

UNIVERSIDAD DE LA REPÚBLICA

FACULTAD DE INGENIERÍA

COMUNICACIONES DIGITALES

---

## Proyecto — ZigBee

---

*Autores:*

Federico BELLO

Julieta UMPIERREZ

28 de septiembre de 2025



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Equipos y topología de la red</b>	<b>2</b>
<b>3. Modulación</b>	<b>3</b>
3.1. BPSK . . . . .	4
3.2. OQPSK . . . . .	4
3.3. <i>Chips</i> . . . . .	5
<b>4. Acceso al medio compartido</b>	<b>6</b>
<b>5. Conformación de tramas</b>	<b>6</b>
<b>6. Errores</b>	<b>7</b>
<b>7. Implementación</b>	<b>8</b>
<b>8. CSS: Chirp Spread Spectrum</b>	<b>9</b>
<b>9. Capa de Red</b>	<b>10</b>
<b>10. Capa de Aplicación</b>	<b>11</b>
<b>11. Conclusiones</b>	<b>11</b>

## 1. Introducción

Zigbee es un protocolo de comunicación inalámbrica ampliamente utilizado en dispositivos de Internet de las cosas (dispositivos IoT) como pueden ser hogares inteligentes, edificios inteligentes y automatización industrial. Esta tecnología permite la conexión de dispositivos de forma simple y eficiente, ya que utiliza un bajo consumo de energía, una baja tasa de transmisión de datos y tiene un bajo costo.

Zigbee funciona con el estándar IEEE 802.15.4, que especifica las capas física y de control de acceso al medio (MAC) para redes inalámbricas personales de baja velocidad (LR-WPAN). Los dispositivos pueden comunicarse entre sí utilizando una topología de red de malla, lo que significa que los dispositivos pueden comunicarse a través de otros cercanos, extendiendo el alcance de la red. Esto, junto con las características mencionadas previamente, hace a Zigbee ideal para diversas aplicaciones de IoT.

Zigbee tiene entonces como objetivo su utilización en redes inalámbricas de corta distancia y de baja transmisión de datos (hasta 250kbps). Utiliza las especificaciones de capa física y capa MAC brindadas por el estándar 802.15.4 y agrega la capa de red y de aplicación arriba de las anteriores.

Mas detalladamente algunas aplicaciones de Zigbee son:

- Seguridad en el hogar: Desde roturas de vidrios hasta detectores de movimiento.
- Medidores de consumo de agua, electricidad, etc que incluso detecten anomalías y actúen en consecuencia por ejemplo desconectando algún aparato que tenga un consumo desmedido o alertando al usuario para que chequee una pérdida, etc.
- Controles de iluminación, humedad y temperatura para implementar encendido de luces automático y sistemas de riego automáticos.
- Marcas activas de identificación por radio frecuencia que pueden funcionar a mayor distancia que las pasivas.

En particular, en el Instituto de Ingeniería Eléctrica de la Facultad de Ingeniería, Udelar, se han utilizado tecnologías similares (basadas en 802.15.4/6lowpan) en diversos ámbitos, algunos de estos incluyen:

- Monitoreo remoto en tiempo real de colmenas de abejas [10]
- Protocolo para el descubrimiento de vecinos (*DANDi*), el cual se implementa y experimenta con el uso del estándar [8]

## 2. Equipos y topología de la red

En el estándar 802.15.4 se definen dos tipos de equipos que pueden interactuar en este tipo de redes:

- Full Function Devices (FFDs): Son capaces de tomar cualquier rol en la red y de realizar cualquier función. Requieren mayor poder de procesamiento.
- Reduced Function Devices (RFDs): Tienen capacidades limitadas, solo se pueden comunicar con un FFD.

A su vez los FFDs pueden ser coordinadores o solo equipos comunes, en el caso de ser coordinadores deben poder recibir un mensaje y enviarlo a destino. Además, cada red debe tener un coordinador PAN (*personal area network*), el cual sera el encargado de administrar la red. Con esto en mente es que surgen diferentes maneras de organizar la red con tres topologías distintas:

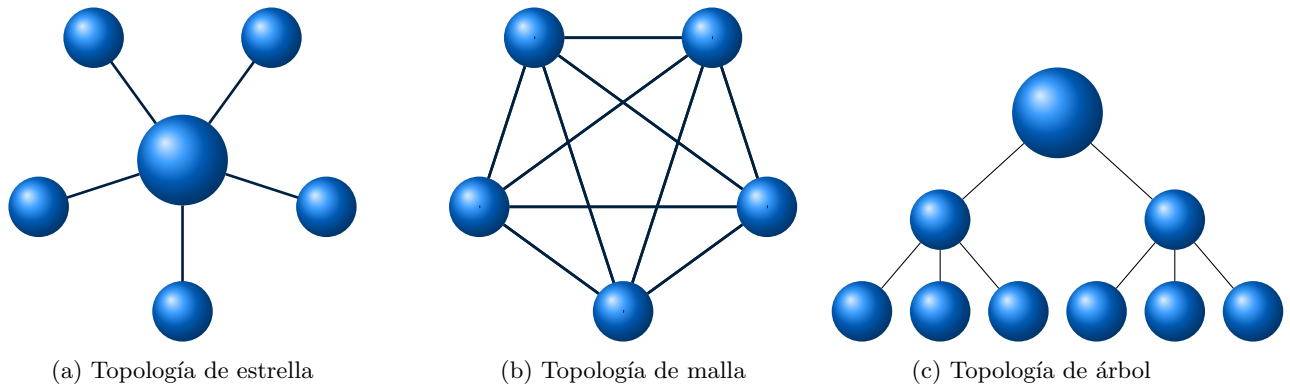


Figura 1: Comparación de topologías de red

- Topología en estrella: Todos los equipos solo se comunican con el coordinador y el coordinador envía al resto.
- Topología en malla: Hay equipos que se pueden comunicar entre si mientras estén lo suficientemente cerca. Si no lo están, hacen uso del coordinador o de otros FFD que le permitan llegar al otro dispositivo con el que se quieren comunicar.
- Topología *cluster-tree*: Es un caso específico de la topología en malla, donde la mayoría de equipos son FFDs y un RFD se conecta a la topología en un extremo de la misma. Cualquier FFD puede actuar de coordinador y proveer sincronización a este nuevo nodo, sin embargo solamente hay un coordinador PAN.

Estas topologías se muestran en la figura 1, donde el nodo mas grande que el resto corresponde al coordinador PAN. Las distintas topologías descritas van en concordancia con el objetivo de Zigbee de poder transmitir una baja tasa de datos en distancias cortas pero que permiten vincular distintos dispositivos para lograr las aplicaciones descritas en la introducción.

Según [5] el nombre Zigbee viene por la utilización de la topología de malla que permite llegar a lugares lejanos enviando mensajes a través de otros equipos que tiene una similitud con como se comunican las abejas. Cuando las abejas se quieren comunicar con la colmena y están a una distancia muy grande usan este pasaje de mensajes, para eso, realizan un baile llamado *zig-zag* que van repitiendo el resto de las abejas hasta que llega a la colmena. De ahí el nombre Zigbee.

### 3. Modulación

Esta tecnología utiliza esquemas de modulación distintos según la frecuencia de trabajo, como se especifica en el estándar IEEE 802.15.4. El estándar define 3 bandas de uso:

- De  $868\text{MHz}$  a  $868,6\text{MHz}$  (banda de  $868\text{MHz}$ )
- De  $902\text{MHz}$  a  $928\text{MHz}$  (banda de  $915\text{MHz}$ )
- De  $2400\text{MHz}$  a  $2483,5\text{MHz}$  (banda de  $2,4\text{GHz}$ )

La primer banda mencionada es utilizada principalmente en Europa, la segunda en Estados Unidos y la de  $2400\text{MHz}$  es de libre acceso. En particular, en Uruguay, además de utilizar la tercer banda se utiliza también la de  $915\text{MHz}$ , solo que se utiliza en la segunda región (de  $915\text{MHz}$  a  $928\text{MHz}$ ). Así lo fija el artículo 5 del capítulo 2 de la UIT. En la tabla 1 se muestra un resumen de las características de cada banda.

Banda de Frecuencia (MHz)	Modulación	Tasa de Bit por canal (Kbps)	Cantidad de canales
868	BPSK	20	1
915	BPSK	40	10
2400	OQPSK	250	16

Cuadro 1: Bandas de frecuencia de Zigbee

Notar entonces que la tecnología utiliza dos tipos de modulaciones distintas, dependiendo de la frecuencia de trabajo. Además, podría llegar a utilizar ASK, ya que el estándar tiene dos especificaciones opcionales para las bandas de 868MHz y 915MHz. Estas especificaciones opcionales son útiles para llegar a tasas de transmisión mayores sin necesidad de utilizar la banda de  $2,4GHz$ , sin embargo, los dispositivos que utilicen este sistema deben ser capaces de cambiar entre el sistema opcional y el obligatorio, cosa que puede ser difícil de implementar. La razón de haber agregado esta alternativa es que la banda de  $2,4GHz$  es compartida con el estándar IEEE 802.11b, utilizado por WiFi y *Bluetooth*, potencialmente generando interferencia. Además, las bajas frecuencias tienen la ventaja de penetrar mejor las paredes y otros objetos.

En la banda de  $2,4GHz$ , cada dispositivo puede utilizar un ancho de banda de como máximo  $2MHz$ , mientras que la separación entre dos canales contiguos es de  $5MHz$ .

A continuación se explica el funcionamiento de los distintos tipos de modulación que los dispositivos deben implementar, especificado por el estándar.

### 3.1. BPSK

Este tipo de modulación vista en el curso se basa en utilizar una portadora y un mapeo tal que la señal resultante quede desfasada  $180^\circ$  entre un símbolo y el otro, de esta forma, la modulación resulta en solamente dos estrellas, la cual se puede observar en la figura 2. Independientemente de la posición de los puntos en el espacio, es importante que tengan simetría central respecto al origen.

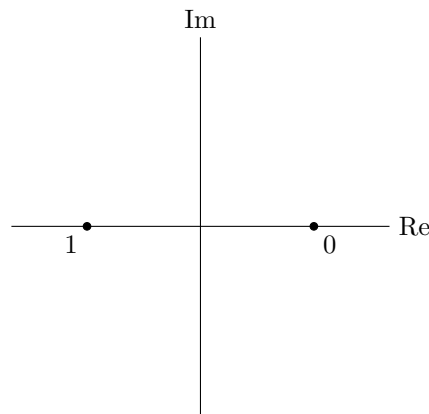


Figura 2: Constelación BPSK

### 3.2. OQPSK

Offset-QPSK es una modificación de QPSK en la que se busca la inmunidad a la distorsión de tercer orden. Similar a QPSK utiliza una portadora y un mapeo tal que la señal resultante de cada símbolo distinto tenga un defasaje de  $90^\circ$ , por lo que la constelación resultante es idéntica (figura 3). La diferencia surge en que, para evitar los cambios de símbolo en fase y cuadratura simultáneamente, se introduce un retardo en la rama real o imaginaria, obteniendo así que solamente cambie un bit a la vez. Al realizar este cambio, el

diagrama IQ se convierte solamente en un cuadrado (se eliminan las diagonales) por lo que la amplitud es *casi* constante, volviéndose *casi* inmune a la distorsión de tercer orden.

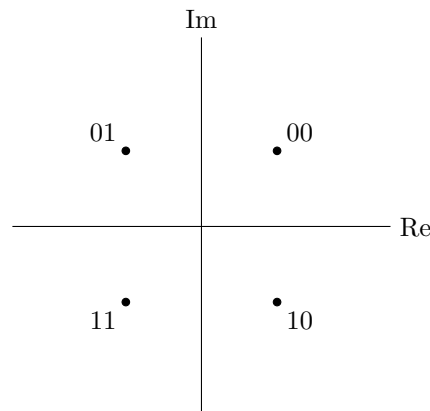


Figura 3: Constelación QPSK

Además, esto se usa en conjunto con un pulso conformador *half-sine*, el mismo se caracteriza por tener un ancho de banda efectivo bajo. Es decir, si bien tiene ancho de banda infinito, un gran porcentaje de la energía del mismo se encuentra en frecuencias bajas, siendo ideal para enventanar.

### 3.3. Chips

Un *chip* no es mas que la codificación de bits con otra secuencia binaria de mucha mayor frecuencia, como se puede ver en la figura 4. En este caso, se utiliza una secuencia de *chips* distinta para la representación de cada símbolo.

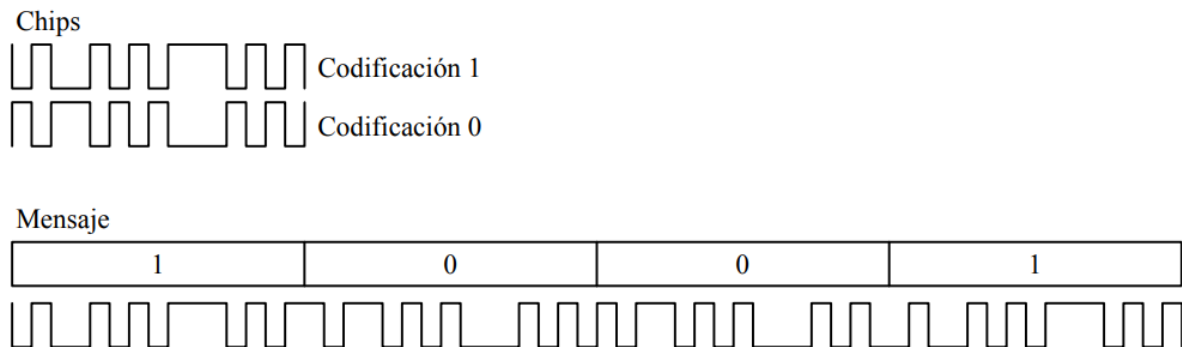


Figura 4: Ejemplo del uso de *chips*, extraído de [7]

La razón de ser de los mismos es que al codificar un símbolo con una secuencia de frecuencia mayor la transmisión se vuelve mas robusta al ruido. Por ejemplo, si se codifica al igual que la figura 4, (1  $\rightarrow$  010010101110101 y 0 con la secuencia opuesta), es decir, 15 *chips* por bit, se obtendría una distancia de Hamming de 15 por lo que sería necesario recibir incorrectamente al menos 8 de esos 15 *chips* para tomar la decisión incorrecta del bit enviado.

Por lo tanto, el esquema final de modulación se puede ver en la figura 5, donde el mismo es idéntico en el caso de estar en modulación BPSK solo que cambiando el respectivo modulador.

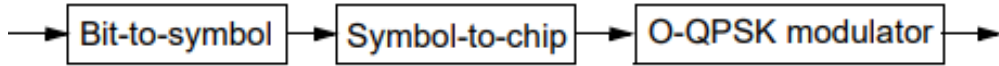


Figura 5: Mapeo y modulación [1]

## 4. Acceso al medio compartido

Esta tecnología utiliza como acceso CSMA-CA (acceso múltiple con escucha de portadora y evitación de colisiones). La idea es primero asegurarse que el canal esta libre y luego de asegurarse se comienza la transmisión. Para ver si esta libre se puede utilizar una etapa de ED (detección de energía) en la cual se sensa el canal para ver si hay un nivel de energía alto, el cual seria indicador de que el canal esta siendo utilizado. Otra opción es el sensado del carrier (CS) que es ver si se esta ocupando el canal con una señal del estándar, si se determina que así es el dispositivo espera por un tiempo (elegido como un random back off).

El algoritmo de CSMA-CA consiste en tres variables, el exponente *back-off* (BE), el numero de *back-offs* (NB) y una ventana de contención (CW). En el caso en el que el dispositivo sence un canal ocupado espera por un tiempo proporcional a un numero aleatorio entre 0 y  $2^{BE}$ , mientras que la variable NB se incrementa en uno. Esta ultima comienza en cero, y una vez que llegue a cierta constante (*maxMaxCSMABackoffs*) el algoritmo reporta una falla en el canal a la capa de red. Por ultimo, la variable CW indica la cantidad de sensados consecutivos que el dispositivo debe hacer y el resultado sea que el canal esta libre. Por ejemplo, si CW es 2, el dispositivo debe sensar el canal, estar libre, esperar un *random back-off* y repetir esta secuencia previo a poder trasmitir. Si alguna de las dos veces el canal no esta libre, se resetea la cuenta.

## 5. Conformación de tramas

De acuerdo al tipo de dato que se quiere intercambiar hay distintos tipos de trama MAC a enviar, beacon, datos, reconocimiento y comando MAC. De manera genérica todos tienen tres secciones: el encabezado (Mac header, MHR por sus siglas en ingles), el *payload* y el pie (Mac Footer, MFR por sus siglas en inglés). De manera genérica el frame se compone por los campos presentados en la figura 6 y se detallan a continuación.

- *Frame control*: Define el tipo de trama, si hay protección de seguridad, si hay datos pendientes de enviar del transmisor al receptor, si se requiere un reconocimiento por parte del receptor, compresión de la Personal Area Network (PAN) utilizada (que solo va si son distintas en las de fuente y destino), si se utilizan direcciones de 16 o 64 bits y la versión del estándar utilizada.
- *Sequence number*: Permite distinguir entre varias secuencias, por ejemplo para saber si alguna trama fue retransmitida.
- Direcciones de origen y destino: Son las direcciones de los equipos de la red. Puede ser de tamaño 0 y en ese caso se sabe que el mensajes es para el coordinador de la red.
- Identificadores PAN: Van cuando se comunican entre distintas PAN por lo que la bandera *PAN ID Compression* va en 0.
- *Security HDR*: Es un campo opcional que lleva información del nivel de seguridad y el tipo.
- *Frame payload*: Va la información a transmitirse.
- *Frame Check Sequence (FCS)*: Se detallará en una sección posterior.

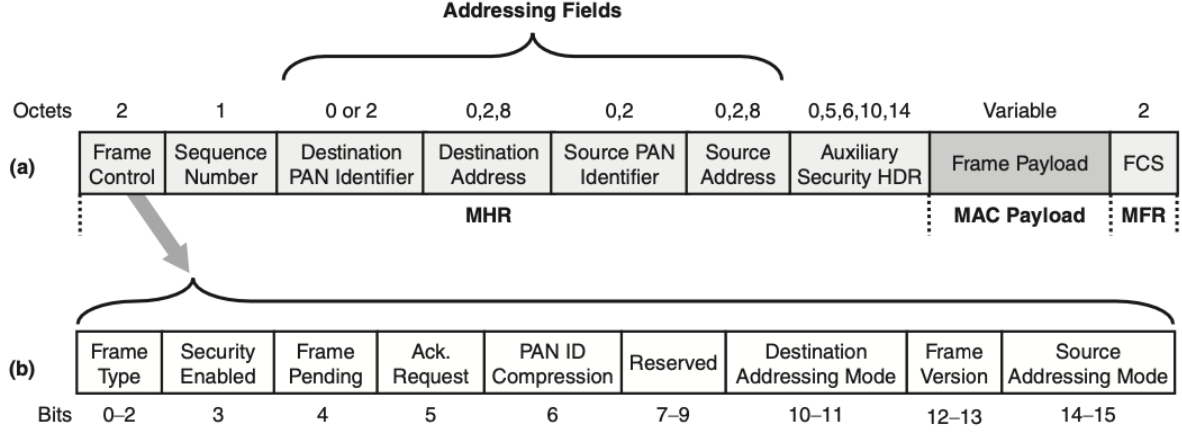


Figura 6: Trama MAC extraída de [5]

## 6. Errores

El estándar IEEE 802.15.4 utiliza un FCS (*frame check sequence*) de 16 bits basado en Códigos de Redundancia Cíclica (CRC) para detectar posibles errores en el paquete de datos. El polinomio utilizado es  $G_{16}(X) = X^{16} + X^{12} + X^5 + 1$ .

Luego, al momento de transmitir se transmite el mensaje en su forma sistemática, para esto es necesario seguir el siguiente algoritmo:

- Sea  $M(X) = b_0X^{k-1} + b_1X^{k-2} + \dots + b_{k-2}X + b_{k-1}$  el polinomio correspondiente a la tira de bits a transmitir.
- Calcular  $R(X)$ , que corresponde al resto de dividir  $X^{16}M(X)$  entre el polinomio generador  $G_{16}$ . Notar que  $R(X)$  tiene grado máximo  $X^{15}$ , mientras que  $X^{16}M(X)$  tiene grado mínimo  $X^{16}$ .
- Enviar el mensaje  $X^{16}M(X) + R(X)$ .

Por lo tanto, la palabra enviada siempre será múltiplo del polinomio generador. Teniendo esto en mente, al recibir un mensaje alcanza con dividir el mensaje por el polinomio generador, si el resto es cero se tiene una palabra de código válida, si es distinto a cero se puede afirmar que hubo un error en la transmisión.

El estándar también define un FCS de 4 octetos, aunque este no es el que utiliza *Zigbee*. Igualmente, el razonamiento es análogo al caso anterior.

Dado que en el párrafo anterior se presentó la etapa de detección de errores surge naturalmente preguntarse si *Zigbee* utiliza también un mecanismo de corrección de errores. En la forma estándar de *Zigbee* no se considera ningún tipo de código convolucional para la corrección de errores, esto que puede sin dudas resultar anti intuitivo puede explicarse de la siguiente manera: Zigbee se caracteriza por la baja tasa de transmisión y su alcance de cortas distancias por lo que re-enviar un paquete cuando se le detectó un error no parece ser algo extremadamente costoso.

Sin embargo en la literatura se ha expuesto varias veces como la performance de esta tecnología se vería mejorada si algún tipo de corrección de errores fuese incluida. Algunos ejemplos de esto son los artículos [2] y [4]. A su vez el estándar de 802.15.4 incluye la definición de un código convolucional de tasa  $\frac{1}{2}$ . Este código, cuyo diagrama de conexiones de tipo *shift-register* se presenta en la figura 7, tiene como función de transferencia en transformada-D la siguiente expresión:

$$[1 + D^2 + D^3 + D^5 + D^6, 1 + D^1 + D^2 + D^3 + D^6]$$



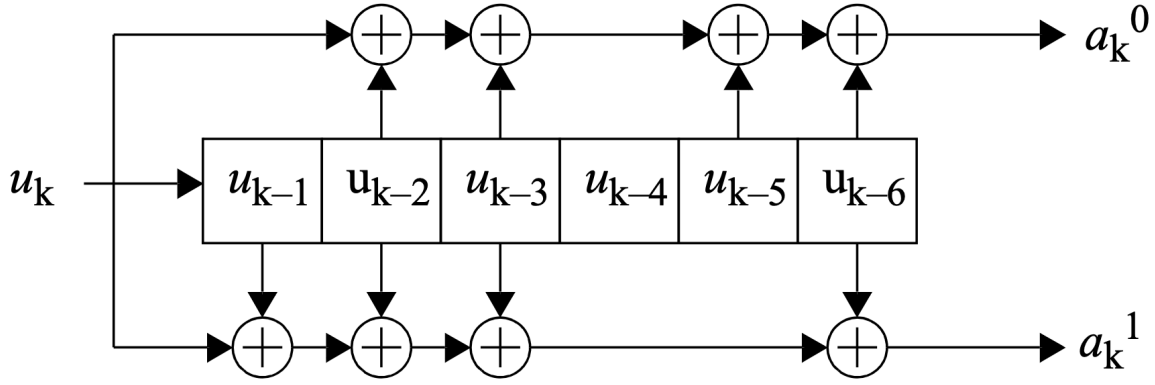


Figura 7: Diagrama tipo *shift-register* extraída de [1]

Dado que ya esta definido en el estándar este podría ser incluido en Zigbee en versiones futuras.

## 7. Implementación

Se explicará una de las posibles implementaciones de la capa física, extraída de [3]. En la figura 8 se observa el bloque de transmisión, donde la explicación se centrará en la modulación del mismo y no en la conformación del paquete. Los bloques *packed to unpacked* y *chunks to symbols* son los encargados de, dado un paquete, obtener el mapeo correspondiente para cierta secuencia de *chips*. En este caso particular, cada bloque de 4 bits es mapeado a una secuencia de 32 *chips*. El mapeo es idéntico al utilizado en QPSK (figura 3). Luego, se multiplica por el pulso conformador, en este caso se utiliza el *half sine*. Hasta este momento el sistema es idéntico a una modulación QPSK. Luego, se separa el complejo en parte real y parte imaginaria, se atrasa la parte imaginaria en un tiempo de *chip* y se vuelve a conformar el complejo. Esto ultimo le agrega el *off-set* a la modulación, evitando así las diagonales en el diagrama IQ.

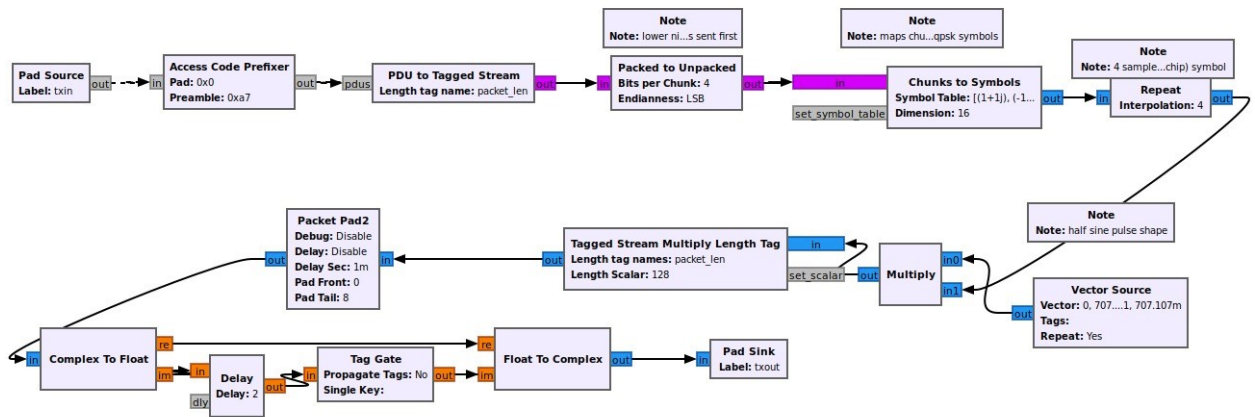


Figura 8: Posible implementación de transmisor OQPSK en GNU Radio

Mientras que el bloque de transmisión parece bastante natural, el bloque de recepción tiene algunas sutilezas. El mismo está representado en la figura 9, consiste en un bloque *quadrature demod*, cuya salida se resta con su salida luego de pasar por un bloque *single pole IIR filter*. El bloque *quadrature demod*, como

indica su nombre, demodula una señal modulada en la cuadratura del complejo, en este caso particular, en su fase. Si la entrada al mismo es  $x[n]$  la salida entonces es  $y[n] = \arg(x[n]\bar{x}[n-1])$ . Por lo tanto, si la entrada es de la forma  $x[n] = Ae^{j(2\pi T_s f_\Delta n + \phi[n])}$  su salida sera:

$$y[n] = \arg(Ae^{j(2\pi T_s f_\Delta n + \phi[n])} Ae^{-j(2\pi T_s f_\Delta (n-1) + \phi[n-1])}) = \arg(e^{j(2\pi T_s f_\Delta n + \phi[n] - 2\pi T_s f_\Delta (n-1) - \phi[n-1])})$$

$$y[n] = (\phi[n] - \phi[n-1] + 2\pi T_s f_\Delta)$$

Por lo tanto, luego del primer bloque se obtiene la información deseada a menos de una constante aditiva. Aquí entra en juego el bloque *single pole IIR filter*, el mismo es un pasabajos de frecuencia de corte bastante estricta, en particular, se supondra que solamente se queda con la componente en continua de la entrada. En este caso,  $2\pi T_s f_\Delta$ . Luego, al hacer la resta entre ambos resultados, se logra demodular correctamente la señal  $(\phi[n] - \phi[n-1])$ . Luego, en el receptor, el bloque *clock recovery MM* es el encargado de aplicar el algoritmo de Müller and Mueller [9] para recuperar la secuencia de *chips* correspondiente.

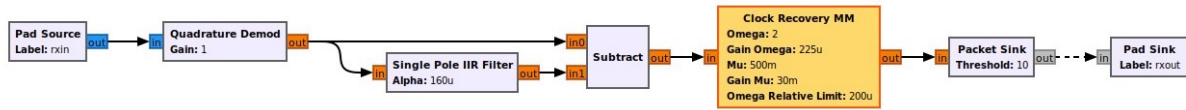


Figura 9: Posible implementación receptor OQPSK en GNU Radio

Vale recalcar que el receptor anterior puede ser mejorado siguiendo las ideas planteadas en [6], alcanza con agregar un filtro pasabajos previo al *quadrature demod* y un *decimating FIR filter* luego de la resta para mejorar considerablemente el desempeño del receptor.

## 8. CSS: Chirp Spread Spectrum

En la sección anterior se presentó la implementación existente de 802.15.4 en GNU radio. Dentro de estos archivos se encontraba el loopback que permitía generar y recibir mensajes usando 802.15.4. En este caso se implementaba una capa física distinta a la mencionada en Zigbee que llamó nuestra atención. Esta utilizaba Chirp Spread Spectrum (CSS) que se describirá a continuación.

En primera instancia este modo de funcionamiento de la capa física incluye la utilización de modulación DQPSK es decir QPSK diferencial. La idea de DQPSK es hacer que los símbolos tengan una cierta fase relativa al símbolo anterior. En recepción la idea es demodular y con un comparador obtener la diferencia entre la fase de la señal recibida y la fase de la predecesora para obtener el símbolo enviado. Esto es muy útil para evitar problemas de coherencia entre los osciladores en transmisión y recepción.

Luego de tener los símbolos DQPSK estos se modulan en sub-chirps. Una señal chirp es una señal cuya frecuencia crece (o decrece) con el tiempo como se ejemplifica en la figura 10. La idea detrás de Chirp Spread Spectrum (CSS) es usar señales de tipo chirp para enviar información utilizando todo el ancho de canal posible. Es robusto al ruido del canal, al fading causado por el multipath, incluso cuando se transmite con baja potencia y al efecto doppler.

En el caso específico definido para la capa física en 802.15.4 se tiene que cada símbolo DQPSK ahora se envía en un sub chirp por lo que cada señal enviada es un sub chirp multiplicado por el símbolo DQPSK de magnitud y fase constante. Luego, se conforma un símbolo chirp con 4 sub chirps. Cuatro posibilidades se adjuntan en la figura 11 en forma de diagramas tiempo-frecuencia.

Teniendo en cuenta las ventajas de utilizar señales de tipo chirp esto podría ser otra adición a futuras ediciones de *Zigbee*.

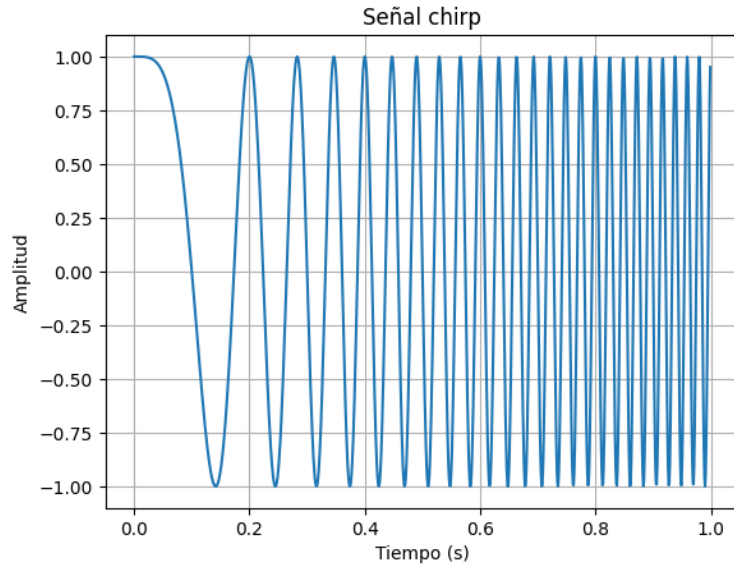


Figura 10: Ejemplo de señal chirp

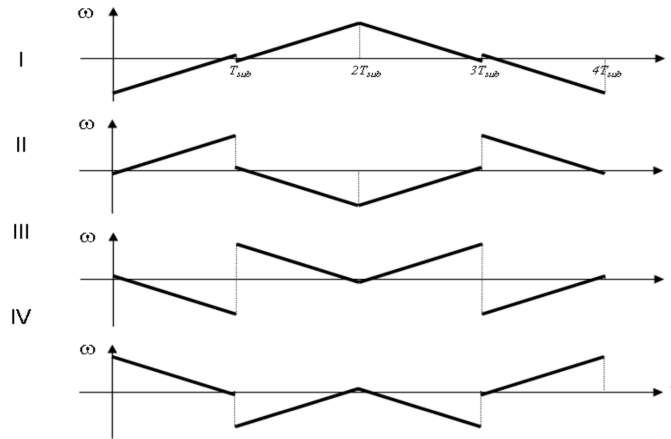


Figura 11: Diagramas tiempo-frecuencia de cuatro posibles secuencias chirp extraída de [1]

## 9. Capa de Red

Como se mencionó previamente, Zigbee toma la capa física y la capa MAC del estándar 802.15.4 de la IEEE, la capa de red y la capa de aplicación se definen específicamente para esta tecnología.

La capa de red es la encargada de manejar la red y realizar el ruteo. Está encargada de asignar un número de 16 bits como identificador de cada dispositivo. Esto es realizado en el equipo que coordina la red, y el identificador debe ser igual a la dirección de 16-bit IEEE 802.15.4 MAC del dispositivo. Algo curioso es que en la topología de malla o de árbol también agrega el número de saltos que puede dar un mensaje Si antes de llegar a destino se pasa de ese numero de saltos se deja de enviar, consiguiendo que la red no se congestione. La idea es simple, cada paquete tiene un parámetro llamado *radio*, el mismo toma un valor inicial y cada vez que pasa por un nodo se lo decrementa en uno. Una vez que llega a cero, el paquete se descarta.

La tecnología soporta tres tipos distintos de mensaje: *broadcast*, *multicast* y *unicast*. El *broadcast* es un mensaje destinado a todos los nodos dentro de una red, el *unicast* uno destinado solamente a uno y el *multicast* es un punto medio entre ambos, donde se envía el mensaje a un grupo de dispositivos.

- *Broadcast*: Por una cuestión de eficiencia cuando se envía un mensaje broadcast (enviado por el coor-

dinador con destino todos los dispositivos de la red) el coordinador solo verifica que los dispositivos vecinos hallan recibido el mensaje utilizando el reconocimiento pasivo, es decir no recibe el reconocimiento de todos los elementos de la red porque seria muy ineficiente. Durante este tipo de mensaje, el paquete pasa por varios dispositivos, por lo que existe la posibilidad de encontrarse con el problema del nodo oculto. Para disminuir los efectos negativos de este problema, previo a cada retransmisión cada dispositivo debe esperar un tiempo aleatorio, este tiempo es conocido como *broadcast jitter*.

- *Multicasting*: Notar que se puede obtener el mismo resultado con cierta cantidad de mensajes *unicast* seguidos, aunque hacer uso de esta funcion es mas eficiente. Se definen grupos de *multicast*, donde cada grupo es identificado con un ID de 16 bits. Un nodo puede pertenecer a mas de un grupo *multicast*. No es necesario que un nodo pertenezca a un cierto grupo para enviar un mensaje *multicast* a ese grupo. En caso de no pertenecer, existe la posibilidad de simplemente enviar un mensaje de tipo *unicast* a un nodo que si pertenezca, el cual se encargará de realizar la trasmisión *multicast*.

En cuanto al ruteo el objetivo es minimizar una función de costo para que llegar a destino con un mensaje sea lo menos costoso posible. Ahora es claro que hay muchas maneras de definir una función de costo, *Zigbee* define el costo de cada link como una función de la probabilidad de entrega exitosa  $p_l$ ,  $C(l) = \text{Min} \left[ 7, \text{round} \left( \frac{1}{p_l} \right) \right]$ , donde esa probabilidad se puede estimar de varias maneras pero todas usan un estimador calculado a nivel de capa MAC que se llama *link quality indicator*. Luego, la función de costo es la suma de el costo de cada link en ese camino. Por ultimo, teniendo el costo de cada camino es que el coordinador escribe una tabla de ruteo que indica los caminos a seguir.

## 10. Capa de Aplicación

Por ultimo se encuentra la capa de aplicación. La misma se divide en tres subcapas, la subcapa APS (*Application Suport*, ZDO (*Zigbee Device Objects*) y el *framework* de aplicación.

- APS: Es la subcapa encargada de proveer una interfaz entre la capa de red y la capa de aplicación. La misma debe, entre otras cosas: mantener las tablas de enlace, enviar los mensajes entre dispositivos enlazados, administrar las direcciones de grupos y hacer el mapeo entre la dirección de 64 bits de IEEE y la dirección de 16 bits de la capa red.
- ZDO: utiliza los servicios de la capa de red y la subcapa APS para: definir el rol de un dispositivo en la red, descubrir los dispositivos en la red y su aplicación, iniciar o responder a pedidos de enlace. Además, es la encargada de implementar las tareas de seguridad.
- *Framework*: es el ambiente en el cual se encuentran alojados los objetos de la aplicación.

## 11. Conclusiones

Se logró poner en practica las herramientas aprendidas a lo largo del curso, teniendo como objetivo principal estudiar a fondo una tecnología en el contexto de comunicaciones digitales. Se exploraron las peculiaridades de la tecnología *Zigbee* así como las del estándar IEEE 802.15.4. Se vieron los casos de uso, ventajas, desventajas y limitaciones de esta tecnología así como posibles futuras variantes que podrían ser obtenidas de las definiciones realizadas en el estándar IEEE 802.15.4.

## Referencias

- [1] Ieee 802.15.4-2015 - IEEE standard for low-rate wireless personal area networks (lr-wpans). <https://ieeexplore.ieee.org/document/9144691>, 2015.
- [2] Hikmat Abdullah, Tariq Salman, and Haider Alobaidy. Performance evaluation of zigbee transceivers using convolutional coding technique. *Communications on Applied Electronics*, 3:32–41, 11 2015.
- [3] Bastian Bloessl. gr-ieee802-15-4: GNU Radio IEEE 802.15.4 Implementation. <https://github.com/bastibl/gr-ieee802-15-4>, 2020.
- [4] Zdenek Bohuslavsek and Ilja Masik. Error characteristics and their prediction in zigbee transmission at coexistence conditions. *WSEAS Transactions on Communications*, 16, 2017.
- [5] Shahin Farahani. *ZigBee Wireless Networks and Transceivers*. Newnes, 2011.
- [6] Evan Faulkner, Zelin Yun, Shengli Zhou, Zhijie J. Shi, Song Han, and Georgios B. Giannakis. An advanced GNU radio receiver of IEEE 802.15.4 OQPSK physical layer. *IEEE Internet of Things Journal*, June 2021.
- [7] Irene Tolosa Florencia Montaldo, Federico Sierra. Parriot: Implementación de rtk sobre red lora/lorawan. Proyecto de fin de carrera, Universidad de la República, Montevideo, Uruguay, Noviembre 2019.
- [8] Nicolás Gammarano, Javier Schandy, and Leonardo Steinfeld. Reducing neighbor discovery time in sensor networks with directional antennas using dynamic contention resolution. *Design Automation for Embedded Systems*, 24(4):223–247, March 2020.
- [9] K. Mueller and M. Muller. Timing recovery in digital synchronous data receivers. *IEEE Transactions on Communications*, 24(5):516–531, May 1976.
- [10] Juan Navarro, Fabio Lima, Marín Porto, and Leonardo Steinfeld. BeeSmart: a real-time remote monitoring and control system for beekeeping. In *2022 Symposium on Internet of Things (SIoT)*. IEEE, October 2022.