica los niveles de voltaje de salida del *bridge* DALI que oscilan adecuadamente entre 3,3 V y 0 V. La señal azul corresponde a los niveles de voltaje en la salida de la interfaz DALI que oscilan entre 15 V y 2,5 V, que son voltajes válidos para los niveles lógicos de voltaje definidos en el estándar del protocolo DALI.

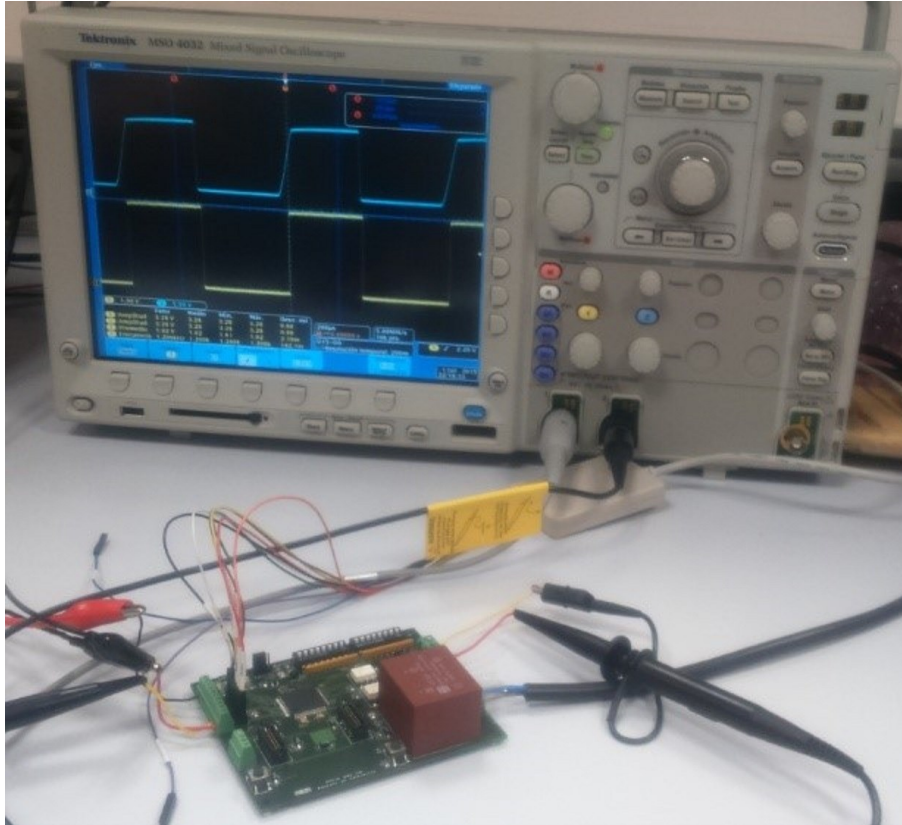


Figura 5.19 Pruebas de tensión en la salida de la interfaz DALI.

Algo fundamental en los dispositivos de HW es determinar la estabilidad y precisión del oscilador que genera la frecuencia de funcionamiento. En el caso de los componentes del nodo sensor inalámbrico la frecuencia depende del oscilador de precisión programable LTC1799. El oscilador se configuró a una frecuencia de 20 MHz tal como se indicó en el apartado 5.1.3. El funcionamiento se comprobó conectando la salida del oscilador al osciloscopio, observando que se tiene una frecuencia estable y precisa, como se muestra en la Figura 5.20.

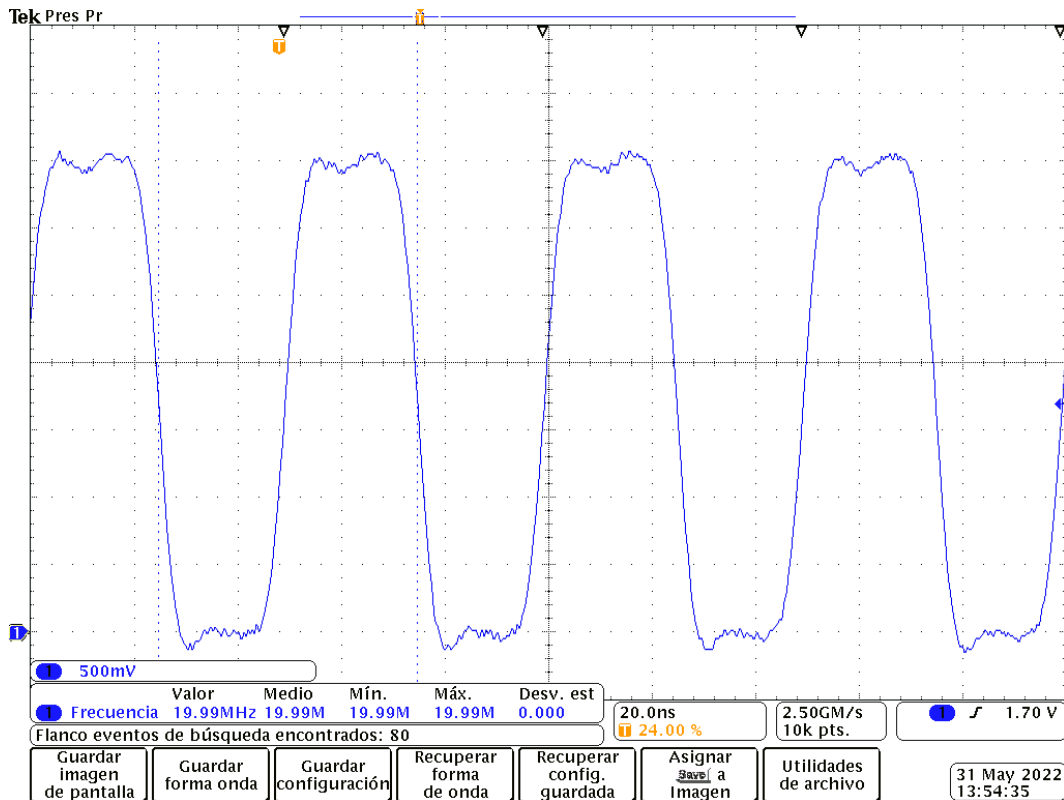


Figura 5.20 Resultado de las pruebas de estabilidad y precisión del oscilador

Se midió la estabilidad en los voltajes de alimentación que son esenciales para el funcionamiento de los componentes del nodo sensor inalámbrico. Los niveles de voltaje necesarios son: 15 V, 3,3 V, 2,5 V y 1,2 V.

La comparación de los voltajes especificados en las restricciones de los componentes y los obtenidos en las pruebas, se muestran en la Tabla 5.3. Los resultados de las pruebas demuestran que los voltajes de alimentación de energía eléctrica cumplen satisfactoriamente con el estándar y permanecen estables, lo que hace que los componentes funcionen de forma adecuada. Una imagen de las pruebas de precisión y estabilidad del voltaje de alimentación de energía eléctrica se muestra en la Figura 5.21.

Tabla 5.3 Comparación de voltajes.

Componente	Voltaje de salida	
	Especificado	Medido
Fuente de alimentación de energía eléctrica conmutada - FC47121	3,30 V	3,25 V
Convertidor de DC/DC - MEU1S0315ZC	15,00 V	14,94 V
Diodo Schottky - SB05-03Q	3,02 V	3,00 V
Regulador de tensión de baja caída - AP1122	1,20 V	1,21 V



Figura 5.21 Prueba de precisión y estabilidad del voltaje de 3,3 V.

Después de las pruebas realizadas se puede afirmar que el funcionamiento del *bridge* DALI, implementado en la FPGA, es adecuado de acuerdo a los parámetros y características del estándar, y que la tensión de alimentación de todos los componentes del nodo sensor inalámbrico son adecuadas y cumplen las especificaciones requeridas.

Capítulo 6. Conclusiones

En este capítulo se expresan las conclusiones, que son consecuencia del análisis y de la reflexión del proceso del presente trabajo de investigación y desarrollo. Con los resultados obtenidos y expuestos en el capítulo anterior se observa que, gracias a la evolución y desarrollo de las FPGAs, ahora se consideran más allá de un simple grupo de componentes lógicos de HW aislados. Su utilización se ha globalizado como una gran alternativa de solución para desarrollar sistemas inteligentes que requieren de una carga elevada de procesamiento computacional y/o implementación de interfaces inteligentes.

Como se ha constatado, en el proceso de la tesis doctoral se demostró que se ha logrado utilizar de forma eficiente FPGAs para liberar el MCU de procesamiento computacional en un nodo sensor inalámbrico, lo que generó un aumento en la confiabilidad de su uso en el contexto de las WSNs. Además, implementar por HW protocolos o procesos que históricamente se han realizados por SW, permitió comprobar que las ventajas inherentes de las FPGAs, como flexibilidad, capacidad para ejecución en paralelo, el procesamiento en tiempo real y computación de alto rendimiento, evita los problemas de sincronización en la conmutación de los puertos de E/S involucrados en la interfaz del protocolo y los cuellos de botella que se generan en la ejecución por SW.

A continuación, se presenta una lista de las conclusiones obtenidas de la investigación del trabajo de tesis realizado:

- Se expone que la descripción e implementación del protocolo DALI por HW en una FPGA de bajo coste y ultra bajo consumo de potencia, es idónea para aplicaciones en sistemas de SSL.
- La elección de la implementación del protocolo DALI por HW, creando un *bridge* entre la interfaz DALI y el MCU, aseguró la sincronización requerida por los tiempos de bit de la interfaz, la codificación Manchester y el formato de la trama. De esta forma se garantiza una ejecución y comunicación con las distintas interfaces y periféricos en tiempo real, y liberó de estas tareas relacionadas con

DALI al MCU, permitiendo integrar otros dispositivos, gobernados por el MCU, como sensores y actuadores en el nodo sensor inalámbrico. Además, el *bridge* DALI liberó a la aplicación, que se ejecuta en el MCU, de la tediosa tarea de componer las tramas enviadas, de realizar la temporización y la codificación/decodificación Manchester.

- Se determina que el tiempo total de ejecución de un comando DALI en el *bridge* DALI es mínimo, ya que está en el orden de microsegundos, lo que es suficientemente rápido para la aplicación. Las señales generadas en código Manchester por el *bridge* DALI cumplen exitosamente con el tiempo de la tasa de transferencia de bits del protocolo, difícilmente garantizable si se utiliza un MCU por la complejidad del código que se ejecuta en las aplicaciones actuales.
- Se consiguió con el uso de la FPGA una precisión muy superior en el tiempo de bit de la interfaz DALI, a la que se consigue con las implementaciones por SW debido a los retardos producidos por interrupciones y eventos originados en la ejecución del código.
- El *bridge* DALI se desarrolló para mejorar uno de los nodos sensores inalámbricos que el equipo de investigación ha utilizado en aplicaciones de sistemas de iluminación inteligente, y en el cual, se integró y verificó la correcta comunicación con las luminarias o balastos.
- Se desarrolló el nodo sensor inalámbrico en una sola PCB teniendo en forma embebida la FPGA, implementando el *bridge* DALI e integrando todos los componentes necesarios para la comunicación y control de los sensores y actuadores específicos para sistemas de SSL. Esto permitió una fácil instalación en las luminarias.
- El *bridge* DALI se describió según el estándar VHDL-93, utilizando los recursos descriptivos de alto nivel de VHDL, por lo cual se obtuvo una descripción portable, que puede ser sintetizada por todas las herramientas de síntesis. Por tanto, se puede adaptar a las especificaciones de cualquier FPGA de cualquier fabricante. Por otra parte, se optimizó la descripción para emplear el menor número posible de recursos lógicos y que estos, a su vez, sólo sean los que tienen todas las FPGAs de bajo coste, como LUTs, flip-flops y los necesarios para implementar circuitos aritméticos, como sumadores y contadores.

- Las FPGAs son una buena elección para implementar protocolos de control, buses e interfaces a periféricos en los nodos sensores inalámbricos, por su flexibilidad, bajo coste y bajo consumo de potencia.
- El espacio de componentes lógicos libres de la FPGA permite contar con la capacidad lógica suficiente para crear módulos para procesamiento, algoritmos, comunicación o control de otros sensores y/o actuadores, independientes del *bridge* DALI.
- Los resultados experimentales comprueban el correcto funcionamiento del nodo sensor inalámbrico. Las mediciones muestran que los niveles de voltaje son los adecuados en la interfaz DALI para controlar las operaciones de iluminación digital en las luminarias, además de las señales de estado, la información sobre eventos, los fallos o mal funcionamiento, para el mantenimiento de las mismas.

Por todo lo anterior mencionado se puede concluir que, el uso de FPGAs en las WSNs son un elemento imprescindible tanto para implementar interfaces inteligentes como DALI, como para aumentar la capacidad de procesamiento del nodo sin un aumento del coste ni del consumo de energía eléctrica, frente a otras alternativas como serían el uso de MCUs.

Los resultados obtenidos en la etapa de diseño y descripción del *bridge* DALI, fueron fundamentalmente importantes para su implementación en la FPGA. Durante el periodo doctoral, se presentaron los siguientes resultados en las publicaciones y congresos:

Revista Indizada.

- Ordaz-García Oscar Osvaldo, Ortiz-López Manuel, Quiles-Latorre Francisco Javier, Arceo-Olague José Guadalupe, Solís-Robles Roberto and Bellido-Outeiriño Francisco José. *DALI Bridge FPGA-Based Implementation in a Wireless Sensor Node for IoT Street Lighting Applications*. Electronics, Volume 9, Issue 11 (November 2020) 1803; <https://doi.org/10.3390/electronics9111803> JCR category rank: 145/273 in Engineering, Electrical & Electronic ISSN: 2079-9292. Factor de Impacto: 2.397 (Q3) (referenciado al año 2020).

Revista arbitrada.

- Ordaz-García Oscar Osvaldo, Ortiz-López Manuel, Quiles-Latorre Francisco Javier, Arceo-Olague José Guadalupe and Bellido-Outeiriño Francisco José. *“Implementation of the DALI Protocol in FPGAs of Low Energy Consumption for use in Wireless Sensor Networks”* Advances in Computing Science and

Applications Research in Computing Science. Vol. 147(12): ISSN 1870-4069 (formerly ISSN 1665-9899). Indexing: DBLP, LatIndex, Periódica pp. 253–263; diciembre 2018.

Congresos.

- Ordaz-García Oscar Osvaldo, Ortiz-López Manuel, Quiles-Latorre Francisco Javier, Arceo-Olague José Guadalupe and Bellido-Outeiriño Francisco José. “*Sistema de control para implementar el protocolo DALI en FPGAs usado en redes inalámbricas de sensores*” Vigésimosexta edición del Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI’19) “Instrumentación y Electrónica Industrial” Universidad de Córdoba. ISBN 978-84-17171-50-6 Depósito Legal: CO 1144-2019. 3, 4 y 5 de julio de 2019, pp 516 – 521.
- Ordaz-García Oscar Osvaldo, Ortiz-López Manuel, Quiles-Latorre Francisco Javier, Arceo-Olague José Guadalupe and Bellido-Outeiriño Francisco José. “*Optimización de hardware para implementar en FPGAs el protocolo DALI usado redes inalámbricas de sensores.*” Tlamati Sabiduría de la Universidad Autónoma de Guerrero. ISSN: 2007-2066. Vol. 9, No. Especial 3, octubre 2018, pp 24. 8º Congreso Internacional de Computación / 3ª. Reunión Red Latinoamericana de Computación. Tecnologías Emergentes en la Cultura Digital. CICOM 2018. México-Colombia.

Conferencias.

- Ordaz-García Oscar Osvaldo y Arceo-Olague José Guadalupe. Conferencia “*Redes Inalámbricas de Sensores y FPGAs*” Jornada Estatal de Ciencia y Tecnología (JECyT) 2021. Consejo Estatal de Ciencia Tecnología e Innovación de Zacatecas (COZCyT) Zacatecas, México; Noviembre – diciembre 2021.
- Ordaz-García Oscar Osvaldo. Conferencia “*Sistema de sensado de Radiación Solar implementado en un FPGA*” Jornada Estatal de Ciencia y Tecnología (JECyT) 2021. Consejo Estatal de Ciencia Tecnología e Innovación de Zacatecas (COZCyT) Zacatecas, México. Noviembre – diciembre 2021.
- Ordaz-García Oscar Osvaldo. Conferencia “*Importancia del uso de los FPGAs en redes inalámbricas de sensores*” Jornada Estatal de Ciencia y Tecnología 2020. “Ciencia, Tecnología y Sociedad ante la emergencia COVID-19”. Consejo Estatal de Ciencia Tecnología e Innovación de Zacatecas, México. Diciembre 2020

- Ordaz-García Oscar Osvaldo. Conferencia “*Implementación de una interface de control de luminarias en FPGAs usada en redes inalámbricas de sensores*”. Actividades de investigación en la Universidad Autónoma de Zacatecas Campus Jalpa, México. 04 de septiembre de 2019.

Finalmente, se concluye que en la investigación y desarrollo de la presente tesis se presenta evidencia que indica que las características de las FPGAs son viables para utilizarlas en las WSNs.

Referencias

1. Forecast. Cisco Visual Networking Index: Global Mobile Data Traffic Update, 2017–2022. Available online: <https://davidellis.ca/wp-content/uploads/2019/12/cisco-vni-mobile-data-traffic-feb-2019.pdf> 2019, (consultado el 15 de junio de 2020).
2. Parrilla, L.; Castillo, E.; López-Ramos, J.A.; Álvarez-Bermejo, J.A.; García, A.; Morales, D.P. Unified Compact ECC-AES Co-Processor with Group-Key Support for IoT Devices in Wireless Sensor Networks. *Sensors* **2018**, *18*, 251, doi:10.3390/s18010251.
3. Fernández Martínez, R.; Ordieres Meré, J.; Martínez de Pisón Ascacibar, F.J.; González Marcos, A.; Alba Elías, F.; Lostado Lorza, R.; Pernía Espinoza, A.V. *Redes inalámbricas de sensores: teoría y aplicación práctica*, Grupo de Investigación EDMANS ed.; Universidad de La Rioja, Madrid, España: 2009.
4. Lee, R.-G.; Lai, C.-C.; Chiang, S.-S.; Liu, H.-S.; Chen, C.-C.; Hsieh, G.-Y. Design and implementation of a mobile-care system over wireless sensor network for home healthcare applications. In Proceedings of 2006 International Conference of the IEEE Engineering in Medicine and Biology Society; pp. 6004-6007.
5. Milenković, A.; Otto, C.; Jovanov, E. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer communications* **2006**, *29*, 2521-2533, doi:10.1016/j.comcom.2006.02.011.
6. Lau, S.P.; Merrett, G.V.; White, N.M. Energy-efficient street lighting through embedded adaptive intelligence. In Proceedings of 2013 International Conference on Advanced Logistics and Transport, Sousse, Tunisia, May 2013; pp. 53-58.
7. Mahoor, M.; Hosseini, Z.S.; Khodaei, A.; Paaso, A.; Kushner, D. State-of-the-art in smart streetlight systems: a review. *IET Smart Cities* **2020**, *2*, 24-33, doi:10.1049/iet-smc.2019.0029.
8. Higuera, J.; Hertog, W.; Perálvarez, M.; Polo, J.; Carreras, J. Smart lighting system ISO/IEC/IEEE 21451 compatible. *IEEE sensors Journal* **2015**, *15*, 2595-2602, doi:10.1109/JSEN.2015.2390262.
9. Echelon. Corp.: Monitored outdoor lighting: Market, Challenges, Solutions and Next steps. In *Document number: SMS33-7 – Version 1.2*, Disponible en línea: <http://www.stitcs.com/EN/LonWorks/StreetlightWhitepaper.pdf> (consultado el 14 de Julio de 2021): 2007; pp 1-19.
10. Martin-Arias, M.; Huerta-Medina, N.; Rico-Secades, M. Using wireless technologies in lighting smart grids. In Proceedings of 2013 International Conference on New Concepts in Smart Cities: Fostering Public and Private Alliances (SmartMILE), Gijon, Spain, December 2013; pp. 1-6.
11. Lau, S.P.; Merrett, G.V.; Weddell, A.S.; White, N.M. A traffic-aware street lighting scheme for Smart Cities using autonomous networked sensors. *Computers & Electrical Engineering* **2015**, *45*, 192-207, doi:10.1016/j.compeleceng.2015.06.011.
12. Merlino, G.; Bruneo, D.; Distefano, S.; Longo, F.; Puliafito, A.; Al-Anbuky, A. A smart city lighting case study on an openstack-powered infrastructure. *Sensors* **2015**, *15*, 16314-16335, doi:10.3390/s150716314.
13. Ni, D.C.; Chen, P. Integration of wired and wireless lighting control systems for green energy management. In Proceedings of 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), St. Petersburg, Russia, May 2017; pp. 206-210.
14. Haba, C.-G.; Breniuc, L.; Gălățanu, C.-D.; Petrișor, D.; Herțanu, R. IoT System for Implementation of Connected Lighting. In Proceedings of 2019 11th International

- Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, Romania, May 2019; pp. 1-5.
15. Zotos, N.; Pallis, E.; Stergiopoulos, C.; Anastasopoulos, K.; Bogdos, G.; Skianis, C. Case study of a dimmable outdoor lighting system with intelligent management and remote control. In Proceedings of 2012 International Conference on Telecommunications and Multimedia (TEMU), Chania, Greece, September 2012; pp. 43-48.
 16. Shahzad, G.; Yang, H.; Ahmad, A.W.; Lee, C. Energy-efficient intelligent street lighting system using traffic-adaptive control. *IEEE Sensors Journal* **2016**, *16*, 5397-5405, doi:10.1109/JSEN.2016.2557345.
 17. Wojnicki, I.; Komnata, K.; Kotulski, L. Comparative Study of Road Lighting Efficiency in the Context of CEN/TR 13201 2004 and 2014 Lighting Standards and Dynamic Control. *Energies* **2019**, *12*, 1524, doi:10.3390/en12081524.
 18. Lee, H.; Choi, C.-h.; Sung, M. Development of a dimming lighting control system using general illumination and location-awareness technology. *Energies* **2018**, *11*, 2999, doi:10.3390/en11112999.
 19. Park, H.; Park, J.; Kim, H.; Lee, S.Q.; Park, K.-H.; Paik, J. Hybrid Sensor Network-Based Indoor Surveillance System for Intrusion Detection. *Symmetry* **2018**, *10*, 181, doi:10.3390/sym10060181.
 20. Shlayan, N.; Challapali, K.; Cavalcanti, D.; Oliveira, T.; Yang, Y. A novel illuminance control strategy for roadway lighting based on greenshields macroscopic traffic model. *IEEE Photonics Journal* **2018**, *10*, 1-11, doi:10.1109/JPHOT.2017.2782801.
 21. Wielgosz, M.; Karwatowski, M. Mapping neural networks to FPGA-based IoT devices for ultra-low latency processing. *Sensors* **2019**, *19*, 2981, doi:10.3390/s19132981.
 22. International Electrotechnical Commission. IEC 60929:2011. AC and/or DC-supplied electronic control gear for tubular fluorescent lamps – Performance requirements., Disponible en línea: <https://webstore.iec.ch/publication/3926> (consultado el 14 de julio de 2021).
 23. DALI. Standard. *International Electrotechnical Commission IEC 62386* **2020**, Disponible en línea: <https://www.digitalilluminationinterface.org/dali/standards.html> (consultado el 14 de Julio de 2021).
 24. Husain, S. Digitally Addressable Lighting Interface (DALI) Communication. AN1465. *Microchip Technology Inc.* **2012**, *DS01465A*, Disponible en línea: <http://ww1.microchip.com/downloads/en/Appnotes/01465A.pdf> (consultado el 20 de agosto de 2021).
 25. Fosler, R. The RS-232/DALI Bridge Interface. AN811. *Microchip Technology Inc.* **2004**, *DS00811A*, Disponible en línea: <http://ww1.microchip.com/downloads/en/Appnotes/00811a.pdf> (consultado el 20 de agosto de 2021).
 26. Husain, S. DALI Control Gear AN1487. *Microchip Technology Inc.* **2012**, *1*, Disponible en línea: <http://ww1.microchip.com/downloads/en/Appnotes/01487A.pdf> (consultado el 5 de mayo de 2022).
 27. 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 (Switzerland)* **2016**, *16*, 597, doi:10.3390/s16050597.

28. Cem Ünsalan, P.D.; Bora Tar, P.D. *Digital System Design with FPGA: Implementation Using Verilog and VHDL*; McGraw-Hill Education: New York, 2017.
29. Taylor, B. State-of-the-Art Programmable Logic. In *Designing with Xilinx® FPGAs*, Springer, Cham: 2017; doi:10.1007/978-3-319-42438-5_1 pp. 1-15.
30. Dello Sterpaio, L.; Marino, A.; Nannipieri, P.; Dinelli, G.; Davalle, D.; Fanucci, L. A Complete EGSE Solution for the SpaceWire and SpaceFibre Protocol Based on the PXI Industry Standard. *Sensors* **2019**, *19*, 5013, doi:10.3390/s19225013.
31. Pang, A.; Membrey, P. What Is an FPGA and What Can It Do? In *Beginning FPGA: Programming Metal*, Apress, B., CA, Ed. Springer: 2017; doi:10.1007/978-1-4302-6248-0_1 pp. 3-12.
32. Venieris, S.I.; Kouris, A.; Bouganis, C.-S. Toolflows for mapping convolutional neural networks on FPGAs: A survey and future directions. *arXiv preprint arXiv:1803.05900* **2018**, *56*, 1–39, doi:10.1145/3186332.
33. Leccese, F. Remote-Control System of High Efficiency and Intelligent Street Lighting Using a ZigBee Network of Devices and Sensors. *IEEE Transactions on Power Delivery* **2013**, *28*, 21-28, doi:10.1109/TPWRD.2012.2212215.
34. Kuo, Y.-S.; Pannuto, P.; Schmid, T.; Dutta, P. Reconfiguring the software radio to improve power, price, and portability. In Proceedings of Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Canada, November 2012; pp. 267-280.
35. Szilvási, S.; Babják, B.; Völgyesi, P.; Lédeczi, A. Marmote SDR: Experimental platform for low-power wireless protocol stack research. *Journal of Sensor and Actuator Networks* **2013**, *2*, 631-652, doi:10.3390/jsan2030631.
36. Roisin, B.; Bodart, M.; Deneyer, A.; D'herdt, P. Lighting energy savings in offices using different control systems and their real consumption. *Energy and Buildings* **2008**, *40*, 514-523, doi:10.1016/j.enbuild.2007.04.006.
37. Liu, J.; Zhang, W.; Chu, X.; Liu, Y. Fuzzy logic controller for energy savings in a smart LED lighting system considering lighting comfort and daylight. *Energy and Buildings* **2016**, *127*, 95-104, doi:10.1016/j.enbuild.2016.05.066.
38. Madzalan, S.N.; Hamid, H.; Hussin, M.F.; Kadir, K.A. A new automatic lighting control system using DALI for unkl BMI library. In Proceedings of 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), Kuala Lumpur, Malaysia, December 2017; pp. 1-4.
39. Adam, G.; Kontaxis, P.; Bouroussis, C.; Ventzas, D.; Topalis, F. Embedded computer communication and control of DALI LED drivers. In Proceedings of Balkan Light Conference, Athens, Greece, September 2015; pp. 125-130.
40. Sambandam Raju, P.; Mahalingam, M.; Arumugam Rajendran, R. Design, Implementation and Power Analysis of Pervasive Adaptive Resourceful Smart Lighting and Alerting Devices in Developing Countries Supporting Incandescent and LED Light Bulbs. *Sensors* **2019**, *19*, 2032, doi:10.3390/s19092032.
41. Adam, G.K. DALI LED Driver Control System for Lighting Operations Based on Raspberry Pi and Kernel Modules. *Electronics* **2019**, *8*, 1021, doi:10.3390/electronics8091021.
42. Teich, J. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. In *Proceedings of the IEEE - PIEEE*, Special Centennial Issue, 2012; Vol. 100, pp 1411-1430.
43. Benini, L.; Farella, E.; Guiducci, C. Wireless sensor networks: Enabling technology for ambient intelligence. *Microelectronics journal* **2006**, *37*, 1639-1649, doi:10.1016/j.mejo.2006.04.021

44. Bhushan, B.; Sahoo, G. Routing Protocols in Wireless Sensor Networks. In *Computational Intelligence in Sensor Networks*, B. B. Mishra et al. (eds.) Studies in Computational Intelligence 776, Ed. Springer, Berlin, Heidelberg: 2019; doi:10.1007/978-3-662-57277-1_10 pp. 215-248.
45. Krishnamachari, B.; Estrin, D.; Wicker, S. Modelling data-centric routing in wireless sensor networks. In *Proceedings of IEEE infocom*; pp. 39-44.
46. Bhanumathi, V.; Kalaivanan, K. Application specific sensor-cloud: architectural model. In *Computational Intelligence in Sensor Networks*, B. B. Mishra et al. (eds.) Studies in Computational Intelligence 776, Ed. Springer, Berlin, Heidelberg: 2019; doi:10.1007/978-3-662-57277-1_12 pp. 277-305.
47. El-Medany, W.M. FPGA-based MIMO system for Wireless Sensor Network. In *System of Systems Engineering, 2009. SoSE 2009. IEEE International Conference on*, Albuquerque, NM, USA, 2009; pp 1-5.
48. Yick, J.; Mukherjee, B.; Ghosal, D. Wireless sensor network survey. *Computer networks* **2008**, *52*, 2292-2330, doi:10.1016/j.comnet.2008.04.002
49. Babber, K.; Randhawa, R. Cross-layer designs in wireless sensor networks. In *Computational Intelligence in Sensor Networks*, Springer-Verlag GmbH Germany, p.o.S.N., Ed. Springer-Verlag GmbH Germany: 2019; doi:0.1007/978-3-662-57277-1_7 pp. 141-166.
50. Anastasi, G.; Conti, M.; Di Francesco, M.; Passarella, A. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks* **2009**, *7*, 537-568, doi:10.1016/j.adhoc.2008.06.003
51. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: a survey. *Computer networks* **2002**, *38*, 393-422, doi:10.1016/S1389-1286(01)00302-4
52. Singh, S.P.; Sharma, S. A survey on cluster based routing protocols in wireless sensor networks. *Procedia computer science* **2015**, *45*, 687-695, doi:10.1016/j.procs.2015.03.133.
53. Rani, R. Distributed query processing optimization in wireless sensor network using artificial immune system. In *Computational Intelligence in Sensor Networks*, Springer-Verlag GmbH Germany, p.o.S.N., Ed. Springer: 2019; 10.1007/978-3-662-57277-1_1 pp. 1-23.
54. Raghavendra, C.S.; Sivalingam, K.M.; Znati, T. *Wireless sensor networks*; Springer: New York, 2006.
55. Znati, T.; Raghavendra, C.; Sivalingam, K. Guest editorial: special issue on wireless sensor networks. *Mobile Networks and Applications* **2003**, *8*, 425-425.
56. Wang, Q.; Balasingham, I. Wireless sensor networks-an introduction. *Wireless sensor networks: application-centric design* **2010**, -, 1-14, doi:10.5772/658
57. Chong, C.-Y.; Kumar, S.P. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE* **2003**, *91*, 1247-1256, doi:10.1109/JPROC.2003.814918
58. Kumar, S.; Shepherd, D. SensIT: Sensor information technology for the warfighter. In *Proceedings of Proc. 4th Int. Conf. on Information Fusion*; pp. 1-7.
59. Pottie, G.J. Wireless Integrated Network Sensors (WINS): The Web Gets Physical. In *Proceedings of Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2001 NAE Symposium on Frontiers of Engineering*; pp. 78-84.
60. Bult, K.; Burstein, A.; Chang, D.; Dong, M.; Fielding, M.; Kruglick, E.; Ho, J.; Lin, F.; Lin, T.; Kaiser, W.J. Low power systems for wireless microsensors. In *Proceedings of Proceedings of 1996 International Symposium on Low Power Electronics and Design*, Monterey, CA, USA, 12-14 Aug. 1996; pp. 17-21.

61. Khan, J.; Katz, R.; Pister, K. Mobile networking for smart dust. In Proceedings of Proc. of ACK/IEEE International Conference on Mobile Computing and Networking (MobiCom).
62. Rabaey, J.; Ammer, J.; Da Silva, J.; Patel, D. PicoRadio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. In Proceedings of Proceedings IEEE Computer Society Workshop on VLSI 2000. System Design for a System-on-Chip Era, 27-28 April 2000; pp. 9-12.
63. Min, R.; Bhardwaj, M.; Cho, S.-H.; Sinha, A.; Shih, E.; Wang, A.; Chandrakasan, A. An architecture for a power-aware distributed microsensor node. In Proceedings of 2000 IEEE Workshop on SiGNAL PROCESSING SYSTEMS. SiPS 2000. Design and Implementation (Cat. No. 00TH8528), Lafayette, LA, USA, 06 August 2002; pp. 581-590.
64. Calhoun, B.H.; Daly, D.C.; Verma, N.; Finchelstein, D.F.; Wentzloff, D.D.; Wang, A.; Cho, S.-H.; Chandrakasan, A.P. Design considerations for ultra-low energy wireless microsensor nodes. *IEEE Transactions on Computers* **2005**, *54*, 727-740, doi:10.1109/TC.2005.98.
65. Sohraby, K.; Minoli, D.; Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*; John Wiley & Sons, Inc., Publication: Wiley-Interscience, 2007; 10.1002/047011276X
66. Behera, T.; Samal, U.; Mohapatra, S. Routing protocols. In *Computational Intelligence in Sensor Networks*, B. B. Mishra et al. (eds.) Studies in Computational Intelligence 776, Ed. Springer, Berlin, Heidelberg.: 2019; 10.1007/978-3-662-57277-1_4pp. 79-99.
67. Lee, J.; Kao, H.-A.; Yang, S. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia cirp* **2014**, *16*, 3-8, doi:10.1016/j.procir.2014.02.001.
68. Ammar, R.; Samer, S. *Internet of Things From Hype to Reality: The Road to Digitization*, 2 ed.; Springer, Cham: Switzerland, 2019; doi:10.1007/978-3-319-99516-8 pp. 373.
69. Rayes, A.; Salam, S. Internet of things (IoT) overview. In *Internet of Things from hype to reality*, 2019, S.N.S.A., Ed. Springer, Cham: Switzerland, 2019; doi:10.1007/978-3-319-99516-8_1 pp. 1-35.
70. Jindal, V. History and architecture of wireless sensor networks for ubiquitous computing. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* **2018**, *7*, 214-217.
71. IEC. *Internet of things: Wireless sensor networks*; International Electrotechnical Commission: Switzerland, 2014.
72. Minoli, D. *Building the internet of things with IPv6 and MIPv6: The evolving world of M2M communications*; John Wiley & Sons: 2013.
73. Nahida, S.; Yasin, S.; Kurra, R.R. Energy Efficient Routing Protocol for Wireless Sensor Networks. *International Journal of Computer Science Engineering & Technology* **2012**, *2*, 1497-1501.
74. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Communications magazine* **2002**, *40*, 102-114, doi:10.1109/MCOM.2002.1024422
75. Pantazis, N.A.; Nikolidakis, S.A.; Vergados, D.D. Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Communications surveys & tutorials* **2013**, *15*, 551-591, doi:10.1109/SURV.2012.062612.00084
76. Morreale, P.; Sohraby, K.; Li, B.; Lin, Y.-B. Active, programmable, and mobile code networking. *IEEE Communications Magazine*: 2000; Vol. 38, pp 122-123.

77. Ulmer, C. Wireless Sensor Networks. Background: Sensor Networks n Array of Sensor Probes (10-1000) n Collect In-Situ Data about Environment n Wireless. 2000.
78. Mokdad, L.; Abdelli, A.; Ben-Othman, J. Detection of greedy behavior in wsn using ieee 802.15 protocol. In Proceedings of 2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems, Paris, France, 9-11 Sept. 2014; pp. 106-111.
79. Zhang, H.; Arora, A.; Choi, Y.-r.; Gouda, M.G. Reliable bursty convergecast in wireless sensor networks. *Computer Communications* **2007**, *30*, 2560-2576, doi:10.1016/j.comcom.2007.05.046
80. Dargie, W.; Poellabauer, C. *Fundamentals of wireless sensor networks: theory and practice*; John Wiley & Sons, Ltd. Publication: 2010; doi:10.1002/9780470666388
81. Pister, K.; Doherty, L. TSMP: Time synchronized mesh protocol. *IASTED International Symposium, Distributed Sensor Networks (DNS 2008)* **2008**, *391*, 61.
82. Swati, A.J.; Priyanka, R. Wireless sensor network (WSN): Architectural design issues and challenges. *Int. J. Comput. Sci. Eng* **2010**, *2*, 3089-3094.
83. Nayak, A.; Stojmenovic, I. *Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
84. Dargie, W. Dynamic power management in wireless sensor networks: State-of-the-art. *IEEE Sensors Journal* **2012**, *12*, 1518-1528, doi:10.1109/JSEN.2011.2174149
85. Kumar, M. AOR-ID-KAP: An Authenticated One-Round Identity-Based Key Agreement Protocol for Wireless Sensor Network. In *Computational Intelligence in Sensor Networks*, B. B. Mishra et al. (eds.) Studies in Computational Intelligence 776, Ed. Springer, Berlin, Heidelberg: 2019; 10.1007/978-3-662-57277-1_18pp. 427-454.
86. Pasolini, G.; Toppan, P.; Zabini, F.; De Castro, C.; Andrisano, O. Design, deployment and evolution of heterogeneous smart public lighting systems. *Applied Sciences* **2019**, *9*, 3281, doi:10.3390/app9163281.
87. Caicedo, D.; Pandharipande, A. Distributed illumination control with local sensing and actuation in networked lighting systems. *IEEE Sensors Journal* **2012**, *13*, 1092-1104, doi:10.1109/JSEN.2012.2228850.
88. Valentová, M.; Quicheron, M.; Bertoldi, P. LED projects and economic test cases in Europe. *International Journal of Green Energy* **2015**, *12*, 843-851, doi:10.1080/15435075.2014.887568.
89. Mahoor, M.; Najafaabadi, T.A.; Salmasi, F.R. A smart street lighting control system for optimization of energy consumption and lamp life. In Proceedings of 2014 22nd Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, May 2014; pp. 1290-1294.
90. Mohamaddoust, R.; Haghighat, A.T.; Motahari Sharif, M.J.; Capanni, N. A novel design of an automatic lighting control system for a wireless sensor network with increased sensor lifetime and reduced sensor numbers. *Sensors* **2011**, *11*, 8933-8952, doi:10.3390/s110908933
91. Liu, D.; Qi, S.; Liu, T.; Yu, S.-z.; Sun, F. The design and realization of communication technology for street lamps control system. In Proceedings of 2009 4th International Conference on Computer Science & Education, Nanning, China, 25-28 July 2009; pp. 259-262.
92. Lin, J.; Jin, X.; Mao, Q. Wireless monitoring system of street lamps based on ZigBee. In Proceedings of 2009 5th International Conference on Wireless

- Communications, Networking and Mobile Computing, Beijing, China, 24-26 Sept. 2009; pp. 1-3.
93. Viani, F.; Polo, A.; Robol, F.; Giarola, E.; Ferro, A. Experimental validation of a wireless distributed system for smart public lighting management. In Proceedings of 2016 IEEE International Smart Cities Conference (ISC2), Trento, Italy; pp. 1-6.
 94. Cimdins, M.; Arlt, W.; Hellbrueck, H. IoTiSS: Internet of Things in Smart Streetlighting. In Proceedings of Mobile Communication-Technologies and Applications; 24. ITG-Symposium, Osnabrueck, Germany, 06 June 2019; pp. 1-6.
 95. Chang, Y.S.; Chen, Y.H.; Zhou, S.K. A smart lighting system for greenhouses based on Narrowband-IoT communication. In Proceedings of 2018 13th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), Taipei, Taiwan, 28 January 2019; pp. 275-278.
 96. Sikder, A.K.; Acar, A.; Aksu, H.; Uluagac, A.S.; Akkaya, K.; Conti, M. IoT-enabled smart lighting systems for smart cities. In Proceedings of 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA; pp. 639-645.
 97. Galkin, I.; Tetervenoks, O. The study of microcontroller based embedded system for smart lighting applications. In Proceedings of 2014 6th European Embedded Design in Education and Research Conference (EDERC), Milan, Italy; pp. 105-108.
 98. Kumar, A.; Shareef, A.; Harn, K.T.; Kar, P.; Panda, S.K. A Complete Hardware Setup for Smart Lighting System. In Proceedings of 2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS), Bhubaneswar, India; pp. 297-301.
 99. Lee, C.K.; Liu, H.; Fuhs, D.; Kores, A.; Waffenschmidt, E. Smart lighting systems as a demand response solution for future smart grids. *IEEE Journal of Emerging and Selected Topics in Power Electronics* **2019**, *8*, 2362-2370, doi:10.1109/JESTPE.2018.2890385
 100. Tripathy, A.K.; Mishra, A.K.; Das, T.K. Smart lighting: Intelligent and weather adaptive lighting in street lights using IoT. In Proceedings of 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, India; pp. 1236-1239.
 101. Halefoglul, L.; Jiang, X.; Kendrick, A.J.; Saunders, G.D.; Sciarrino, M.; Vizner, G.; Bailey, R. Smart lighting: Developing a smarter control mechanism for park trail lighting. In Proceedings of 2016 IEEE Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA; pp. 277-282.
 102. Jin, D.; Hannon, C.; Li, Z.; Cortes, P.; Ramaraju, S.; Burgess, P.; Buch, N.; Shahidehpour, M. Smart street lighting system: A platform for innovative smart city applications and a new frontier for cyber-security. *The Electricity Journal* **2016**, *29*, 28-35, doi:10.1016/j.tej.2016.11.011
 103. Smith, J.R. Chapter 20 - X-10 Home Automation. In *Programming the PIC Microcontroller with MBASIC*, Smith, J.R., Ed. Newnes: Burlington, 2005; doi:10.1016/B978-075067946-6/50022-5 pp. 453-486.
 104. Burroughs, J. X-10® home automation using the PIC16F877A. *Lamp Microchip Technology Inc.* **2010**, *10*.
 105. Cho, S.; Dhingra, V. Street lighting control based on LonWorks power line communication. In Proceedings of 2008 IEEE International Symposium on Power Line Communications and Its Applications, Jeju, Korea (South); pp. 396-398.
 106. Muresan, N.; Orza, B.; Vlaicu, A. Communication Between Intelligent Devices in LONWorks Control Networks. *Acta Technica Napocensis - Electronics and Telecommunications* **2010**, *51*, 57-63.

107. Chinchero, H.F.; Alonso, J.M.; Ortiz, H. LED lighting systems for smart buildings: a review. *IET Smart Cities* **2020**, *2*, 126-134, doi:10.1049/iet-smc.2020.0061
108. Ching, T.W. Transition-mode dimmable LED driver for illumination applications. In Proceedings of The 2010 International Power Electronics Conference-ECCE ASIA-, Sapporo, Japan; pp. 398-402.
109. Ching, T.W. Modular dimmable light-emitting-diode driver for general illumination applications. In Proceedings of CCECE 2010, Calgary, AB, Canada, 16 September 2010; pp. 1-4.
110. Jiang, W.; Jiang, Y.; Ren, H. Analysis and prospect of control system for stage lighting. In Proceedings of 2010 3rd International Congress on Image and Signal Processing, Yantai, China, 29 November 2010; pp. 3923-3929.
111. Kudryashov, A.V.; Galishheva, E.S.; Kalinina, A.S. Lighting Control Using DALI Interface. In Proceedings of 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Moscow, Russia, Russia, June 2019; pp. 1-5.
112. Rasal, S.P. Energy Efficient Building Automation Systems Using Protocols and Control Methods. *Development (IJECIERD)* **2013**, *3(4)*, 65-72.
113. International Electrotechnical Commission. IEC 62386-201:2015. Digital Addressable Lighting Interface – Part 201: Particular requirements for control gear – Fluorescent lamps., Disponible en línea: <https://webstore.iec.ch/publication/22534> (consultado el 14 de Julio de 2021).
114. ZVEI-Division-Luminaires. Manual DALI AG (Digital Addressable Lighting Interface Activity Group) **2001**, Frankfurt, Germany, Disponible en línea: <http://www.susaeta.net/attachments/article/42/manual%20dali.pdf> (consultado el 14 de Julio de 2021).
115. Barkalov, A.; Titarenko, L.; Kolopiencyk, M.; Mielcarek, K.; Bazydlo, G. Field Programmable Gate Arrays in FSM Design. In *Logic Synthesis for FPGA-Based Finite State Machines*, doi:10.1007/978-3-319-24202-6_2 ed.; Springer, Cham: 2016; Vol. 38, pp. 33-64.
116. Carter, W.; Duong, K.; Freeman, R.H.; Hsieh, H.C.; Ja, J.Y.; Mahoney, J.E.; Ngo, L.T.; Sze, S.L. A User Programmable Reconfigurable Gate Array. In Proceedings of IEEE Custom Integrated Circuits Conference; pp. 233-235.
117. Trimberger, S.M.S. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology: This Paper Reflects on How Moore's Law Has Driven the Design of FPGAs Through Three Epochs: the Age of Invention, the Age of Expansion, and the Age of Accumulation. *IEEE Solid-State Circuits Magazine* **2018**, *10*, 16-29, doi:10.1109/MSSC.2018.2822862.
118. Pang, A.; Membrey, P. FPGA Development Timeline. In *Beginning FPGA: Programming Metal*, Apress, B., CA, Ed. Springer: 2017; doi:10.1007/978-1-4302-6248-0_5 pp. 107-115.
119. Brown, S.; Rose, J. FPGA and CPLD architectures: A tutorial. *IEEE design & test of computers* **1996**, *13*, 42-57, doi:10.1109/54.500200.
120. Trimberger, S.M. *Field-Programmable Gate Array Technology*; Boston, MA, USA.: 1994.
121. Kean, T.A. Configurable logic: a dynamically programmable cellular architecture and its VLSI implementation. *Ph.D. dissertation CST62-89 1991*, Dept. Comput. Sci., Univ. Edinburgh, United Kingdom.
122. Cong, J.; Ding, Y. FlowMap: An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs. *IEEE Transactions on*

- Computer-Aided Design of Integrated Circuits and Systems* **1994**, *13*, 1-12, doi:10.1109/43.273754
123. Betz, V.; Rose, J.; Marquardt, A. Architecture and CAD for Deep-Submicron FPGAs. The Kluwer International Series in Engineering and Computer Science. Academic.: 1999; Boston, MA, USA.
 124. Betz, V.; Rose, J. VPR: A new packing, placement and routing tool for FPGA research. In Proceedings of International Workshop on Field Programmable Logic and Applications; pp. 213-222.
 125. Frankle, J. Iterative and adaptive slack allocation for performance-driven layout and FPGA routing. In Proceedings of IEEE Design Autom. Conf, DAC; pp. 536-542.
 126. McMurchie, L.; Ebeling, C. PathFinder: a negotiation-based performance-driven router for FPGAs. In Proceedings of Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays, Monterey, California, USA; pp. 111-117.
 127. Roychowdhury, V.P.; Greene, J.W.; El Gamal, A. Segmented channel routing. *IEEE transactions on computer-aided design of integrated circuits and systems* **1993**, *12*, 79-95, doi:10.1109/43.184845.
 128. DeHon, A. Balancing Interconnect and Computation in a Reconfigurable Computing Array (or, why you don't really want 100% LUT utilization). In Proceedings of Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays, Feb. 1999; pp. 69-78.
 129. Halfhill, T. MicroBlaze v7 Gets an MMU. *Microprocessor Rep.* **2007**.
 130. Kuon, I.; Rose, J. Measuring the gap between FPGAs and ASICs. *IEEE Transactions on computer-aided design of integrated circuits and systems* **2007**, *26*, 203-215.
 131. Lewis, D.; Ahmed, E.; Cashman, D.; Vanderhoek, T.; Lane, C.; Lee, A.; Pan, P. Architectural enhancements in Stratix-III™ and Stratix-IV™. In Proceedings of Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays, February 2009; pp. 33-42.
 132. Tuan, T.; Kao, S.; Rahman, A.; Das, S.; Trimberger, S. A 90nm low-power FPGA for battery-powered applications. In Proceedings of Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays, February 2006; pp. 3-11.
 133. Pang, A.; Membrey, P. Our Weapon of Choice. In *Beginning FPGA: Programming Metal*, Apress, B., CA, Ed. Springer: 2017; doi:10.1007/978-1-4302-6248-0_2 pp. 13-21.
 134. Lattice-Semiconductor. iCE40 LP/HX Family Data Sheet. *Disponible en línea: <http://www.latticesemi.com/en/Products/FPGAandCPLD/iCE40> (consultado el 20 de agosto de 2021)* **2018**, *FPGA-DS-02029-3.5*.
 135. Oliveira, D.; Costa, M.; Pinto, S.; Gomes, T. The future of low-end motes in the Internet of Things: A prospective paper. *Electronics* **2020**, *9*, 111, doi:10.3390/electronics9010111.
 136. Lattice-Semiconductor. Accelerating Implementation of Low Power Artificial Intelligence at the Edge. *Disponible en línea: <https://www.latticesemi.com/Search.aspx?&lcid=9&q=&t=-420&qdoctype=d0698b4192224dfda5a6f5add96ae1bd> (consultado el 25 de enero de 2022)* **2018**, *Technical Report*.
 137. Mo, J.P.T.; Nemes, L. *Global Engineering, Manufacturing and Enterprise Networks: IFIP TC5 WG5. 3/5.7/5.12 Fourth International Working Conference on the Design of Information Infrastructure Systems for Manufacturing (DIISM 2000)*.

- November 15–17, 2000, Melbourne, Victoria, Australia, 1 ed.; Springer Science & Business Media: Springer, Boston, MA, 2001; Vol. 63.
138. Woods, R.; McAllister, J.; Lightbody, G.; Yi, Y. *FPGA-based implementation of signal processing systems*; Wiley Online Library: 2017; doi:10.1002/9781119079231.
 139. Dimond, R.; Racaniere, S.; Pell, O. Accelerating large-scale HPC Applications using FPGAs. In Proceedings of 2011 IEEE 20th Symposium on Computer Arithmetic, Tuebingen, Germany, 25-27 July 2011 pp. 191-192.
 140. Rodriguez Andina, J.J.; De la Torre Aranz, E.; Valdes Peña, M.D. *FPGAs: fundamentals, advanced features, and applications in industrial electronics*; CRC Press: 2017; doi:10.1201/9781315162133 pp. 265.
 141. LaMeres, B.J. Modeling Finite State Machines. In *Quick Start Guide to VHDL*, Springer, Cham: 2019; doi:10.1007/978-3-030-04516-6_9 pp. 127-141.
 142. LaMeres, B.J. Behavioral Modeling of Sequential Logic. In *Introduction to Logic Circuits & Logic Design with VHDL*, Springer, Cham: 2019; 10.1007/978-3-030-12489-2_9 pp. 329-360.
 143. Pang, A.; Membrey, P. Combinatorial Logic: Putting It All Together on the FPGA. In *Beginning FPGA: Programming Metal*, Apress, B., CA, Ed. Springer: 2017; doi:10.1007/978-1-4302-6248-0_11 pp. 191-217.
 144. LaMeres, B.J. Sequential Logic Design. In *Introduction to Logic Circuits & Logic Design with VHDL*, Springer, Cham: 2019; 10.1007/978-3-030-12489-2_7 pp. 211-284.
 145. Barkalov, A.; Titarenko, L.; Kolopienczyk, M.; Mielcarek, K.; Bazydło, G. Background of Finite State Machines and Programmable Logic. In *Logic Synthesis for FPGA-Based Finite State Machines*, doi:10.1007/978-3-319-24202-6_1 ed.; Springer, Cham: 2016; Vol. 38, pp. 1-31.
 146. Maxinez, D.G.; Alcalá Jara, J. *VHDL: El arte de programar sistemas digitales*, 7ma ed.; Grupo Editorial Patria: México, 2010.
 147. Chu, P.P. *FPGA prototyping by VHDL examples: Xilinx Spartan-3 version*; John Wiley & Sons: Cleveland State University, 2011.
 148. Gupta, A. UART Communication. In *The IoT Hacker's Handbook: A Practical Guide to Hacking the Internet of Things*, Springer Science+Business - Apress Media: Walnut, CA, USA, 2019; doi:10.1007/978-1-4842-4300-8_4 pp. 59-80.
 149. Subero, A. USART, SPI, and I2C: Serial Communication Protocols. In *Programming PIC Microcontrollers with XC8*, Springer Science+Business - Apress Media: 2018; doi:978-1-4842-3273-6_9 pp. 209-276.
 150. Jing, C.; Shu, D.; Gu, D. Design of streetlight monitoring and control system based on wireless sensor networks. In Proceedings of 2007 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China, 24 September 2007; pp. 57-62.
 151. Müllner, R.; Riener, A. An energy efficient pedestrian aware Smart Street Lighting system. *International Journal of Pervasive Computing and Communications* **2011**, *7*, 147–161, doi:10.1108/17427371111146437
 152. Pantoni, R.; Brandão, D. A confirmation-based geocast routing algorithm for street lighting systems. *Computers & Electrical Engineering* **2011**, *37*, 1147-1159, doi:10.1016/j.compeleceng.2011.06.004.
 153. Leccese, F.; Leonowicz, Z. Intelligent wireless street lighting system. In Proceedings of 2012 11th International Conference on Environment and Electrical Engineering, Venice, Italy, June 2012; pp. 958-961.

154. Yusoff, Y.M.; Rosli, R.; Karnaluddin, M.U.; Samad, M. Towards smart street lighting system in Malaysia. In Proceedings of 2013 IEEE Symposium on Wireless Technology & Applications (ISWTA), Kuching, Malaysia, 19 December 2013; pp. 301-305.
155. Zhang, J.; Qiao, G.; Song, G.; Sun, H.; Ge, J. Group decision making based autonomous control system for street lighting. *Measurement* **2013**, *46*, 108-116, doi:10.1016/j.measurement.2012.05.025
156. Kovács, A.; Bátai, R.; Csáji, B.C.; Dudás, P.; Háy, B.; Pedone, G.; Révész, T.; Váncza, J. Intelligent control for energy-positive street lighting. *Energy* **2016**, *114*, 40-51, doi:10.1016/j.energy.2016.07.156
157. Todorović, B.M.; Samardžija, D. Road lighting energy-saving system based on wireless sensor network. *Energy Efficiency* **2017**, *10*, 239-247, doi:10.1007/s12053-016-9447-6
158. Kaleem, Z.; Ahmad, I.; Lee, C. Smart and energy efficient led street light control system using ZigBee network. In Proceedings of 2014 12th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, June 2015; pp. 361-365.
159. Elejoste, P.; Angulo, I.; Perallos, A.; Chertudi, A.; Zuazola, I.J.G.; Moreno, A.; Azpilicueta, L.; Astrain, J.J.; Falcone, F.; Villadangos, J. An easy to deploy street light control system based on wireless communication and LED technology. *Sensors (Basel, Switzerland)* **2013**, *13*, 6492-6523, doi:10.3390/s130506492.
160. Leccese, F.; Cagnetti, M.; Trinca, D. A smart city application: A fully controlled street lighting isle based on Raspberry-Pi card, a ZigBee sensor network and WiMAX. *Sensors* **2014**, *14*, 24408-24424, doi:10.3390/s141224408.
161. García-Castellano, M.; González-Romo, J.M.; Gómez-Galán, J.A.; García-Martín, J.P.; Torralba, A.; Pérez-Mira, V. ITERL: a wireless adaptive system for efficient road lighting. *Sensors* **2019**, *19*, 5101, doi:10.3390/s19235101.
162. Domingo-Perez, F.; Gil-de-Castro, A.; Flores-Arias, J.; Bellido-Outeirino, F.; Moreno-Munoz, A. Lighting control system based on DALI and wireless sensor networks. In Proceedings of 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), Washington, DC, USA, April 2012; pp. 1-6.
163. Pinto, M.F.; Soares, G.M.; Mendonça, T.R.; Almeida, P.S.; Braga, H.A. Smart modules for lighting system applications and power quality measurements. In Proceedings of 2014 11th IEEE/IAS International Conference on Industry Applications, Juiz de Fora, Brazil, March 2015; pp. 1-8.
164. Wang, S.-C.; Liu, Y.-H.; Chen, Y.-L.; Chen, J.-Y. Development of DALI-based electronic ballast with energy saving control for ultraviolet lamps. In Proceedings of 2010 8th IEEE International Conference on Industrial Informatics, Osaka, Japan, August 2010; pp. 214-219.
165. Adam, G.K.; Kontaxis, P.A.; Doulos, L.T.; Madias, E.-N.D.; Bouroussis, C.A.; Topalis, F.V. Embedded microcontroller with a CCD camera as a digital lighting control system. *Electronics* **2019**, *8*, 33, doi:10.3390/electronics8010033.
166. Texas-Instruments. Application Report: Digital Addressable Lighting Interface (DALI) Implementation Using MSP430 Value Line Microcontrollers, SLAA422A. *Texas-Instruments* **2012**, *SLAA422A*, Disponible en línea: <http://www.ti.com/lit/an/slaa422a/slaa422a.pdf> (consultado el 14 de Julio de 2021).
167. tridonic. DALI Interface RS232 PS/S PC interface module for DALI systems. *Lighting Controls and Connectivity*. Disponible en línea: https://www.tridonic.com/com/en/download/data_sheets/DALI_RS232_Interface_e_n.pdf (consultado el 5 mayo 2022).

168. de la Piedra, A.; Braeken, A.; Touhafi, A. Sensor systems based on FPGAs and their applications: A survey. *Sensors* **2012**, *12*, 12235-12264, doi:10.3390/s120912235.
169. Pena, M.D.V.; Rodriguez-Andina, J.J.; Manic, M. The internet of things: The role of reconfigurable platforms. *IEEE Industrial Electronics Magazine* **2017**, *11*, 6-19, doi:10.1109/MIE.2017.2724579.
170. Gomes, T.; Salgado, F.; Tavares, A.; Cabral, J. CUTE mote, a customizable and trustable end-device for the internet of things. *IEEE Sensors Journal* **2017**, *17*, 6816-6824, doi:10.1109/JSEN.2017.2743460.
171. Engel, A.; Koch, A. Heterogeneous wireless sensor nodes that target the Internet of Things. *IEEE Micro* **2016**, *36*, 8-15, doi:10.1109/MM.2016.100.
172. Silva, M.; Tavares, A.; Gomes, T.; Pinto, S. ChamelloT: An agnostic operating system framework for reconfigurable IoT devices. *IEEE Internet of Things Journal* **2018**, *6*, 1291-1292, doi:10.1109/JIOT.2018.2863545.
173. El Kateeb, A. Mote design supported with remote hardware modifications capability for wireless sensor network applications. *International Journal of Advanced Smart Sensor Network Systems (IJASSN)* **2013**, *3*, 13-21, doi:10.5121/ijassn.2013.3301.
174. Gomes, T.; Pinto, S.; Salgado, F.; Tavares, A.; Cabral, J. Building IEEE 802.15. 4 accelerators for heterogeneous wireless sensor nodes. *IEEE Sensors Letters* **2017**, *1*, 1-4, doi:10.1109/LSENS.2017.2681625.
175. Patil, V.; Deshpande, S. Design of FPGA Soft Core Based WSN Node Using Customization Paradigm. *Wireless Personal Communications* **2022**, *122*, 783-805, doi:10.1007/s11277-021-08925-y.
176. Patil, V.S.; Mane, Y.B.; Deshpande, S. FPGA based power saving technique for sensor node in wireless sensor network (WSN). In *Computational Intelligence in Sensor Networks*, B. B. Mishra et al. (eds.) Studies in Computational Intelligence 776, Ed. Springer, Berlin, Heidelberg: 2019; doi:10.1007/978-3-662-57277-1_16 pp. 385-404.
177. Toubal, A.; Bengherbia, B.; Zmirli, M.O.; Guessoum, A. FPGA implementation of a wireless sensor node with built-in security coprocessors for secured key exchange and data transfer. *Measurement* **2020**, *153*, 107429, doi:10.1016/j.measurement.2019.107429.
178. Mahdi, S.Q.; Gharghan, S.K.; Hasan, M.A. FPGA-Based neural network for accurate distance estimation of elderly falls using WSN in an indoor environment. *Measurement* **2021**, *167*, 108276, doi:10.1016/j.measurement.2020.108276.
179. Sun, Y.; Li, L.; Luo, H. Design of FPGA-based multimedia node for WSN. In *Proceedings of 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, China 10 October 2011; pp. 1-5.
180. Bengherbia, B.; Ould Zmirli, M.; Toubal, A.; Guessoum, A. FPGA-based wireless sensor nodes for vibration monitoring system and fault diagnosis. *Measurement: Journal of the International Measurement Confederation* **2017**, *101*, 81-92, doi:10.1016/j.measurement.2017.01.022.
181. Wei, J.; Wang, L.; Wu, F.; Chen, Y.; Ju, L. Design and implementation of wireless sensor node based on open core. In *Proceedings of 2009 IEEE Youth Conference on Information, Computing and Telecommunication*, Beijing, 15 January 2010; pp. 102-105.
182. Zhiyong, C.H.; Pan, L.Y.; Zeng, Z.; Meng, M.Q.-H. A novel FPGA-based wireless vision sensor node. In *Proceedings of 2009 IEEE International Conference on Automation and Logistics*, Shenyang, China, 5-7 Aug. 2009; pp. 841-846.

183. Quiles, F.J.; Ortiz, M.; Gersnoviez, A.; Brox, M.; Olivares, A.; Glosekotter, P. Development of a wireless low power datalogger with high performance converter. *Elektronika ir Elektrotechnika* **2015**, *21*, 21-27, doi:10.5755/j01.eee.21.3.10044.
184. iCEcube2. Design Software (Lattice Semiconductor). *Ultra-productive design tools for ultra-low density devices*. **2020**, Disponible en línea: <http://www.latticesemi.com/iCEcube2> (consultado el 21 de septiembre de 2021).
185. Micron-Technology-Inc. M25P40 3V 4Mb Serial Flash Embedded Memory. Disponible en línea: www.micron.com/-/media/client/global/documents/products/data-sheet/nor-flash/serial-nor/m25p/m25p40.pdf?rev=9762aaf3985f4b978574ebe826f994fe (consultado el 20 de agosto de 2021) **2011**, CCMTD-1725822587-8430, Rev. H 05/18.
186. Linear-Technology/Analog-Devices. IC Precision Oscillator Silicon - LTC1799CS. Disponible en línea: <https://www.analog.com/media/en/technical-documentation/data-sheets/LTC1799.pdf> (consultado el 20 de agosto de 2021) **2020**, Rev. E.
187. Texas-Instruments. CC2530 Second Generation - A True System-on-Chip Solution for 2.4 GHz IEEE 802.15. 4/RF4CE/ZigBee Applications. *CC2530F32, CC2530F64, CC2530F128, CC2530F256* **2011**, *SWRS081B*, Disponible en línea: <http://www.ti.com/lit/ds/symlink/cc2530.pdf> (consultado el 20 de agosto de 2021).
188. Texas-Instruments. CC2530 Development Kit User's Guide. *Texas-Instruments* **2010**, *SWRU208B*, Disponible en línea: <http://www.ti.com/lit/ug/swru208b/swru208b.pdf> (consultado el 27 de abril de 2022).
189. Myrra-Electronic. Transformers Datasheet—2.5 W and 5 W. **2015**, Disponible en línea: <http://www.acalbf.com/be/Power-supplies/AC-DC/Electronic-transformers/p/Electronic-transformer--Switching-power-supply--3-3V--2-5W--regulated/0000003LZW> (consultado el 20 de agosto de 2021).
190. Murata-Manufacturing-Company-Ltd. Converters MEU1S0315ZC. Disponible en línea: <https://www.murata.com/products/productdetail?partno=MEU1S0315ZC> (consultado el 20 de agosto de 2021). **2013**.