Índice

Resumen

Abstract

1. Introducción
   1. Objetivos
   2. Desarrollo del trabajo
   3. Estructura de la memoria
2. Redes de sensores inalámbricas cognitivas
   1. Redes de sensores inalámbricas
   2. Redes cognitivas
   3. Redes de sensores inalámbricas cognitivas
      1. Nodos para CWSN
3. Estudio previo
   1. Hardware del cNGD
   2. Firmware del cNGD
   3. Arquitectura cognitiva
   4. Algoritmos a implementar
4. Implementación del algoritmo de seguridad
   1. Funciones de la arquitectura cognitiva
      1. Optimizer
      2. Discovery
      3. ………..Todos los submódulos que utilice el algoritmo
   2. Comentarios (o posibles modificaciones) a la implementación propuesta
5. Implementación del algoritmo de reducción de consumo
   1. Funciones de la arquitectura cognitiva
      1. Optimizer
      2. Discovery
      3. …………. Todos los submódulos que utilice el algoritmo
   2. Comentarios (o posibles modificaciones) a la implementación propuesta
6. Aplicación de prueba de los algoritmos
   1. Diseño
   2. Implementación
7. Conclusiones y líneas futuras
   1. Conclusiones
   2. Líneas futuras

Bibliografía

Lista de acrónimos

## Capítulo 1. Introducción

En este capítulo se van a llevar a cabo una introducción al trabajo en la que se describirán los objetivos principales del trabajo y se detallará el proceso para desarrollarlo. Por último, se describirá la estructura de esta memoria.

## 1.1. Objetivos

Aksdlfh

## 1.2. Desarrollo del trabajo

El trabajo se ha dividido en las siguientes etapas:

* Estudio previo. El primer paso tomado para la consecución de este trabajo ha sido la adquisición de conocimientos sobre las redes de sensores inalámbricas cognitivas y con el entorno de trabajo.
  + Estudio de trabajos realizados anteriormente sobre CWSN en el laboratorio y de la documentación facilitada sobre los algoritmos que se han implementado.
  + Familiarización con las herramientas de programación de Microchip, en concreto MPLAB X, y revisión de los conceptos de programación de microcontroladores en C.
* Desarrollo. Consiste en la implementación de los dos algoritmos sobre los nodos disponibles en el laboratorio. Tras la implementación de los dos algoritmos se ha desarrollado una aplicación que sirva de demostración del correcto funcionamiento del código implementado.
* Pruebas. Se han realizado las pruebas necesarias para comprobar el comportamiento de los algoritmos en el nodo. Por ejemplo, introduciendo una fuente de ruido cerca de un nodo para comprobar que se inicia el algoritmo de reducción de consumo.
* Documentación. La escritura de esta memoria ha sido simultánea a las etapas anteriores y tras terminar la implementación de cada algoritmo.

## 1.3. Estructura de la memoria

## Capítulo 2. Redes de sensores inalámbricas cognitivas

En este capítulo se resumirán las principales características de las redes de sensores inalámbricas (WSN, en sus siglas en inglés) para, a continuación, introducir una evolución de éstas denominada CWSN (Cognitive Wireless Sensor Networks). Por último, se presentarán algunos de los nodos para CWSN que existen en la actualidad y sus principales características.

## 2.1. Redes de sensores inalámbricas

Una red de sensores es aquella formada por una serie de dispositivos con acceso a información del medio cuya misión es la de monitorizar diferentes parámetros del entorno. Las WSN están formadas por dispositivos con conectividad inalámbrica, lo que les da mayor versatilidad y flexibilidad. El número de nodos que forman una red de este tipo puede variar desde unos pocos hasta varios cientos y pueden conectarse siguiendo diferentes topologías como podemos ver en la Figura 2.1.

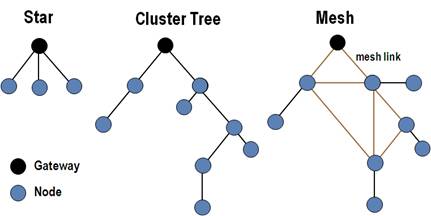


Figura **¡Error! No hay texto con el estilo especificado en el documento.**.1 Topologías de red WSN, obtenida de [1]

En todas estas topologías de red existe un nodo, con más recursos que el resto, que hace de puerta de enlace o de coordinador para el resto de nodos, como es el caso de las redes *mesh*.

En cuanto a las diferentes tecnologías y protocolos de comunicación implementados en WSN la mayoría están basados en el estándar 802.15.4 del IEEE (*Institute of Electrical and Electronics Engineers*) [2] para WPAN (Wireless Personal Area Network). Uno de los protocolos basados en este estándar es MiWi™ [refMiWi] y es el que incorpora el nodo con el que vamos a trabajar.

Otro estándar muy utilizado es el IEEE 802.11 [ref802.11] en el que está basado Wi-Fi [refWiFi]. Debido a la extensión en el uso de Wi-Fi, otro trabajo que se está desarrollando en el LSI está dando conectividad mediante este estándar a los nodos.

## 2.2. Redes cognitivas

## 2.3 Redes de sensores inalámbricas cognitivas

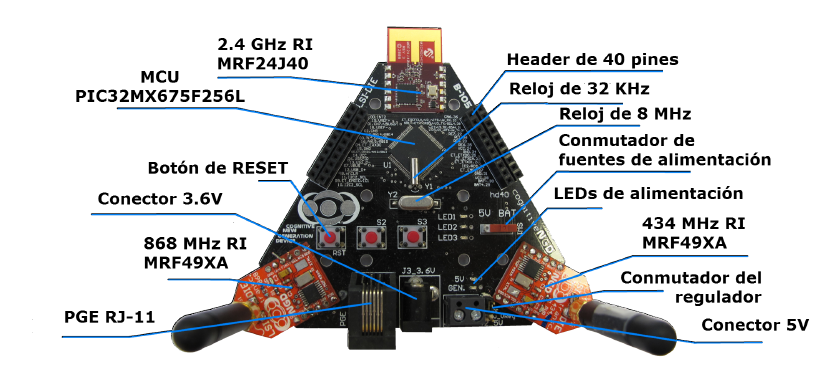
### 2.3.1. Nodos para CWSN

## Capítulo 3. Estudio previo

En este capítulo se va a presentar la plataforma sobre la que se ha realizado el trabajo. Se detallarán tanto el hardware como el firmware del que disponen los nodos y la arquitectura cognitiva donde se ha desarrollado este trabajo. Por último, se presentarán los algoritmos que han sido el objetivo de este trabajo.

### 3.1. Hardware del cNGD

El hardware sobre el que se ha desarrollado en este trabajo es el nodo cNGD desarrollado en el LSI y el cual viene detallado en [1]. Aquí expondremos brevemente algunas de sus características más importantes y que se han tenido en cuenta a la hora de realizar el trabajo ya que influyen a la hora de implementar código sobre el nodo.



(a) Vista superior.

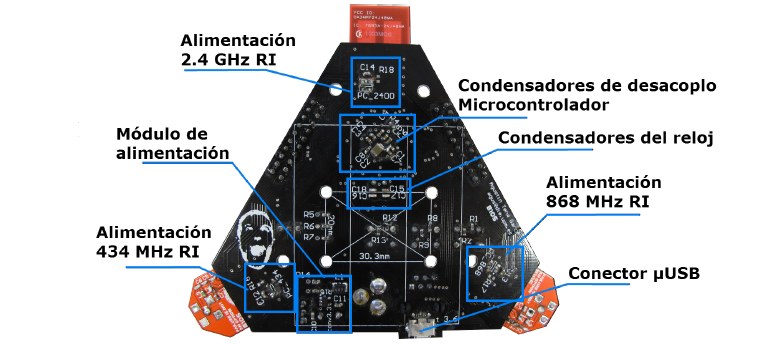


Figura **¡Error! No hay texto con el estilo especificado en el documento.**.a Vista detallada del cNGD.

El hardware del nodo tiene que cumplir unos requisitos necesarios en cuanto a consumo, bajos recursos, bajo coste y varias bandas de frecuencias para las comunicaciones. Por tanto pasamos a describir algunas de sus características principales:

* Microcontrolador. El MCU que incorpora el nodo es el PIC32MX675F256L [2], de 32 bits y fabricado por Microchip. Tiene 100 pines y sus características son:
  + Memoria. 256 kB de memoria flash y 64 kB de memoria RAM.
  + Reloj interno. Frecuencia máxima de funcionamiento de 80 MHz.
  + Modos de funcionamiento. Varios modos para reducir el consumo.
  + Timers. Cinco timers de 16 bits, pudiendo utilizar dos de ellos para hacer uno de 32 bits.
* Interfaces radio. El nodo dispone de tres interfaces radio, por lo que es capaz de transmitir y recibir a través de tres frecuencias diferentes. Éstas son 434 MHz, 868 MHz y 2,4 GHz. Con esto cubre todas las bandas ISM de Europa. Para poder reducir el consumo de los nodos sin tener que renunciar a tener las tres interfaces, éstas se pueden activar o desactivar cuando no se estén utilizando.
* Alimentación. El nodo tiene varias posibilidades de alimentación, siendo la principal las baterías. También se puede alimentar a través de USB, del conector RJ-11 o del conector de 3.6 V.
* Módulos de expansión. El nodo tiene la posibilidad de expandir su funcionalidad mediante módulos de expansión que se conectan a los *headers* disponibles. Uno de los que se han utilizado en este trabajo es el que permite comunicarse a través de línea serie RS232, permitiendo comprobar la funcionalidad del código.

Como vemos, las características del cNGD satisfacen las especificaciones necesarias para un nodo para CWSN, ya que es capaz de trabajar en diferentes bandas de frecuencia, en este caso todas las bandas ISM de Europa, es capaz de trabajar en modos de bajo consumo tanto con los diferentes modos de funcionamiento del microcontrolador como apagando las interfaces radio que no utilice. Además, permite el desarrollo de nuevas funcionalidades mediante los módulos de expansión.

### 3.2. Firmware del cNGD

El firmware implementado en el nodo y que fue desarrollado en [3] tiene la función de optimizar, adaptar e integrar las pilas de protocolos de cada uno de los transceptores en una única pila, y de proporcionar una interfaz que simplifica el trabajo del programador mediante una serie de funciones que son las que acceden al hardware.

La pila de protocolos de los transceptores se resume en la Figura 3.2, obtenida de [3], donde se ve que entre la aplicación y el transceptor sólo se pueden comunicar mediante paquetes, registros y mensajes.

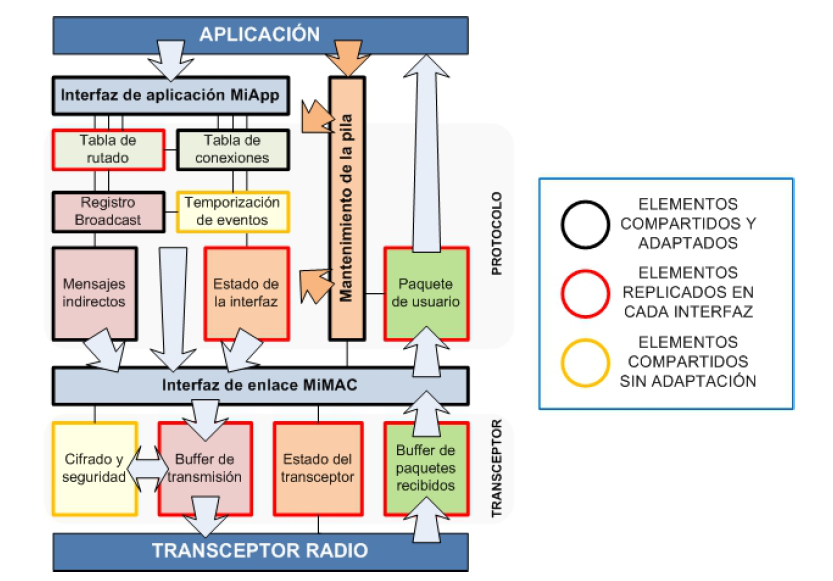


Figura 2

Con la adaptación de [3], que denominamos HAL, lo que se consigue es simplificar la labor del programador a la hora de utilizar la pila de protocolos de las interfaces, teniendo que hacer llamadas a funciones para acceder a los recursos y, además, la HAL da flexibilidad para cambiar los transceptores, añadir nuevos o implementar nuevas funcionalidades. Todo esto queda resumido en la Figura 3.3.

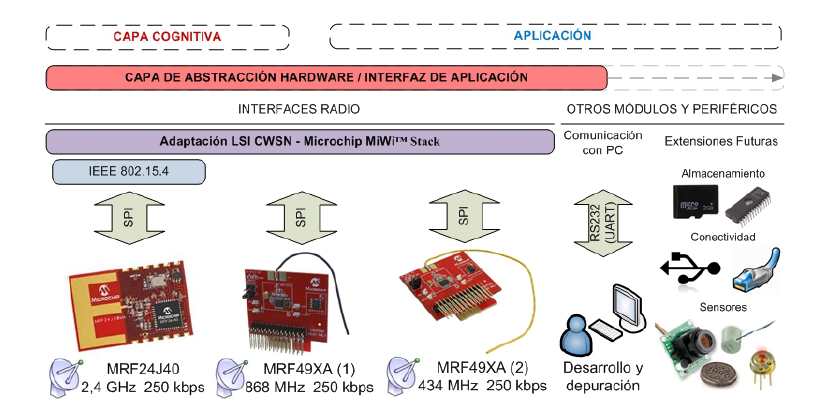
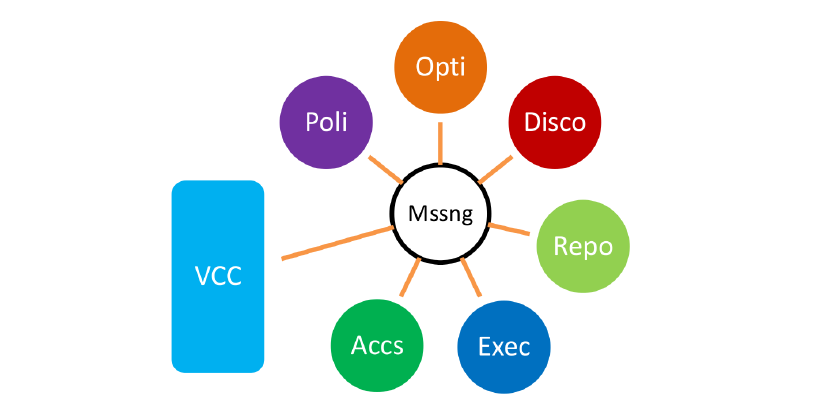


Figura 2

Las funciones implementadas actualmente en la HAL van desde la inicialización del nodo hasta la gestión de las comunicaciones (saber el canal activo en una interfaz, enviar y recibir paquetes, comprobar la tabla de conexiones, etc.).

### 3.3. Arquitectura cognitiva

La implementación de la arquitectura cognitiva realizada en [4] es la encargada de dar soporte a la implementación de estrategias cognitivas en el nodo. El esquema general de la arquitectura se puede ver en la Figura 3.4.



La función de cada sub-módulo de la arquitectura es:

* Repository. Sub-módulo encargado de almacenar la información necesaria para la estrategia cognitiva. Aquí se recibirá la información y se almacenará para cuando se la pida cualquier sub-módulo de la arquitectura.
* Discovery. Es el encargado de caracterizar diferentes parámetros del entorno. Este sub-módulo será el encargado de obtener el nivel de ruido en los canales.
* Optimization. En este sub-módulo se implementarán las rutinas de las estrategias cognitivas. Mientras este sub-módulo realiza el proceso cognitivo podrá realizar peticiones a otros sub-módulos del CRModule o incluso a otros sub-módulos en otros nodos.
* Execution. Los resultados del proceso de optimización tendrán que ser ejecutados. Este sub-módulo es el encargado de ejecutar las decisiones tomadas por el sub-módulo Optimization.
* Access Control. Debido a la naturaleza cooperativa de las estrategias cognitivas, son necesarios mecanismos de seguridad y control que sepan qué nodos tienen permisos para hacer acciones sobre el resto de nodos de la red. Éste sub-módulo se encarga de manejar la información de los permisos que tienen los nodos conocidos para realizar acciones en los sub-módulos del CRModule al que pertenece.
* Policy Support. Las estrategias cognitivas tienen unos valores que determinan las decisiones que se toman. Por tanto, este sub-módulo tiene la información sobre esos valores y es consultado por el resto de sub-módulos, sobre todo por Optimization, para tomar las decisiones oportunas.
* Messenger. Es el sub-módulo central de la arquitectura. Se encarga de conectar el resto de sub-módulos entre ellos. Maneja los mensajes que se mandan el resto de sub-módulos y comprueba, si el mensaje proviene de otro nodo, si tiene permisos o no mediante petición al sub-módulo Access Control.

La implementación realizada en [4] fue desarrollada para funcionar directamente sobre la pila de protocolos de Microchip y para funcionar sobre una plataforma hardware distinta. Por ello, posteriormente, en [5], se realizó una adaptación de esta arquitectura para funcionar con la pila de protocolos y el firmware mencionados en el apartado anterior.

### 3.4. Algoritmos a implementar

A continuación se resumirán los algoritmos que se han implementado en la realización de este trabajo, dejando los detalles y el resultado de las simulaciones para la consulta en los documentos referenciados.

#### 3.4.1. Algoritmo de seguridad

Este algoritmo, desarrollado en [6], consiste en evitar que un nodo se haga pasar por usuario primario de la red, denegando el uso de la misma a otros nodos.

Para la detección de los nodos intrusos, o que tienen un funcionamiento anómalo, el algoritmo requiere de dos fases:

* Fase de aprendizaje. En esta fase el nodo procesa los paquetes que recibe, guardando el valor del RSSI y el tiempo que ha transcurrido entre dos paquetes. Con esto construye una lista de clústers que tienen de coordenadas el valor del RSSI y del tiempo entre paquetes y un radio. Para la creación de los clústers primero se normalizan los valores almacenados y luego se van procesando cada par de coordenadas incluyéndolas en un clúster.
* Fase de detección. Es el tiempo restante que el nodo esté trabajando. En esta fase el algoritmo se encargará de coger el valor del RSSI y tiempo entre paquetes de los paquetes que va recibiendo y comprobando que están contenidos en un clúster de los anteriores. Si un paquete no está contenido en un clúster, se marca el nodo del que procedía como atacante y se informa al resto de nodos de la red. Si un número determinado de nodos detectan a un mismo nodo como atacante desconectan ese nodo de la red.

#### 3.4.2. Algoritmo de reducción de consumo

Este algoritmo, detallado en [7], trata de reducir el consumo de los nodos de la red mediante la teoría de juegos.

El algoritmo aprovecha que es más costoso transmitir por un canal ruidoso que cambiar todos los nodos a un canal con mejor calidad de enlace. Por tanto, el algoritmo consiste en conocer el estado del canal por el que se está transmitiendo mediante el RSSI de los paquetes que se reciben, si se baja de un umbral predefinido, se lanzará el proceso de decisión de cambio de canal. La decisión se toma calculando los costes asociados a transmitir por un canal ruidoso, con un número de retransmisiones por paquete, y el coste de sensar el medio y enviar mensajes para cambiar de canal. Si se decide cambiar de canal, el coste de cambiar es menor que el coste de transmitir en el canal en el que se está transmitiendo, se procede a elegir el canal menos ruidoso y a decidir entre todos los nodos de la red a qué canal se cambia definitivamente.

# Implementación del algoritmo de reducción de consumo

En este capítulo se va a detallar el trabajo desarrollado para la implementación del algoritmo desarrollado en [refPaperElena]. Se explicarán las nuevas funciones añadidas en el CRModule y los mensajes y respuestas que se intercambian los nodos para conseguir el cambio a un canal más óptimo para la transmisión de mensajes de aplicación.

## 5.1. Funciones de la arquitectura cognitiva

La implementación de este algoritmo se ha realizado siguiendo un esquema de máquinas de estados finitos tal y como se presenta en la Figura 5.1.

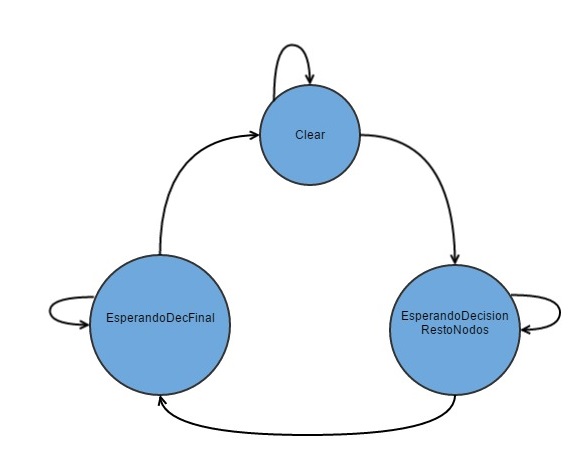


Figura 5.1 Diagrama de estados de la implementación propuesta

Las transiciones entre estados de la Figura 5.1 dependen de las respuestas de los otros nodos a las peticiones de cambio de canal que se les hace. Todos los nodos comenzarán en estado “Clear” en el que comprobarán, con cada mensaje de aplicación que envíen, el número de retransmisiones que se han realizado. Tras enviar el mensaje de aplicación y en la ejecución de la rutina de atención a la interrupción del *timer* 4, que se produce cada milisegundo, el nodo calcula los costes asociados a la transmisión en el canal en el que está transmitiendo y los costes asociados al cambio de canal. Cuando el coste de cambio es menor que el coste de transmitir en el canal actual, ese nodo inicia el proceso de elección del canal al que cambiar y notifica al resto de nodos en su red.

El paso de mensajes de petición de cambio de canal y las respuestas se producen a través de VCC, habiendo reservado la interfaz de 434 MHz disponible en los nodos para tal efecto.

A continuación se pasa a describir las funciones de cada uno de los sub-módulos del CRModule para procesar las peticiones de cambio y obtener el resultado final.

### 5.1.1. Optimizer

El sub-módulo Optimizer es el encargado de la ejecución del algoritmo, decidirá el inicio del cambio de canal, procesará las respuestas de los otros nodos de la red y pedirá al resto de sub-módulos la ejecución o la información que necesite durante la ejecución.