Universidad de Huelva

Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática

Sensores para seguridad y fiabilidad en sistemas de comunicaciones

Memoria para optar al grado de doctor
presentada por:

Jonatan Medina García

Fecha de lectura: 23 de marzo de 2018

Bajo la dirección de los doctores:

Juan Antonio Gómez Galán

Raúl Jiménez Naharro

Huelva, 2018

Universidad de Huelva

Tesis Doctoral

Jonatan Medina García

SENSORES PARA SEGURIDAD Y FIABILIDAD EN SISTEMAS DE COMUNICACIONES

Universidad de Huelva

Departamento de Ingeniería Electrónica,

de Sistemas Informáticos y Automática

Programa de Doctorado

Ciencia y Tecnología Industrial y Ambiental

Tesis Doctoral

SENSORES PARA SEGURIDAD Y FIABILIDAD EN SISTEMAS DE COMUNICACIONES

Autor

Jonatan Medina García

Directores

Dr. Juan Antonio Gómez Galán

Dr. Raúl Jiménez Naharro

Huelva, febrero de 2018

Gracias por soportar el tiempo que os he quitado, gracias por estar siempre ahí, a mis padres, hermanos, esposa e hijos.

Agradecimientos

En agradecimientos, no me gustaría dejar fuera a nadie de las personas que me han ayudado a ver este mundo, un mundo de tecnología, innovación y sobre todo sufrimiento. Un sufrimiento menospreciado en muchas ocasiones; creo que ese grupo de personas que han estado ahí se merecen que hoy sean recordadas aquí.

No voy a poner posiciones, sino motivos por estos agradecimientos. Al Dr. Fernando Gómez Bravo, por su inquietud y fuerza a que las cosas salgan adelante. Al Dr. Raúl Jiménez Naharro por su trabajo y colaboración. Y para terminar, a un gran amigo, el Dr. Juan A. Gómez Galán, que me abrió una puerta para que terminara unos estudios que inicié; hay que saber que esto lo hizo a sabiendas de que yo no era un alumno cualquiera, sino que llevaba una pequeña mochila a cuestas.

También quiero agradecer a esos amigos que hoy no están dentro de la universidad, pero que bajo mi humilde punto de vista, son grandes en sus labores. Gracias JJ (Juan José Chica Barrera) por compartir esas tardes de sufrimiento, y Paco (Francisco José Aguilar Nieto), amigo, hermano, padre, pocos son los adjetivos que encuentro para describir mi relación contigo; hemos sufrido, trabajado y luchado en momentos muy cruciales consiguiendo grandes objetivos. Me encantaría volver a morir trabajando con vosotros.

El genio se hace con un 1% de talento, y un 99% de trabajo.

Albert Einstein

Resumen

El auge de las comunicaciones inalámbricas ha sido significativo en las últimas décadas propiciando el desarrollo de diferentes estándares y su uso en múltiples aplicaciones, pero por otro lado las comunicaciones cableadas siguen siendo ampliamente utilizadas en determinados sectores. Actualmente, la seguridad y fiabilidad en cualquier sistema de comunicaciones se ha convertido en una prioridad de máximo nivel en todos los ámbitos: consumo, doméstico, industrial, defensa, etc.

En esta línea, la presente tesis aborda aspectos de seguridad y fiabilidad en las comunicaciones tanto inalámbricas como cableadas. Para ello, se han elegido tres aplicaciones que permitan estudiar algunas problemáticas en este tipo de comunicaciones y aportar soluciones a las mismas.

En una primera línea de trabajo se ha estudiado el estándar IEEE 802.15.4 y se ha propuesto un nuevo mecanismo para resolver el problema relacionado con el elevado tiempo de asociación de nodos móviles, donde sus continuas asociaciones afectan al tiempo de actividad. El procedimiento propuesto permite la transmisión de datos desde un nodo móvil y la recepción de datos desde nodos estáticos con un tiempo de actividad mínimo y una alta fiabilidad

de comunicación, reduciendo además el consumo de potencia y la pérdida de datos.

Continuando con esta línea, se ha desplegado una red de sensores inalámbricos para detectar fallos en motores facilitando su mantenimiento preventivo. Por un lado, la fiabilidad en la detección se asegura mediante la combinación de varios parámetros (vibraciones, corrientes y temperatura); y por otro lado, el uso del modo balizado con intervalo de tiempo garantizado (GTS) asegura la correcta transmisión de los datos. Teniendo en cuenta que en este tipo de aplicaciones inalámbricas es obligatorio reducir el consumo de potencia, se ha desarrollado una estrategia hardware y software de forma que los nodos alcancen un alto grado de autonomía.

En una segunda línea de trabajo se ha considerado una comunicación cableada basada en el protocolo I2C. En este caso, se ha estudiado la vulnerabilidad en la comunicación asociada a la utilización de alta impedancia como uno de los niveles lógicos. Este hecho, que garantiza una mayor escalabilidad en el bus, puede introducir fallos en las comunicaciones, ya sean tanto intencionados como no intencionados. En primer lugar, se ha estudiado el efecto de estos fallos en la trayectoria seguida por un robot móvil, mostrando que dicha trayectoria puede ser alterada. En segundo lugar, se ha desarrollado una estrategia de defensa para identificar este tipo de situaciones y actuar en consecuencia. La estrategia se basa en el uso de un sensor de frecuencia que detecta anomalías en la línea de temporización del bus.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1 Motivación

El desarrollo de la electrónica y las tecnologías de la información ha propiciado la llegada de la Sociedad Digital. Hoy en día casi todas las actividades en las que el hombre se involucra (económicas, productivas o sociales) están soportadas, de alguna manera, por el uso del computador y el intercambio de información. En este contexto, la seguridad y la fiabilidad de los sistemas electrónicos de comunicación y de control son de especial interés para las empresas en particular y la sociedad en general.

Sirva como ejemplo el artículo publicado el 4 se septiembre del 2017, por el periódico El País: "*Una ola de ciberataques bloquea LexNext, el sistema de notificaciones judiciales*". En él se relata cómo un ciberataque dejó sin comunicación la intranet de las sedes judiciales, no permitiendo a letrados consultar y enviar documentación. Con el auge de los dispositivos IoT (*Internet of Things*) se expone de forma pública nuestros datos, poniendo en entredicho toda la capacidad de seguridad y fiabilidad de las comunicaciones como se ha demostrado anteriormente.

Cuando se habla de seguridad en comunicaciones existen muchas variables de estudio, pudiendo ser consideradas dos vertientes principales. Una de ellas hace referencia a la seguridad en sí misma (sería el caso de las técnicas de encriptación donde se pretende que nadie más que el receptor adecuado descodifique correctamente la información). La segunda vertiente hace referencia a la fiabilidad de las comunicaciones, es decir, la probabilidad de que el mensaje llegue de forma correcta y con éxito a su destino.

Según cita [Ros05], en los años 70's, IBM desarrolló un sistema de autentificación de usuarios ATM (*Automated Teller Machine Security*). Se trataba de un proceso encriptado de verificación de usuarios mediante PIN (*Personal Identification Numbers*) que sólo permitía el acceso a usuarios convenientemente autentificados. Durante la década de los 90's, los procesos de encriptación fueron incrementando su utilidad como protección SSL (*Secure Socket Layer*), siendo usados para el acceso a servidores, defendiendo dichos sistemas mediante el empleo de algoritmos propietarios. Se trataba de sistemas de bajo coste que actualmente se implementan en dispositivos *radio-transceiver* para comunicaciones inalámbricas.

La otra vertiente de estudio en temas de seguridad puede implementarse con la inserción de fallos como se comenta en [Hagai06]. En dicho trabajo se proponen una serie de técnicas comunes para insertar fallos en un sistema:

- Variaciones en la fuente de alimentación. Los fallos recogidos con estas variaciones son mal interpretados o producen saltos de comandos.

- Variaciones en un reloj externo. Producen la pérdida de instrucciones, y por lo tanto, desajustes en la comunicación.

- Temperatura. Cuando la temperatura cambia del rango marcado por el fabricante suelen ocurrir dos efectos: uno es la modificación de datos en RAM, y otro que los datos escritos en la memoria ROM no son correctos.

- Luz blanca. Los circuitos electrónicos son fotosensibles. La corriente inducida por los fotones puede alterar el correcto funcionamiento del circuito.

- Fallos producidos por variación de partículas debido a un láser. Los efectos a los que se exponen los circuitos son similares a los producidos por el efecto anterior.

- Haces de iones y Rayos X, es otra fuente de error en ensayos nucleares. Estos defectos se producen sin la necesidad de desacoplar el micro del circuito donde se encuentra instalado.

Estas inserciones de fallos pueden provocar dos efectos distintos: un efecto que sea provisional y otro que sea destructivo. En el primer caso el circuito recupera sus características originales [Hagai06]; sin embargo, el segundo es un efecto permanente, y por lo tanto, irreversible. De esta forma, la inserción de fallos, en cualquiera de sus modalidades, supone un ataque particular a la fiabilidad del sistema de comunicación.

La fiabilidad es un concepto muy utilizado en comunicaciones inalámbricas [IEE06]. Al igual que sucede en sistemas cableados, se suelen utilizar diversos mecanismos para que la comunicación entre nodos sea lo más fiable posible.

Desde el punto de vista del modelo de comunicación OSI, la fiabilidad y la calidad son variables que debe asegurar la capa de transporte. En los diversos sistemas de comunicaciones, tanto cableados como inalámbricos, se usan sistemas muy similares para asegurar el éxito al llevar el dato de un punto a otro.

El mecanismo más utilizado es el acuse de recibo (*Ack*), el cual dota de gran fiabilidad a los sistemas cableados e inalámbricos. En el caso de los buses I2C o SPI se utiliza el *Ack* para que el máster del bus de datos sea consciente de que los datos han llegado con éxito; en el caso de comunicaciones inalámbricas, además de usar el *Ack*, la capa física usa un sistema de modulación para encriptar y acceder a la red de forma segura [IEE06].

No obstante, en muchos casos el uso del *Ack* no es suficiente para garantizar la fiabilidad, debido a que pueden existir varios dispositivos que traten de transmitir al mismo tiempo, pudiéndose ocasionar colisiones en el medio de transmisión. En estos casos, los mecanismos usados son: CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), Aloha, Asentamiento de tramas y verificación de datos [IEE06].

Las diferentes normas, cableadas e inalámbricas, definen el modo de acceso al canal. En este caso, los sistemas más utilizados son CSMA/CA o CSMA/CD (*CSMA with Collision Detection*). Se trata de mecanismos definidos en la capa de enlace que detectan o evitan colisiones en el proceso de comunicación, y aseguran a cualquier estación base el acceso al canal por el que quieran comunicar.

El interés por la seguridad y la fiabilidad en los procesos de comunicación en aplicaciones de automatización o robóticas ha motivado la investigación desarrollada.

Particularmente, esta tesis presta especial atención al análisis de distintas causas que ponen en peligro la fiabilidad en sistemas de comunicaciones, tanto inalámbricos como cableados.

1.2 Objetivos

Al tratar dos conceptos diferenciados en su definición, como pueden ser seguridad y fiabilidad en las comunicaciones, los objetivos perseguidos en la tesis abarcan gran número de aplicaciones: desde dotar a un estándar de comunicaciones de un algoritmo capaz de gestionar nodos móviles con total fiabilidad, hasta resolver de forma satisfactoria el problema debido a la inserción de fallos con el fin de provocar un mal funcionamiento en el sistema completo.

En primer lugar se ha realizado un estudio del estándar de comunicaciones inalámbricas IEEE 802.15.4. El número de aplicaciones para este tipo de redes es tan elevado, que debido al avance en las investigaciones, el estándar deja algunas líneas abiertas. Tras realizar un estudio inicial, se han detectado algunas carencias, donde la fiabilidad del mismo se ve comprometida. Utilizando las herramientas que aporta el estándar para establecer todas las comunicaciones, se le ha dotado de un mecanismo capaz de gestionar nodos móviles, y realizar de forma fiable este tipo de comunicaciones en IEEE 802.15.4.

El conocimiento adquirido de este estándar en esta primera fase de la tesis ha permitido abordar su implementación en una aplicación específica. Para ello, se ha desplegado una red de sensores con topología en estrella para transmitir información de variables en motores de AC que permitan monitorizar y detectar en tiempo real fallos en su funcionamiento. Para establecer una comunicación fiable y segura se ha optado por un tipo de comunicación balizada con GTS (*Guaranteed Time Slot*). Este hecho limita el número de nodos dentro de la red, pero aumenta la fiabilidad en zonas de mucho ruido electromagnético. Además, se ha diseñado una estrategia hardware de muy bajo consumo para dotar a los nodos de una elevada autonomía como es obligatoria en redes WSN (*Wireless Sensor Network*).

Tras realizar el estudio de un estándar de comunicaciones inalámbricas en base a la fiabilidad, en la segunda parte de la tesis se opta por estudiar una comunicación cableada. En el estudio y desarrollo de algunas aplicaciones se han encontrado algunas deficiencias debido al hardware utilizado en el protocolo de comunicaciones. Dichas deficiencias representan una vulnerabilidad del sistema, pues permiten alterar el comportamiento correcto del mismo si, de forma intencionada, se provoca un fallo del hardware involucrado. Se trata de lo que ha venido a denominarse un ataque hardware.

Para realizar este estudio, se ha diseñado una plataforma de experimentación abierta con las siguientes capacidades:

- Monitorización de cualquier tipo de variables, como puede ser el estado en el que se encuentra el sistema en tiempo real.

- Alteración de las comunicaciones a través de inserción de fallos. Un ejemplo de esta alteración puede ser la modificación de la trayectoria seguida por un robot móvil.

Para conocer el alcance de los fallos insertados en el protocolo de comunicaciones, el sistema anterior se ha implementado en una FPGA, que a su vez se ha conectado a una plataforma robótica. A partir de esta arquitectura, se han estudiado cada una de las vulnerabilidades propuestas, implementando ataques hardware al sistema completo mediante la inserción de fallos.

El estudio se ha centrado en un protocolo de comunicaciones cableado muy utilizado en entornos robóticos, como es el protocolo I2C [Gom01],[Ham13]. El mecanismo de inserción de fallos se ha basado en un método reversible, fundamentado en la modificación de la frecuencia de la señal de temporización (generalmente denominada *scl*) del protocolo [Hagai06]. El efecto de la inserción de los fallos se ha caracterizado gracias a la utilización de un esclavo estándar, conectado externamente al dispositivo FPGA. La verificación a bajo nivel del proceso está garantizada debido a una de las capacidades mencionadas anteriormente.

Adicionalmente, en el dispositivo FPGA se ha diseñado una estrategia de defensa a los mencionados ataques. La principal estrategia de defensa para estos ataques es la utilización de sensores de frecuencia para determinar si la frecuencia está en el rango permitido de funcionamiento. Se han realizado varios experimentos relacionados con la navegación de robots móviles en circunstancias particulares, demostrando que los ataques al reloj del bus I2C aumentan de forma considerable la vulnerabilidad del sistema. Los experimentos validan la eficiencia del sensor ante estos ataques.

Los objetivos perseguidos en cada uno de los capítulos son diversos:

- Diseño e implementación de un nuevo método para la gestión de nodos móviles en redes WSN de baja latencia y baja tasa de datos. Se ha buscado un sistema de comunicaciones que carecía de este tipo de redes, y mediante un sistema de gestión de nodos, se ha dotado al estándar de esta vertiente de forma fiable y segura.

- Diseño e implementación de un sistema de instrumentación capaz de detectar fallos en motores AC. Basado en el protocolo de comunicaciones IEEE 802.15.4, se ha buscado una forma fiable y segura de transmitir datos sensibles a una estación base, para que los operarios puedan tomar decisiones sobre las posibilidades de mantenimientos en las maquinarias.

- Diseño e implementación de una plataforma capaz de monitorizar cada una de las variables que se puedan producir en un ataque hardware. Dicha plataforma ha sido implementada en un dispositivo FPGA.

- Implementación de un sensor capaz de detectar ataques "*Clock Glitching*"; se ha utilizado un sensor digital que detecta irregularidades en las acciones del reloj. Para mantener la uniformidad de las implementaciones se ha elegido como plataforma un dispositivo FPGA, aunque la implementación está preparada para su uso en otras plataformas.

1.3 Estructura de la tesis

Los contenidos están organizados de la siguiente forma:

En el Capítulo 2 se ha realizado un pequeño estudio de sistemas de comunicaciones de muy bajo consumo y bajo coste seleccionando el más adecuado para redes de sensores inalámbricos (WSN) basadas en el estándar IEEE 802.15.4. Se ha presentado un mecanismo en el estándar de comunicaciones seleccionado, aportando una comunicación bidireccional entre nodos móviles y una red estática, operando en el modo no balizado de la norma. El mecanismo usado utiliza las transmisiones *broadcast* para emitir información desde el nodo móvil; esto no requiere una asociación por cada transmisión, ya que el nodo se encuentra asociado continuamente a la red. Para validar el correcto funcionamiento del mecanismo se ha realizado una serie de experimentos, donde los resultados alcanzados han sido satisfactorios. Para ello, se ha implementado un sistema hardware, de bajo coste y muy bajo consumo.

En el Capítulo 3 se diseña un sistema inalámbrico para la gestión del mantenimiento predictivo en motores AC, buscando el estudio de la fiabilidad en la norma IEEE 802.15.4 basado en una comunicación con balizas y GTS. Para ello se han realizado dos tipos de experimentos. En primer lugar, los nodos inalámbricos se han ubicado entre motores para analizar las pérdidas de información entre dos puntos, y poder estudiar las influencias de los campos magnéticos generados en los motores. En segundo lugar, se han realizado experimentos, tanto en laboratorio para probar el correcto funcionamiento de los sensores, como en campo para probar la fiabilidad del sistema de comunicaciones. Además, se han realizado unas estrategias de gestión de

energía mediante la sinergia hardware y firmware, permitiendo alcanzar al sistema una elevada autonomía.

En el Capítulo 4 se ha diseñado una plataforma de experimentación para la monitorización de las diferentes variables de un sistema incluido en la misma. Dicha plataforma ha sido particularizada para el estudio de la comunicación a través del protocolo I2C. Con el fin de aumentar la capacidad de configuración de la misma, se ha optado por una implementación en un dispositivo FPGA. El diseño de la plataforma ha incluido los siguientes elementos: una estrategia de multiplexación de señales para la monitorización de las variables; una estructura del protocolo I2C incluyendo un elemento maestro y un elemento esclavo (sólo utilizado para tareas de simulación); una estrategia para incluir fallos en la señal de temporización del protocolo de comunicaciones; y un mecanismo de comunicación con Matlab basado en un transceiver RS-232 para el envío de comandos. Se han realizado una serie de experimentos para comprobar la fragilidad del bus de comunicaciones I2C, realizando una serie de ataques con la intención de alterar la trayectoria de la plataforma robótica.

Tomando como base la plataforma del capítulo anterior, en el Capítulo 5 se ha propuesto una estrategia de defensa a los ataques por inserción de fallos en la señal de temporización del protocolo de comunicaciones. Se propone el diseño de un nuevo sensor, que representa una efectiva defensa frente a las diversas perturbaciones aplicadas. Las estrategias de defensa y ataques han sido validadas en una plataforma experimental, que emula el funcionamiento de un robot diferencial. Se han realizado varios experimentos relacionados con la navegación de robots móviles (para lo cual se ha incluido una comunicación inalámbrica basada en el protocolo Bluetooth). En circunstancias particulares se ha caracterizado que los ataques al reloj del bus I2C aumentan de forma

considerable la vulnerabilidad del sistema. Los experimentos demuestran la eficiencia del sensor en la defensa del sistema.

Finalmente, en el Capítulo 6 se presentan las conclusiones y las líneas futuras de investigación.

Finalmente, se incluyen dos Anexos que aportan información genérica sobre diferentes estándares de comunicaciones inalámbricas así como de comunicaciones cableadas.

Capítulo 2. Mecanismos de baja actividad para redes de sensores basadas en IEEE 802.15.4

2.1 Introducción

En esta tesis doctoral la tecnología elegida para la realización de una red WSN es IEEE 802.15.4 por las características que presenta: baja tasa de datos para redes inalámbricas de área personal (LR-WPAN, *Low Rate Wireless Personal Area Network*); permite flexibilidad en la red; permite crear diferentes topologías; se puede realizar una monitorización con un gran número de nodos en la misma red para abarcar una mayor cobertura, y sobre todo, el bajo consumo. Además, al poseer una pila de protocolos muy reducida los dispositivos necesitan muy poca memoria.

2.1.1 Fiabilidad

El estándar utiliza una serie de mecanismos para que la comunicación entre nodos sea lo más fiable posible, los cuales son: CSMA/CA, Aloha, asentamiento de tramas y verificación de datos.

- *Mecanismos CSMA/CA*.

La norma define la comunicación, de forma que el acceso al canal se puede realizar de dos formas que son dependientes de la red. En redes sin balizamientos se usa CSMA/CA *no slotted* como mecanismo de acceso al canal; esto significa que cuando un dispositivo desea transmitir una trama, éste espera un tiempo aleatorio. En el caso de que el canal se encuentre ocupado, si tras este tiempo el canal vuelve a estar ocupado, el nodo puede esperar otro tiempo. En el caso contrario, el nodo transmite la información al canal.

Por otro lado, cuando el acceso al canal es balizado se usa CSMA/CA *slotted*, donde los periodos de tiempos son llamados *backoff slots*; los slots se alinean con el comienzo de la transmisión de la baliza. Los nodos deben sincronizar los periodos con el coordinador que transmite las balizas. Cada vez que un dispositivo quiera transmitir tramas de datos, lo debe de hacer en el periodo de la supertrama que le corresponda. El periodo de trama utilizado para transmitir datos es llamado CAP (*Contention Access Period*); éste localiza el límite del siguiente *backoff* y espera un número aleatorio de *backoffs*; si en estos momentos el canal se encuentra ocupado, el nodo puede volver a esperar otro número aleatorio de *backoffs* antes de volver a intentar acceder al canal. Si por el contrario se encuentra vacío, el nodo puede comenzar la transmisión en el límite del siguiente *backoff* disponible. Las tramas de asentamiento y de balizamiento son transmitidas sin usar CSMA/CA.

- *Aloha*.

El mecanismo de acceso Aloha tiene su origen en 1970 de la mano de Norman Abrason, en la universidad de Hawai. Presenta un nuevo y elegante mecanismo para la solución del problema de la asignación de canal en un medio compartido. Su aplicación principal se centra en los sistemas de radio. Aunque se basa en algoritmos muy simples, su sencillez ha dado lugar a un uso

muy extenso en protocolos actuales y ha generado una familia de protocolos derivados tales como CSMA, PRMA o DORUMA que mejoran el rendimiento.

Aloha permite el acceso para transmisión en cualquier instante de tiempo, cuando el emisor dispone de datos. Este mecanismo supone un riesgo de dos o más transmisiones simultáneas o coincidencia de señales en el medio, ya sea aéreo o cableado. Este estado se denomina colisión e implica la destrucción de las formas de ondas, y por tanto, de la información que las señales transportan.

- *Asentamiento de tramas*.

En un modo opcional, el nodo puede transmitir una validación de datos tras la recepción de una serie de tramas mediante unos comandos MAC. Este mensaje se lanza cuando el nodo receptor desencripta la información; en caso contrario, no lanzaría dicho mensaje.

Si el nodo que ha transmitido el mensaje no recibe el acuse de recibo durante un periodo de tiempo determinado, éste asume que la transmisión es incorrecta, y reintentará la transmisión de la trama. Si tras varios intentos no se recibiese el asentamiento, el nodo puede elegir entre terminar la transmisión o esperar un intervalo de nuevo.

En caso de que no se haya configurado al nodo para esta forma de comunicación, el nodo cree que la transmisión siempre ha sido correcta.

- *Verificación de datos*.

Para captar errores a nivel de bit se emplea el mecanismo CRC (*Cyclic Redundancy Check*) de 16 bits que es usado para detectar errores en cada trama. Este código se almacena en el campo FCS de cada trama.

2.1.2 Seguridad

En las redes inalámbricas no existen niveles físicos capaces de evitar escuchas o modificaciones debido a que se utiliza el espectro electromagnético, por lo que un ataque con la cobertura adecuada, puede interceptar la información sin que sea detectado.

Para evitar dichos ataques, el estándar establece una serie de medidas que permiten autentificar los nodos que forman parte de la red inhabilitando la asociación de aquellos nodos que no estén autorizados [Car10]. La norma establece un algoritmo de cifrado que se debe utilizar en las operaciones de encriptación, pero no especifica cómo han de gestionarse las claves o las políticas de autenticación.

La seguridad se obtiene mediante un cifrado simétrico, el algoritmo de cifrado es AES (*Avanced Ecryption Standard*) con una longitud de claves de 128 bits. El algoritmo además de cifrar la información, la valida mediante un código de integridad de mensaje (MIC, *Message Integrity Code*) añadido al final del mensaje. Este código asegura la integridad de la cabecera MAC y del *payload*, a la vez que asegura al emisor lo que debe decir. El cifrado de ciertas partes de la cabecera MAC se construye con la clave establecida en la política de gestión de claves, y debe ser conocida por los nodos que se estén comunicando; en caso de recibir alguna trama que no se corresponda con el código MIC generado, se niega la comunicación.

El MIC se compone de varios tamaños, 32, 64 y 128 bits, aunque el algoritmo que se utiliza es AES 128 bits. Este tamaño indica sólo cuantos bits se añadirán al final de cada trama. La confidencialidad de las comunicaciones se consigue cifrando el contenido del *payload* mediante el algoritmo AES 128 bits.

2.1.3 Formatos de la trama

Para la gestión de operaciones de seguridad es necesario una serie de campos IEEE 802.15.4 como se muestra en Figura 2.1:

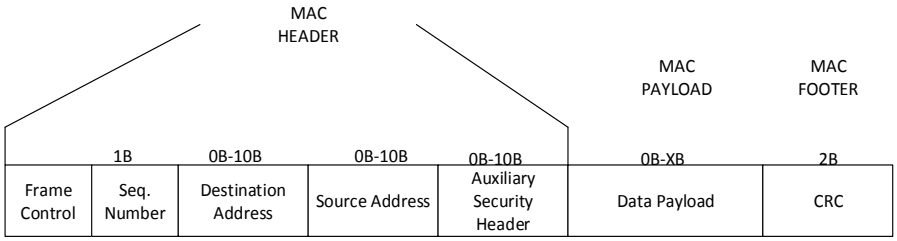


Figura 2.1 Cabecera IEEE 802.15.4.

Auxiliary Security Header, sólo se activa si el bit *Security Enabled*, del campo *Frame Control* es habilitado a nivel alto.

Security Control, indica el nivel de seguridad seleccionado para esta trama.

Frame Counter, es un contador proporcionado por el emisor de la trama para proteger ante ataques de repetición. Por esta razón, cada mensaje tiene un número de secuencia único representado por este campo, no necesariamente correlativo.

Key Identifier, especifica la información necesaria para seleccionar la clave en el nodo receptor. Los nodos han de contener las mismas claves organizadas de la misma manera.

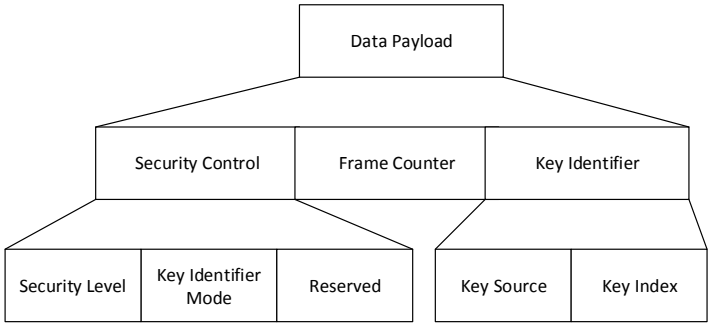


Figura 2.2 Sub-campos ASH.

El sub-campo *Security Control* es el lugar donde se ubica la política de seguridad que seleccionará el modo de funcionamiento AES y el modo de identificación de la clave, que puede ser implícito o explícito. El resto del espacio está reservado para posibles ampliaciones. Los valores posibles de *Key Identifier Mode* son:

- **0**, el valor de la clave es conocido de manera implícita por el emisor y por el receptor, por lo que no se especifica en este el mensaje.

- **1**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y el parámetro estático *macDefautKeyStore*.

- **2**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y los 4 bytes de *Key Source*.

- **3**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y los 8 bytes de *Key Source*.

Según esta configuración, el número máximo de claves que pueden utilizarse es de 272, lo que implicaría un consumo máximo de memoria para las claves de 272 x 16B [Car10], lo cual no es factible en ningún sistema. Lo importante de este aspecto no es el número en sí, sino que el número es suficientemente grande para que sea escalable a diferentes políticas de gestión de claves.

La implementación que se hace de la seguridad indica que ésta se efectúa a razón de cada trama. Esto quiere decir que un receptor para cada trama recibida tendrá que seleccionar la clave correspondiente, actualizar los valores de contador y realizar la operación criptográfica correspondiente. Teóricamente, un mismo nodo emisor podrá enviar dos tramas diferentes a un

mismo receptor con niveles de seguridad distintos, lo que aporta una gran flexibilidad. Por ejemplo, se podrán enviar tramas *beacon* garantizando la identidad del coordinador pero sin cifrar el contenido, y utilizar un cifrado completo en caso de enviar tramas de datos.

Los niveles de seguridad que ofrece IEEE 802.15.4 se especifican en el sub-campo *Security Level* de la cabecera auxiliar de seguridad. Estos niveles definen el modo de funcionamiento del algoritmo AES proporcionando autenticación, confidencialidad o ambas. Los 3 bits de este campo permiten seleccionar entre 7 niveles de seguridad, desde lo más bajo, que no realiza ninguna operación criptográfica, hasta el nivel que ofrece más garantías. La siguiente tabla especifica las características de cada nivel.

Valor	Cifrado	Operación
0	Sin seguridad	Datos en claro, Autentificación sin validar
1	AES CBC MAC 32	Datos en claro, Autentificación validada
2	AES CBC MAC 64	Datos en claro, Autentificación validada
3	AES CBC MAC 128	Datos en claro, Autentificación validada
4	AES CTR	Datos cifrado, Autentificación sin validar
5	AES CCM 32	Datos cifrado, Autentificación validada
6	AES CCM 64	Datos cifrado, Autentificación validada
7	AES CCM 128	Datos cifrado, Autentificación validada

Tabla 2.1 Niveles de seguridad.

Esencialmente, lo que indica la Tabla 2.1 es que existen 3 modos diferenciados (CBC-MAC, CTR y CCM) de realizar operaciones criptográficas y que aportarán funcionalidades diferentes.

2.1.4 Canales de transmisión

En IEEE 802.15.4 se definen 27 canales de frecuencias entre las tres bandas (ver Figura 2.3 y Tabla 2.2). La capa física de los 868/915 MHz soporta un solo canal entre los 868 y los 868.8 MHz, y diez canales entre los 902.0 y 928.0 MHz. Debido al soporte regional de esas dos bandas de frecuencias, es muy improbable que una sola red utilice los 11 canales. Sin embargo, las dos bandas se consideran lo suficientemente cercanas en frecuencias, de forma que se puede utilizar en el mismo hardware y así reducir costes. La capa física de los 2.4 GHz soporta 16 canales entre los 2.4 y los 2.4835 GHz, con un amplio ancho de banda entre canales de 5 MHz, con el objetivo de facilitar los requerimientos de filtrado en la transmisión y en la recepción de información.

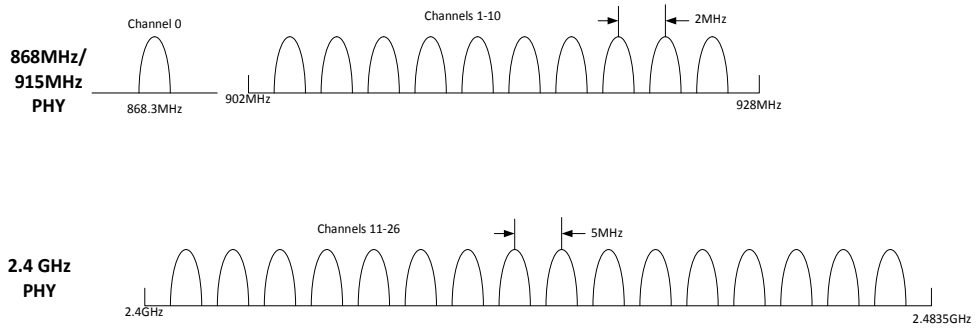


Figura 2.3 Estructura de los canales.

Número de canales	Frecuencia central del canal MHz
1	868.3
K=1,2,3,….,10	906+2(k-1)
K=11,12,…….,26	2405+5(k-11)

Tabla 2.2 Canales del estándar IEEE 802.15.4.

El estándar fue diseñado para implementar una selección dinámica de canales a través de una selección específica de algoritmos, la cual es responsable de la capa de red. La capa MAC incluye funciones de búsqueda que sigue paso a paso a través de una lista de canales permitidos en busca de una señal de sincronización; mientras que la PHY contiene varias funciones de bajo nivel,

tales como la detección de los niveles de energía recibidos, indicadores de calidad en el enlace, así como de conmutación de canales, lo que permite asignación de canales y agilidad en la selección de frecuencias. Esas funciones son utilizadas por la red para establecer su canal inicial de operación, y para cambiar canales, existe una pausa muy prolongada.

2.1.5 Modelos de transferencia de datos

El estándar propone tres tipos de transferencia de información: la primera es la transmisión de un nodo a un coordinador de la red; la segunda es la transferencia de datos entre dos nodos comunes; y la tercera opción es el envío de información entre dos pares de dispositivos. En una topología estrella sólo se utiliza la primera de estas opciones, ya que los datos sólo pueden ser transferidos entre el nodo coordinador y un dispositivo, y viceversa. En una red peer-to-peer, la comunicación entre nodos es constante en la red, por lo que en este tipo de topologías se darán las tres formas de propagación de la señal, que se describen en los siguientes puntos.

Los mecanismos para cada tipo de transferencia de datos permite la sincronización mediante balizas. Una red sincronizada por *beacons* permite utilizar dispositivos en baja latencia, sin embargo, la sincronización sigue siendo necesaria para la asociación de la red [IEE06].

2.1.5.1 Transferencia a un coordinador

Esta operación de transferencia de datos es el mecanismo que se utiliza para comunicarse entre un nodo cualquiera de la red y un coordinador PAN cuando se encuentran dentro del mismo radio de cobertura. Cuando el dispositivo desea comunicarse con el coordinador, en primer lugar debe permanecer a la escucha en la red de las señales de sincronización; una vez el

dispositivo ha detectado la sincronización, observa si el canal se encuentra ocupado: en caso de ser afirmativo espera una señal de sincronización más, y en caso de ser negativo, lanza los datos para que el nodo coordinador los recoja. Aunque es opcional, el nodo coordinador puede lanzar un *ack* para confirmar la llegada de los paquetes. En la Figura 2.4(a) se muestra el flujo de datos según [IEE06].

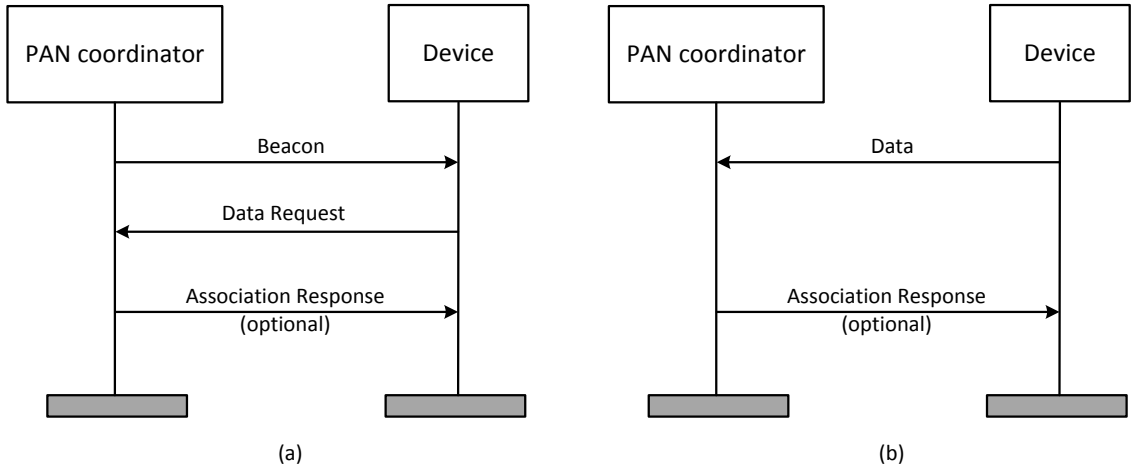


Figura 2.4 Comunicación entre un nodo y un coordinador. a) Beacon enabled mode. b) Non neacon enable mode.

Cuando la red trabaja sin balizas de sincronización, es decir, *non-beacon enable*, simplemente transfiere la trama de datos al medio utilizando mecanismos de acceso al medio CSMA/CA. El nodo coordinador de la red recoge la información y lanza una trama *ack*, y de esta forma, confirma la llegada de los paquetes, como se puede observar en Figura 2.4(b).

2.1.5.2 Transferencia desde un coordinador

Esta operación de transferencia de datos es el mecanismo que utiliza el coordinador de la red para evitar enviar datos a un nodo asociado a él. Cuando el coordinador desea transferir datos a un nodo, lanza una baliza de mensaje. El nodo que se encuentra escuchando la red, recibe la baliza que indica que está pendiente de datos. En ese momento transmite un mensaje de la MAC que

solicita los datos utilizando CSMA/CA. El coordinador reconoce la recepción mediante un *ack* opcional. El coordinador lanza al medio un *ack+data* para que el nodo dé por buena la comunicación; para finalizar el protocolo, el nodo vuelve a comunicar un *ack* que es recogido por el coordinador.

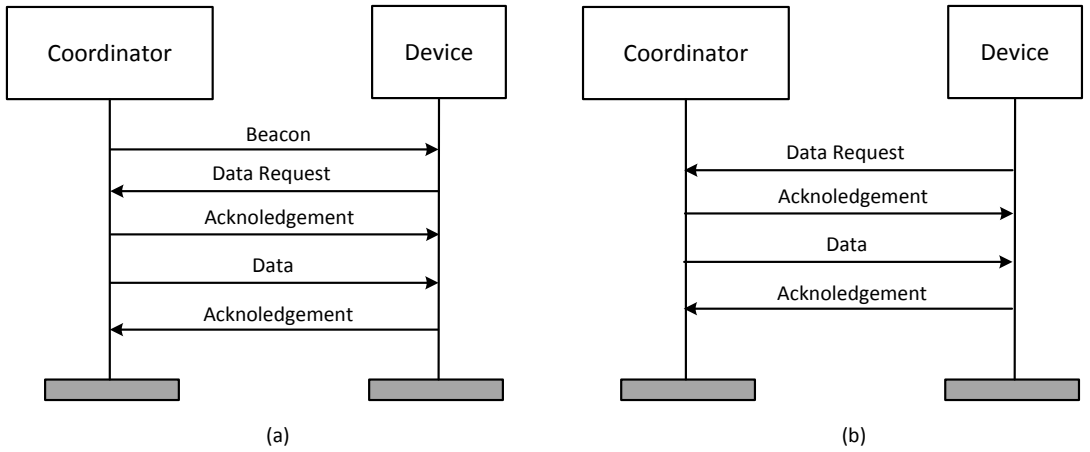


Figura 2.5 Comunicación desde un coordinador hasta un nodo. a) Beacon enabled mode. b) Non neacon enable mode.

Cuando la aplicación requiere que las balizas de sincronización no se encuentren habilitadas, el coordinador que desea transmitir datos a un dispositivo de la red, almacena los datos para el dispositivo, y de esta forma establecer el contacto entre ambos. Cuando el nodo se pone en contacto con el coordinador, lo realiza mediante los mecanismos CSMA/CA utilizando slots de tiempo que lanza respuestas al canal. Si el coordinador desea comunicarse con él, se lanza un *ack+data* y espera la recepción del *ack* para dar por terminada la comunicación. En la Figura 2.5(a) se observa una comunicación con balizas habilitadas, y en la Figura 2.5(b) sin habilitar las balizas.

2.1.5.3 Comunicación P2P

En una red *peer to peer* todos los dispositivos se pueden comunicar con cualquier dispositivo siempre que se encuentre dentro de su radio de cobertura. Con la finalidad de hacerlo de manera efectiva, los dispositivos que deseen

comunicarse deberán de estar a la escucha de forma constante, o sincronizar la red mediante balizas de sincronización. En el primer caso, utiliza mecanismos de acceso al medio como CSMA/CA; en el último caso se utilizan otras medidas para lograr la sincronización; estas medidas de *mesh* se encuentran fuera de la norma y depende del diseñador de la red.

2.1.6 Procesos de asociación

Los procesos de asociación de los nodos a la red pasan por realizar una series mecanismos de detección del nodo vecino (*Scan Active* and *Scan Pasive*), y por las negociaciones con el nodo padre. Estos mecanismos se realizan tanto en redes balizadas como en redes no balizadas.

Las estrategias de detección del nodo cercano es un tiempo necesario para que el nodo que desea asociarse recoja la información y decida a qué nodo asociarse, basándose en una serie de criterios programados en alto nivel.

ScanDuration es un parámetro de tiempo programable del estándar que establece el tiempo de espera a la respuesta del nodo padre al que se quiere asociar un hijo. La cantidad de canales a escanear también es otro parámetro de importancia, ya que puede elevar demasiado el tiempo de asociación y el nodo durante este tiempo sólo atiende a la respuesta, quedando anulada cualquier funcionalidad operativa.

Según fija el estándar [IEE06]:

$$ScanDuration = aBaseSuperframeDuration * (2^n + 1) \qquad (2.1)$$

Donde n es un parámetro programable de 0-14.

$$aBaseSuperframeDuration = 15.36ms$$

Donde como mínimo se necesita un tiempo de 30.72 ms por canal.

Una vez seleccionado el nodo al que se quiere asociar, se establece unas negociaciones descritas en el estándar tal y como se muestra en Figura 2.6.

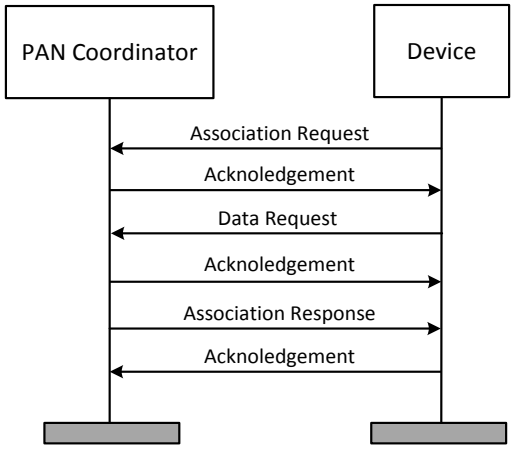


Figura 2.6 Negociaciones de asociación.

El tiempo de respuesta en las negociaciones viene dado por *macResponseWaitTime*, parámetro definido por el estándar, y que es programable.

2.2 Estrategias de control para nodos móviles

Como ya se ha comentado previamente, el estándar 802.15.4 se ha convertido en los últimos años en un referente para aplicaciones inalámbricas de baja tasa de datos y baja eficiencia energética. La norma especifica el diseño de redes LR-WPAN, ya que ofrece una baja complejidad, bajo consumo de potencia y un bajo coste económico. Todas las características nombradas hacen de esta tecnología una opción para un amplio rango de aplicaciones [Yic05],[Wen06]. Sin embargo, en los sistemas basados en IEEE 802.15.4 es

bastante preocupante la gestión de nodos móviles, ya que el estándar no especifica el diseño para nodos móviles y tiene una serie de inconvenientes a la hora de utilizarse en aplicaciones que sean sensibles al tiempo. Esto se debe a que tiempo de asociación en los nodos se considera lento [Zhe06], resultando un inconveniente en aplicaciones con nodos móviles. Además del tiempo de asociación, otro factor que influye es el consumo de potencia durante los procesos de asociación, lo que limita la vida de las baterías.

2.3 Nodos móviles en IEEE 802.15.4

Un hecho importante cuando se trabaja con el estándar IEEE 802.15.4 es el modo de funcionamiento de la red, sobre todo cuando se diseñan sistemas que integran nodos móviles. Como se ha podido ver en los apartados anteriores, este estándar especifica dos modos de transferencia de información entre nodos dependiendo de si la red está configurada en *beacon enable mode* o *non-beacon enable mode*.

En *beacon enable mode*, los nodos utilizan una estructura para la transferencia de datos entre un nodo (nodo padre) y los nodos asociados a él (nodos hijo) denominada supertrama, como se muestra en Figura 2.7. Esta estructura permite a los nodos competir para el acceso al medio en el espacio temporal CAP (*Contention Active Period*) de acceso libre. Para aplicaciones de baja latencia un dispositivo puede usar porciones definidas previamente de la supertrama activa *(GTS, Guaranteed Time Slot*) para la transmisión de datos al nodo asociado. El acceso de los nodos al medio viene determinado por el mecanismo *CSMA/CA Slotted* sincronizado por medio de las balizas de dicha estructura.

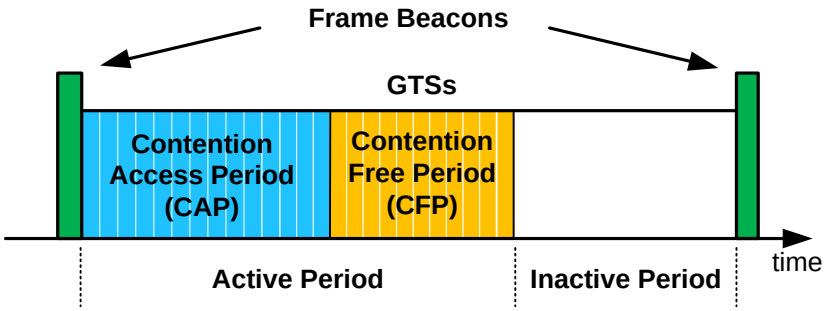


Figura 2.7. Estructura de la supertrama en *beacon enabled mode*.

Así pues, el modo de funcionamiento balizado se basa en el seguimiento constante de balizas por parte de los nodos asociados al emisor de dichas balizas, o lo que es lo mismo, el nodo siempre tiene que estar asociado a algún nodo de la red. De esta forma, cuando un nodo se desplaza y pierde esa asociación, éste debe permanecer en estado de actividad para encontrar nuevas balizas (nuevo nodo padre) que le permitan asociarse de nuevo a la red sincronizándose a la supertrama de otro nodo. Por tanto, una red funcionando en *beacon enable mode* implica un tiempo de actividad en el seguimiento de balizas, aun cuando no tenga información que transmitir o recibir, además del protocolo de asociación posterior. Este tiempo es inevitable con esta configuración de la red, ya que es inherente al modo de funcionamiento balizado, llegando a ser determinante en aplicaciones críticas en consumo o tiempo.

Aunque se han propuestos mecanismos para mejorar la eficiencia [Zha08], el cambio constante de nodo hace que el sistema sea inviable desde el punto de vista del consumo y del retraso si el sistema integra nodos con movilidad media o alta [Zen08]. Otro aspecto a tener en cuenta cuando se diseña una red con balizamiento es la complejidad añadida de utilizar varios niveles jerárquicos. Dado que los nodos móviles se van a apoyar en una estructura estática, la necesidad de separar las zonas activas de las distintas supertramas de cada

nodo, sugiere un mecanismo de gestión de alto nivel que permita la utilización de distintas ventanas de tiempo que evite una posible colisión entre distintas balizas o transmisiones [Kou07], lo que dificulta una posible implementación real.

Respecto al modo de funcionamiento *non-beacon enable mode.* En este tipo de red no existe estructura para la transferencia de datos, ni existe un método de sincronización entre nodos mediante balizas, sino que simplemente se utiliza un mecanismo *unslotted CSMA/CA* cuando un nodo desea acceder al medio de transmisión un mensaje. En este tipo de redes no balizadas, aunque no se obliga al seguimiento constante de balizas, cuando un nodo desea transmitir o recibir un mensaje, este debe encontrarse asociado a la red. Por tanto, un nodo que desea interactuar con otros nodos y que ha perdido la asociación debe asociarse a la red mediante el proceso que define el estándar. Este modo de funcionamiento es más simple que el anterior, aun cuando tiene el inconveniente de que un nodo no sabe cuándo ha perdido la asociación hasta que decide transmitir un mensaje a la red.

En lo sucesivo nos centraremos en este modo de funcionamiento, ya que da lugar a implementaciones mucho más sencillas cuando se tienen redes con varios niveles jerárquicos. Además, como veremos, es posible evitar el problema de las asociaciones en *non-beacon enable mode* logrando una mejora apreciable tanto en el tiempo de actividad como en el tiempo de comunicación en la información móvil.

2.4 Mecanismos de baja actividad propuestos

En este apartado se propone un nuevo mecanismo que elimina la necesidad de asociaciones continuas en redes *non beacon* para nodos móviles, y

que por tanto disminuye tanto el retraso de la información como el tiempo de actividad necesario en estos nodos. Este procedimiento incluye además características de fiabilidad en la transferencia de información entre un nodo móvil y una estructura estática.

El mecanismo propuesto es válido tanto para la transferencia de información desde un nodo móvil como para la transmisión de información hacia un nodo móvil, lo que proporciona una elevada versatilidad en el diseño de aplicaciones.

La diferencia entre nodos estáticos y nodos móviles radica en sus direcciones de red, por lo que para la utilización de esta forma de gestión es necesario un mecanismo de asignación de direcciones que diferencie ambos tipos de nodos. De esta forma, ambos tipos de nodos se detectarán de forma simple.

El procedimiento desarrollado está diseñado para que los nodos móviles puedan moverse libremente a través de la cobertura proporcionada por una estructura estática formada por dispositivos FFD (*Full Function Device*). Dicha estructura será la encargada, en última estancia, de la transferencia de información al nodo centralizador de la información (coordinador PAN), y los nodos móviles sólo intercambian información con dicha estructura.

Los nodos móviles en la aplicación son también dispositivos FFD pero con las asociaciones deshabilitadas (ningún otro nodo puede asociarse a un nodo móvil). La forma en que estos nodos interactúan con la estructura estática es mediante mensajes tipo *broadcast*, tanto para la transmisión de información como para la recepción de la misma.

La utilización de dispositivos FFD en lugar de dispositivos RFD (*Reduced-Function Device*) permite que un nodo móvil pueda transmitir y recibir información de forma directa con cualquier otro nodo de la red, evitando la necesidad de transferencias indirectas, que obligan al nodo a estar asociado a un nodo concreto. Esta forma de funcionamiento tiene la ventaja de no necesitar una asociación para la transferencia de información, aunque tiene un coste asociado, ya que un dispositivo FFD necesita de más recursos hardware para su funcionamiento, aunque los beneficios son grandes en eficiencia.

Como ya se ha comentado, los nodos móviles interactúan con la red estática mediante mensajes tipo *broadcast*. Esta característica evita la necesidad de continuas asociaciones en la red. Estos mensajes en el estándar IEEE 802.15.4 tienen la característica de que cualquier nodo FFD puede recibir este mensaje siempre que se encuentre en el radio de propagación del nodo transmisor, independientemente de su dirección de red origen o destino. Por tanto, la utilización de este tipo de transmisiones tiene la ventaja de no necesitar conocer la dirección de destino.

En [Wan07] se presenta un mecanismo similar para solventar el problema del elevado tiempo de las asociaciones, proponiendo trasmisiones *broadcast* para la transferencia de información hacia la estructura estática. Mientras que, al ser el nodo móvil un dispositivo RFD, se mantienen las transferencias indirectas para la comunicación hacia los nodos móviles. Este procedimiento tiene la desventaja de que, dado que los mensajes *broadcast* no pueden ser asentidos según el estándar, no existe robustez en la transferencia de información hacia la estructura estática. Una pérdida de mensaje no podría ser detectada mediante este método. Además existe el problema de que para la transferencia indirecta es necesario que el dispositivo móvil se encuentre asociado a la red. Con lo que

no se pueden evitar las asociaciones en la recepción de información por parte del dispositivo RFD (nodo móvil).

Para solventar estos inconvenientes se ha dotado al mecanismo de fiabilidad adicional, utilizando también transmisiones *broadcast* para el intercambio de información entre la estructura estática y los nodos móviles, sirviendo estos de asentimiento además de para transmitir información. A continuación se explica con mayor detalle cómo se realiza la gestión de ambas comunicaciones por parte del procedimiento diseñado.

2.4.1 Gestión de la comunicación desde un nodo móvil hacia la estructura estática

En primer lugar, un nodo móvil que desee transmitir algún tipo de información (monitorización periódica o mensaje de evento) transmite un mensaje de datos en modo *broadcast* hacia todos los nodos de la estructura estática que se encuentren dentro de su espacio de propagación como se muestra en la Figura 2.8.

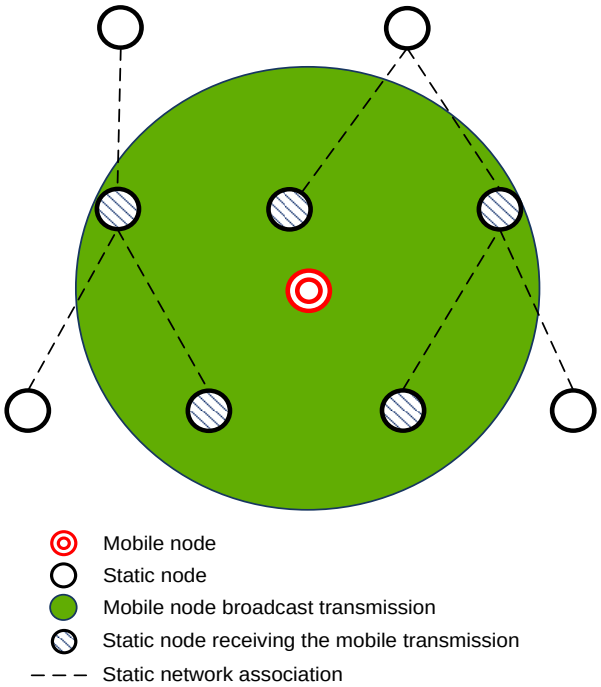


Figura 2.8 Transmisión de datos *broadcast* desde un nodo móvil.

Una vez recibido este mensaje por cualquier nodo de la estructura estática, será reconocido como un mensaje procedente de un nodo móvil mediante el análisis de la dirección del nodo origen del mensaje.

Todos los nodos estáticos que reciban este mensaje responderán también mediante la transmisión de un mensaje en modo *broadcast*, que al ser recibido por el nodo móvil dará por válido la transmisión de los datos, como se muestra en la Figura 2.9. Este mensaje de respuesta no tiene por qué llevar datos de información (*dummy data*) sirviendo sólo de asentimiento al mensaje móvil.

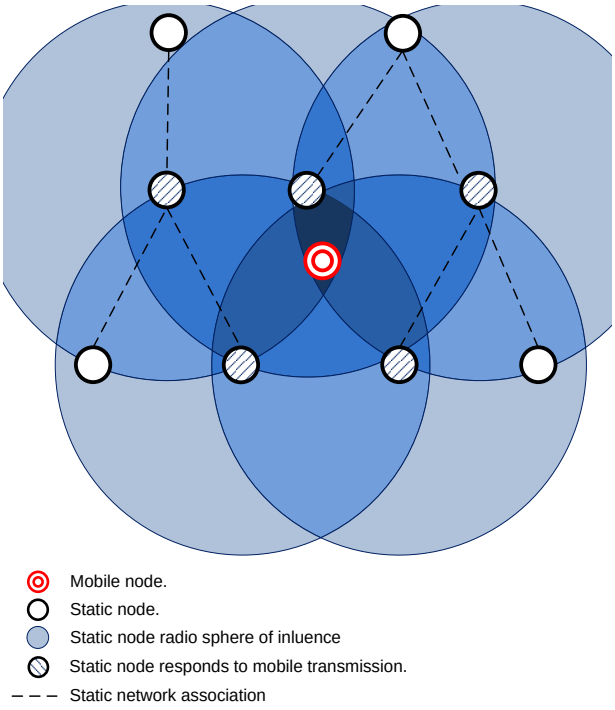


Figura 2.9 Respuesta *broadcast* desde los nodos estáticos.

Aunque puede responder más de un nodo estático a la información transmitida por el nodo móvil, para reducir el tiempo de actividad en dicho nodo, éste sólo espera el primer mensaje que asiente su transmisión, pasando a modo de bajo consumo a partir de ese momento. Como se puede observar, no existen asociaciones a la red para la transmisión de información, utilizando solamente una asociación inicial y siendo el nodo móvil el que decide cuándo transmitir información, ya sea periódica o información de eventos.

2.4.2 Gestión de la comunicación desde la estructura estática hacia el nodo móvil

Cuando la red estática tiene información que transmitir a un nodo móvil (como por ejemplo datos de configuración o actuación), esta información espera en los nodos estáticos hasta que el nodo móvil pregunta por ella, utilizando una transmisión de datos (*broadcast transmission*), como se observa en la Figura 2.8. Esta transmisión, en el caso concreto de que el nodo móvil solamente quiera actualizar su información, almacena un dato reservado para la petición de información. Mientras que si además el nodo móvil tiene información útil que transmitir, el mensaje almacenara dicha información.

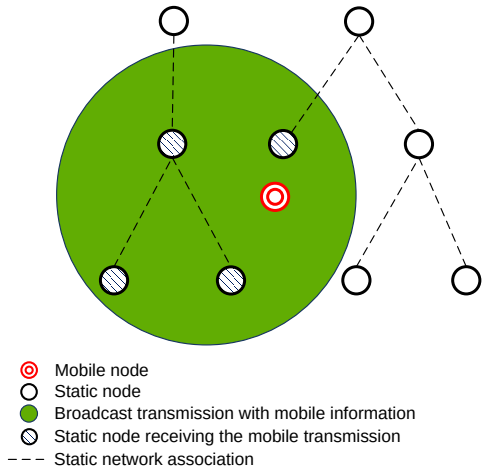
Para la transmisión de la información desde la estructura estática a los nodos móviles, tras la petición de dicha información, se utiliza el mismo mecanismo mostrado que en la Figura 2.9. La información con destino a los nodos móviles se transmite almacenada en los mensajes *broadcast* que sirven de asentimiento a los nodos móviles, de manera que cuando el nodo móvil lo requiere, o bien transmite su información, actualiza su propia configuración. Al igual que en el caso anterior, este mecanismo de gestión evita la necesidad de continuas re-asociaciones que provocarían una gestión poco eficaz desde un punto de vista temporal y energético.

2.4.3 Gestión de la información móvil por parte de la estructura estática

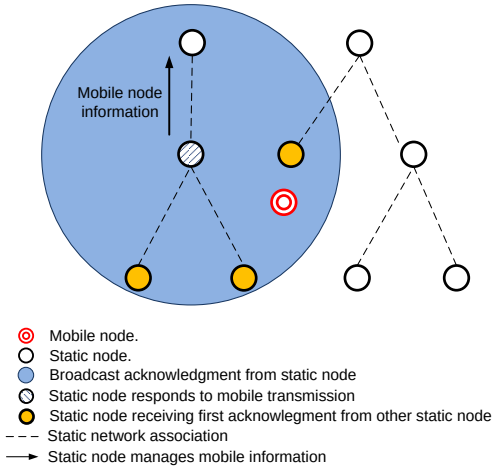
El procedimiento diseñado para el tratamiento de nodos móviles presupone que una vez que el nodo móvil transmite su información, ésta es gestionada por una estructura estática para su transmisión hacia el nodo coordinador PAN.

Dado que la información de un nodo móvil puede llegar a varios nodos estáticos, se puede provocar un aumento considerable del tráfico en la red. Para

minimizar este hecho se puede utilizar, además de alguna técnica clásica como agregado de datos [Hei99],[Mar13], los mensajes de asentimiento de la estructura estática (Figura 2.10) de la siguiente forma: Cuando se asiente un mensaje de información móvil mediante transmisiones *broadcast*, es probable que dicho mensaje también sea recibido por otros nodos de la estructura estática, de manera que, un nodo estático detecta que dicho número de mensaje ya ha sido asentido y por tanto no necesita asentirlo de nuevo dejando la gestión del mensaje hacia el coordinador PAN al primer nodo que respondió a la transmisión. Un ejemplo de este comportamiento se observa en la Figura 2.10.



(a)



(b)

Figura 2.10 Ejemplo de minimización del tráfico en la estructura estática. a) Transmisión *broadcast* desde el nodo móvil. b) Sólo un nodo maneja el mensaje de la información móvil.

2.5 Fiabilidad del mecanismo propuesto

Para el intercambio de información con origen en el nodo móvil, el procedimiento indica que dicha interacción de un nodo móvil con la red estática se realice mediante mensajes tipo *broadcast*. Dado que los mensajes *broadcast* no admiten petición de asentimiento en el estándar IEEE 802.15.4, cuando un nodo estático recibe un mensaje de datos procedente de un nodo móvil, realiza una transmisión de datos también en modo *broadcast*, que sirve como asentimiento al nodo móvil.

Por tanto, dado que un nodo móvil realiza una transmisión de datos a todos los nodos estáticos que se encuentran dentro de su rango de cobertura, la fiabilidad es bastante elevada, ya que, incluso en condiciones de escalabilidad media/baja en la red, varios nodos recibirán el mensaje de datos siendo difícil la pérdida de dicha información.

Un nodo móvil siempre espera al menos un mensaje que sirve de asentimiento para dar validez a su transmisión. En caso de pérdida de la información móvil o su asentimiento, ésta es detectada por la no recepción del asentimiento transcurrido un tiempo. Tras expirar este tiempo se realizaría una retransmisión de la información desde el propio nodo móvil, quedando de nuevo a la espera de la recepción de al menos un mensaje de asentimiento.

Para el caso de la transferencia de información entre la estructura estática y el nodo móvil, y dado que esta se encuentra almacenada en el mensaje de asentimiento, tendremos una situación parecida. En la peor de las situaciones, es decir, un solo nodo asintiendo para minimización de tráfico, la pérdida de dicho mensaje originaría una nueva transmisión por parte del nodo móvil, y por tanto, una nueva transmisión de la información desde la estructura estática.

De esta forma se garantiza la integridad de la comunicación en ambos sentidos frente a posibles interferencias o colisiones en el medio de transmisión.

2.5.1 Implementación hardware

Los sistemas hardware para la evaluación de dicho procedimiento se han diseñado utilizando la solución *on chip* MC13213 de Freescale Semiconductor. Esta plataforma trabaja en la banda libre de frecuencias de 2.4 GHz (banda ISM) disponible de forma global en el mundo, y sus características más relevantes vienen resumidas en la Tabla 2.3.

Frequency Band	2.4 GHz ISM 16 selectable channels
Data Rate	250 kbps O-PQSK modulation
Sensitivity (1% PER)	-92 dBm
Output Power	0 dBm nominal (programmable from -27 to +3 dBm)
Internal Memory	60 KBytes Flash, 4 K bytes RAM
Core	40 MHz Freescale HCS08 low-voltage, low-power
Several low power modes	Doze, Hibernate and Off

Tabla 2.3 Solución Freescale MC13213. Características para aplicaciones IEEE 802.15.4.

Equipado con un transceptor completamente compatible con el estándar IEEE 802.15.4, soporta 250 kbps, con datos O-QPSK en canales de 5.0 MHz y codificación y decodificación de espectro extendido completo. Además, el fabricante proporciona la pila de protocolo IEEE 802.15.4 Standard-Compliant MAC (2006) para el diseño de aplicaciones sobre dicho estándar. La siguiente figura muestra los nodos diseñados para las pruebas del sistema basados en la plataforma descrita.

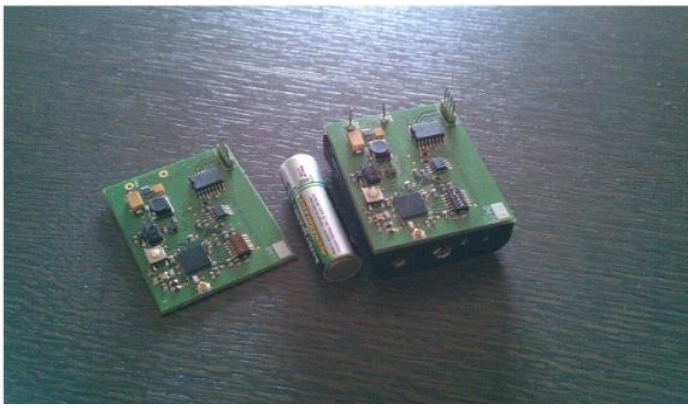


Figura 2.11 Nodo estático y nodo móvil diseñados para el test experimental.

La diferencia de los módulos diseñados para las pruebas radica en el sistema de alimentación, donde los módulos estáticos disponen de un convertidor AC/DC para su conexión a la red eléctrica, además de la posibilidad de utilizar baterías. Obviamente, los nodos móviles sólo disponen de la posibilidad de utilizar baterías. Se colocó un conector *mmcx* en la placa para permitir el uso de múltiples antenas.

2.5.2 Test del sistema

Para las pruebas del sistema se han instalado 25 nodos estáticos siguiendo un grid de 30m x 30m cubriendo un espacio total de 14.400 m^2 tal y como se indica la Figura 2.12. A través de esta estructura se han dispuesto 5 nodos móviles que pueden moverse libremente por la zona de cobertura. La localización de la red 802.15.4 ha sido en un entorno *outdoor* sin obstáculos y se han utilizado potencias de transmisión nominales de 0 dBm con antenas omnidireccionales para las pruebas.

La red estática ha sido inicializada siguiendo criterios de robustez en los enlaces, es decir, para la decisión sobre los enlaces se han tenido en cuenta solamente criterios de maximización de potencia recibida. Las transmisiones de los nodos móviles son recogidas por los nodos estáticos y retransmitidas hacia

el nodo coordinador PAN (nodo centralizador de la información) siguiendo los enlaces creados en la inicialización.

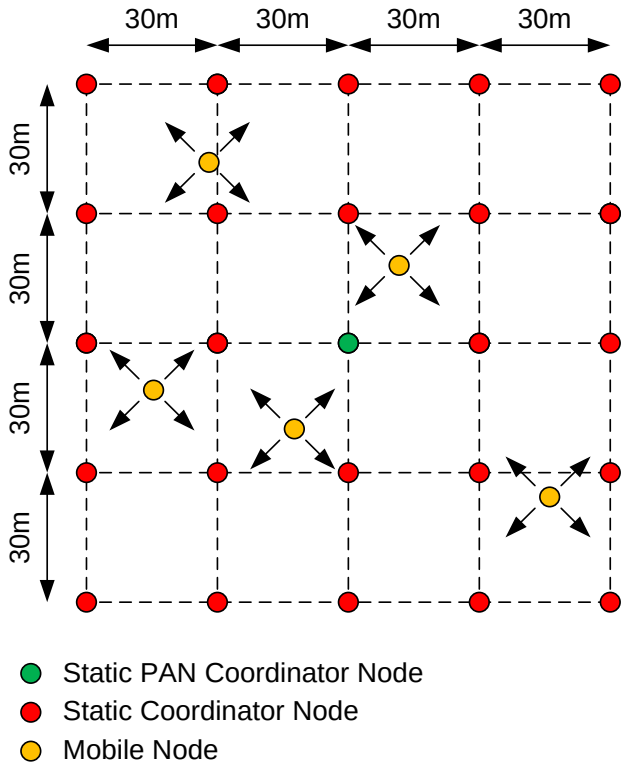


Figura 2.12 Diagrama de localización de los nodos estáticos y móviles en el entorno de test.

Tras la verificación del correcto funcionamiento de la red, se han realizado pruebas para medir tanto el tiempo necesario para la comunicación de información desde un nodo móvil a la estructura estática (*delay time of mobile monitoring data*) como el tiempo de actividad necesario en un nodo móvil para la transmisión y recepción de información con la estructura estática (*activity time for sensing and actuation task*). Mediante estas mediciones se pretende demostrar las prestaciones del mecanismo de gestión de nodos móviles propuesto.

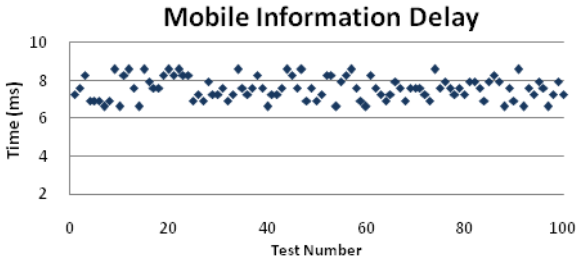
2.5.3 Medidas de retardo de la monitorización de datos en el nodo móvil

Para la medición de este tiempo se ha programado un nodo móvil para que transmita un mensaje cada vez que ocurre una interrupción externa forzada

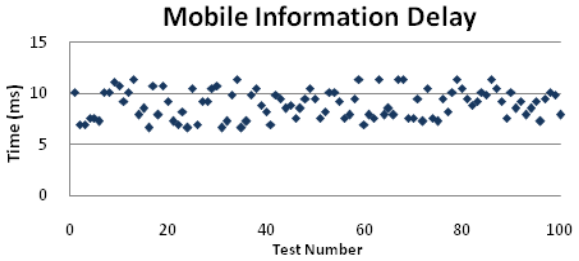
por el usuario. A partir de ese momento el nodo móvil genera un mensaje con dos bytes de información (*monitoring message*) que transmite siguiendo el mecanismo implementado. Cuando este mensaje llega a un nodo de la estructura estática, genera una interrupción en el nodo receptor que tras analizar que procede de un nodo móvil (*short address* correspondiente a un nodo móvil), genera un pulso en uno de los pads del microcontrolador (MCU).

En el experimento se han generado 100 interrupciones y se ha medido el tiempo que transcurre desde la generación de la interrupción en el nodo móvil hasta el pulso generado en uno de los nodos estáticos receptores. En la Figura 2.13 se muestran los resultados de estas mediciones para distintos valores de BE (*backoffexponent*). Como se puede observar, todas las medidas están muy por debajo del tiempo de *scan* (0,26 seg. por canal) y del protocolo de asociación (0,49 seg. aprox.), existiendo una gran variabilidad en la medida, procedente principalmente de la configuración del mecanismo CSMA/CA de acceso al medio. El tiempo medido sólo incluye el tiempo debido a las comunicaciones. En el diseño de aplicaciones reales habrá que sumar el tiempo necesario para el sensado, el cual dependerá del sensor que se utilice.

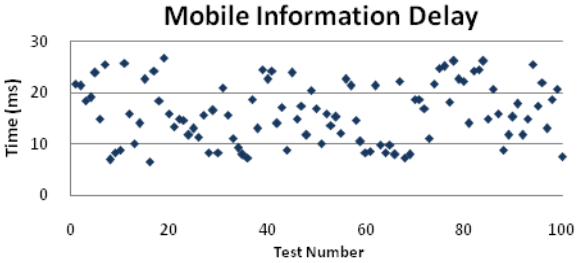
Obviamente, la latencia total del mensaje dependerá de la latencia de la estructura estática, ya que ésta es la encargada de tramitar la información con origen en los nodos móviles. Por tanto, es recomendable que dicha estructura disponga de algoritmos de encaminamiento que minimicen el número de saltos, así como un mecanismo de sincronización que evite colisiones en situaciones de tráfico elevado.



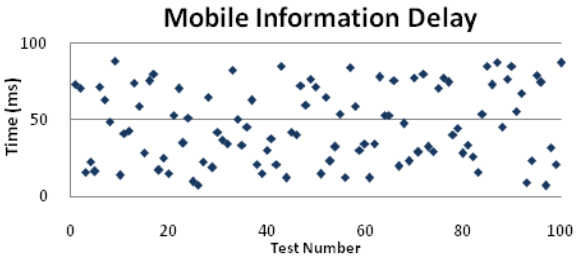
(a)



(b)



(c)



(d)

Figura 2.13 Tiempos de retardo en el nodo móvil para los diferentes parámetros de BE, a) BE=2, b) BE=4, c) BE=6, d) BE=8.

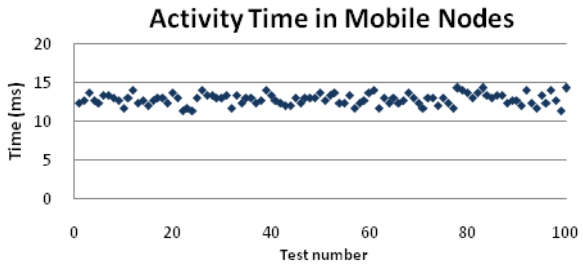
2.5.4 Medidas del tiempo de actividad en el nodo móvil para tareas de monitorización y actuación

Esta prueba tiene como objeto medir el tiempo de actividad necesario en un nodo móvil para transmitir su información a la estructura estática (*mobile sensing data*), así como recibir datos de ésta (*mobile actuation data*).

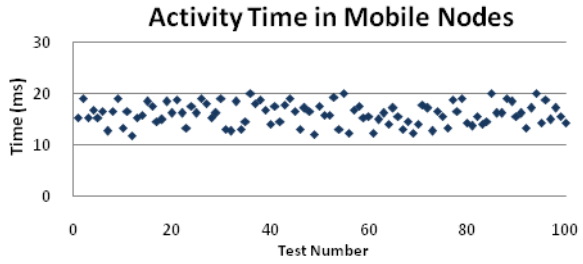
Para la medición de este tiempo se ha programado un nodo móvil para que transmita un mensaje de datos con dos bytes de información cuando se genera un evento externo (generado por el usuario), y reciba también un mensaje de datos con dos bytes de información desde la estructura estática mediante el mensaje que sirve de asentimiento, tal y como se define en el mecanismo. Cuando este mensaje de asentimiento con datos para el nodo móvil es recibido, genera una interrupción en el nodo móvil que genera un cambio de nivel en uno de los pads del MCU.

El tiempo de actividad ha sido calculando midiendo entre el espacio de tiempo entre la generación del evento (interrupción externa) y la generación del pulso tras la recepción del mensaje. Este experimento ha sido repetido en 100 ocasiones variando el parámetro BE (*Backoffexponent*) de acceso al medio. Los resultados de la prueba se muestran en la Figura 2.14.

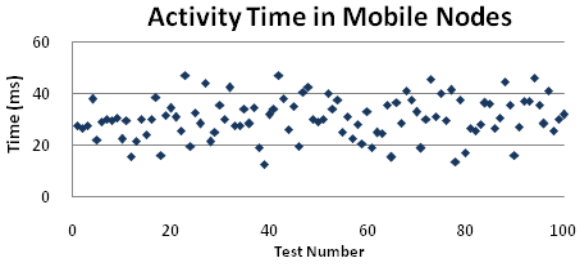
Como se ha comentado anteriormente, con la finalidad de disminuir el tiempo de actividad, el nodo móvil sólo espera para generar la interrupción al primero de los mensajes de asentimiento (e información), pasando a estado de inactividad inmediatamente después de recibir este mensaje. El tiempo obtenido en las pruebas, como se puede comprobar, es en cualquier caso muy inferior a los tiempos estimados necesarios para realizar una asociación, por lo que la mejora del método propuesto respecto al uso de asociaciones tradicionales es importante.



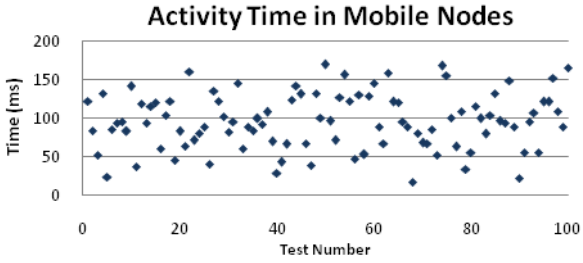
(a)



(b)



(c)



(d)

Figura 2.14 Tiempo de actividad en los nodos móviles para diferentes parámetros de BE, a) BE=2, b) BE=4, c) BE=6, d) BE=8.

Este tiempo medido incorpora los retrasos debido a la gestión de la comunicación y el procesamiento del estándar asociado, pero no recoge los

tiempos necesarios para pasar de estado de inactividad a actividad (*start up time*) y viceversa (*sleep time*). Este tiempo es variable dependiendo tanto del modo de bajo consumo seleccionado, así como de la plataforma hardware que se utilizase. Modos de bajo consumo muy profundos implican la necesidad de un mayor tiempo para pasar a estados de actividad, por lo que es necesario encontrar una solución de compromiso entre el bajo consumo y el tiempo de actividad. En el caso de las pruebas realizadas se ha configurado la plataforma utilizada en un modo de bajo consumo no muy profundo (≈ 100 μA), pero con un rápido paso a estado de actividad (≈ 350 μs *to idle*). Como se puede apreciar este tiempo se puede considerar casi despreciable en relación con los tiempos debidos a la comunicación. Es por esta razón que el tiempo medido puede ser estimado como el tiempo de actividad necesario en un nodo móvil para completar la comunicación bidireccional con la red.

Al igual que en la prueba anterior no se incluye el tiempo necesario para el sensado o actuación, ya que éste dependería del sensor y/o actuador utilizado, y por tanto, de la aplicación diseñada. La variabilidad del tiempo de actividad calculado es debida sobre todo al método de acceso al medio (CSMA/CA) empleado en el estándar.

2.6 Conclusiones

Se ha diseñado un nuevo mecanismo basado en el estándar IEEE 802.15.4 que permite el intercambio bidireccional de información de forma robusta entre un nodo móvil y una estructura estática funcionando en *non-beacon enable mode*. Dicho método utiliza transmisiones tipo *broadcast* para la transmisión de información desde el nodo móvil, evitando la necesidad de continuas re-asociaciones que influyen negativamente en el retraso de la información y el tiempo de actividad de éstos. Además, utilizando transmisiones *broadcast* desde

el nodo estático que recibe el mensaje podemos establecer un mecanismo de fiabilidad basado en asentimientos a la vez que permite transferir información desde la estructura estática a los nodos móviles.

Este mecanismo ha sido implementado en una plataforma hardware real para verificar su validez, y se han realizado pruebas para medir el tiempo de actividad y el tiempo de transmisión de la información móvil medido en dicha plataforma hardware/software. Los tiempos obtenidos en las pruebas muestran las prestaciones del método propuesto, obteniendo una mejora ostensible respecto a los tiempos que se obtendrían mediante la utilización de un procedimiento clásico basado en continuas asociaciones de los nodos móviles.

Capítulo 3. Red WSN optimizada en consumo para la monitorización y detección de fallos en motores

3.1 Introducción

El proceso de monitorizar un parámetro de un equipo industrial con objeto de identificar que un cambio significativo en el mismo sea indicativo de un fallo en su funcionamiento, se denomina en inglés, *condition monitoring*. En la industria supone una tarea de alta prioridad debido a su amplio uso y constituye la base del mantenimiento predictivo que tanto auge ha tenido en los últimos años [LuB09],[Bog13],[Qia15],[Lee15]. Este método emplea datos en tiempo real, determina el buen funcionamiento del equipo, optimiza los recursos de mantenimiento, y mejora la fiabilidad del sistema y la vida útil de los equipos.

En los países desarrollados la mayor parte de la energía consumida por la industria se debe a los motores, siendo los motores de inducción trifásicos los dominantes debido a su robustez y facilidad de mantenimiento. Uno de los métodos usados con mayor éxito en la prevención y detección de fallos en motores es la medida de vibraciones y su posterior análisis. Otro método empleado para el diagnóstico de fallos en motores es la medida de la temperatura [Zah07]: una excesiva fricción o deterioro de los rodamientos

provocará un incremento en la temperatura nominal de funcionamiento del motor [Gou12]. Además, elevados picos de consumo debido a la pérdida de una fase en el motor también son parámetros a considerar [GuF15],[Yan09]. Problemas en los parámetros mencionados requieren actuaciones inmediatas para prevenir daños irreparables en el motor. La experiencia práctica sugiere que para predecir correctamente los fallos en los sistemas es importante reunir cuanta más información mejor. Por lo tanto, la combinación de la técnica de las vibraciones junto con otros parámetros eléctricos puede proporcionar un mayor éxito en la detección de fallos.

En los siguientes apartados se presenta el diseño e implementación de un sistema inalámbrico de sensores capaz de predecir algunos fallos en motores mediante las medidas en tiempo real de vibraciones, temperatura y consumo de corriente. El sistema tiene una alta eficiencia energética y despliega una red estática de sensores basada en el estándar de comunicaciones IEEE 802.15.4 usando el modo *beacon-enabled (modo balizado)* y empleando el mecanismo GTS (*Guaranteed Time Slot*) consiguiendo una latencia fija en la monitorización de datos y un periodo de muestreo ajustable. Aunque la industria siempre ha sido reticente a implementar sistemas inalámbricos por su menor fiabilidad en la transmisión de datos que los sistemas cableados, el uso del estándar IEEE 802.15.4 en modo *beacon-enabled* permite la sincronización entre los nodos sensores o nodos *end-devices* y el nodo coordinador y se asegura la correcta transmisión de datos entre ambos.

3.2 Descripción hardware del sistema

A continuación se describe la arquitectura hardware, los sensores usados y la funcionalidad del sistema. La Figura 3.1 muestra los tres subsistemas que se han diseñado: los dispositivos sensores, el nodo coordinador y el sistema de

gestión de la información en la estación base donde el operario realiza un seguimiento de la evolución de los motores; la red está desplegada en una topología estrella. El nodo coordinador está conectado vía USB a un ordenador donde los datos son evaluados y procesados. Este nodo recoge toda la información de la red de sensores y la transmite a la aplicación de monitorización, que fue desarrollada para controlar la red. También se evaluó la alternativa de procesar localmente en los nodos sensores los datos recogidos de los motores y sólo transmitir al nodo coordinador el resultado de la evaluación y procesado de los mismos (por ejemplo, la alarma de un fallo), reduciéndose así el tráfico de datos, pero a costa de incrementar los requerimientos hardware y software de los nodos de sensado.

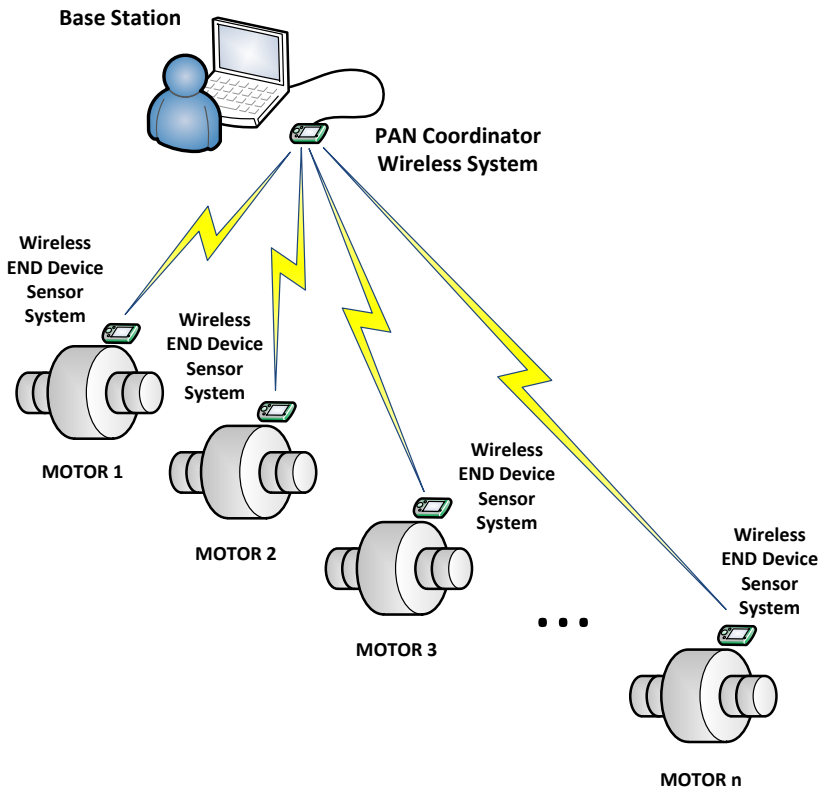


Figura 3.1 Arquitectura de la red inalámbrica de sensores diseñada.

El sistema inalámbrico ha sido implementado usando dispositivos que cumplen requisitos de bajo coste, reducido tamaño y una robusta tecnología de

radio. Además, se ha seleccionado electrónica de bajo consumo para los sensores, sus interfaces de acondicionamiento, el microcontrolador y el transceptor. La red puede también ser fácilmente escalable.

En la estación base se ha desarrollado un instrumento virtual basado en LabVIEW que monitoriza los datos recibidos por cada uno de los nodos *end devices*. Estos datos son tratados y representados a un nivel estándar marcado por [ISO10], donde la superación de unos valores límites impuestos por la norma en algunos casos, y en otros casos propuestos por los operarios, harían saltar una alarma de fallo o posible fallo en el sistema.

3.2.1 Nodos sensores

En la Figura 3.2 se muestra el diagrama de bloques de los nodos sensores. Cada nodo se estructura en una arquitectura de dos procesadores funcionando en paralelo para equilibrar costes, consumo de potencia y rendimiento, y consiste en dos circuitos independientes: uno basado en el microcontrolador ATmega328 y otro en el radio transceiver ATmega128RFA1; ambos son de 8 bits y del fabricante Atmel.

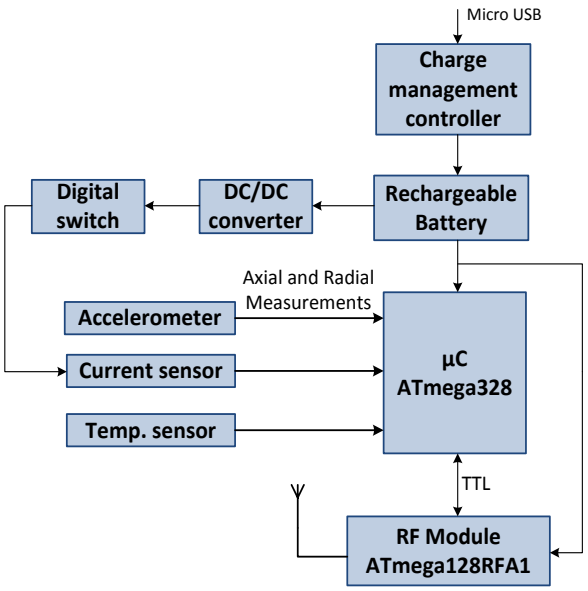


Figura 3.2 Diagrama de bloques de la estructura hardware del nodo sensor.

El microcontrolador recoge los datos de los sensores y el transceptor controla la comunicación inalámbrica IEEE 802.15.4. La comunicación entre el microcontrolador y el radio transceiver se realiza mediante comandos TTL. Además, el nodo incluye un conjunto de tres tipos de sensores con salidas analógicas que miden los parámetros requeridos, esto es, vibraciones, temperatura del motor y corrientes de las fases del motor. Si se empleara un único microcontrolador sería más difícil conseguir una monitorización en tiempo real de las vibraciones porque el número de muestras requeridas para una evaluación exacta estaría más limitado. El nodo posee una alimentación externa mediante una batería AAA de 3.3V; se trata de una pila recargable para lo cual se ha instalado el dispositivo MCP73831 de Microchip que regula la carga de esta batería mediante un conector micro USB. En la siguiente figura se observa la implementación del nodo sensor.

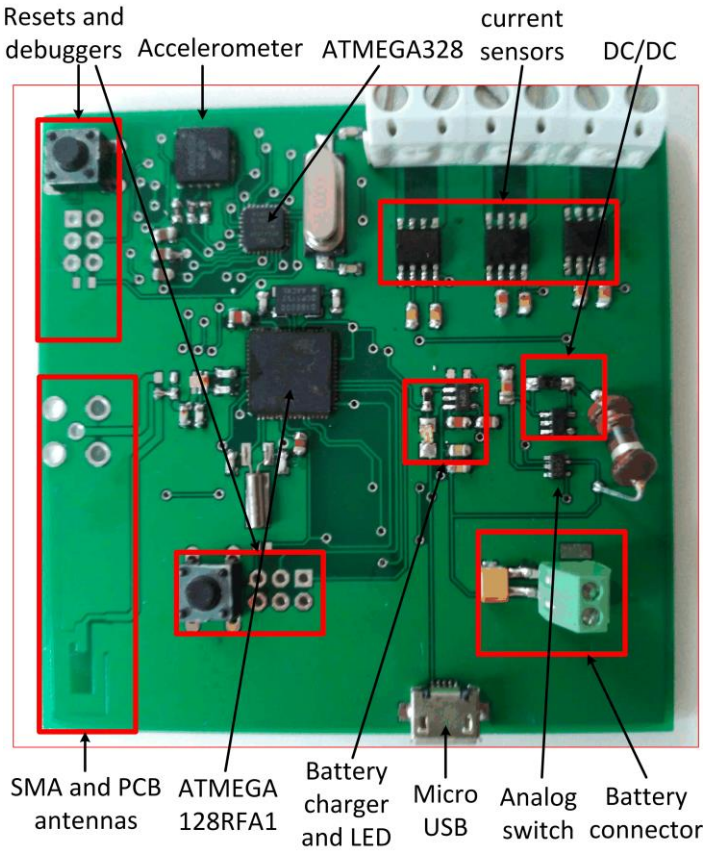


Figura 3.3 Imagen de la PCB del nodo sensor.

3.2.1.1 Modulo inalámbrico y microcontrolador

El módulo inalámbrico ATmega128RFA1 es un microcontrolador de 8 bits combinado con transceptor de alta tasa de datos para ZigBee e IEEE 802.15.4 para la banda ISM de 2.4GHz. El radio transceptor proporciona una comunicación muy robusta y emplea un número mínimo de componentes externos. Combina excelente funcionamiento RF con bajo coste, pequeño tamaño y bajo consumo de potencia. Para la optimización de energía el nodo sensor sólo está alimentado el tiempo específico para el proceso de captura de datos. Después de este proceso, el nodo permanece en el denominado modo *deep sleep* hasta la próxima recogida de datos. Las principales características que del módulo inalámbrico se recogen en la Tabla 3.1.

Frequency Band	2.4 GHz ISM 16 selectable channels
Data Rate	250 kbps -100dBm 2000 kps -86dBm
Sensitivity	-100 dBm
Memory	128 KBytes Flash, 16 KBytes RAM, 4Kbytes EEPROM
Microcontroller	8-bit
Very low power consumption	< 250nA Sleep Mode 12.5 mA RX mode 14.5 TX mode (0dBm)

Tabla 3.1 Características del módulo inalámbrico ATmega128RFA1 compatible con el estándar IEEE 802.15.4.

Como se ha comentado previamente, teniendo en cuenta que el ATmega128RFA1 es responsable de ejecutar la pila de protocolo IEEE 802.15.4, y tiene que cumplir con exigentes requisitos de tiempo en esta aplicación, se decidió emplear otro microcontrolador, el ATmega328, que sincronice el

proceso de adquisición de datos de los sensores y la comunicación con el transceptor. Las salidas analógicas de los sensores se convierten a digital por el convertidor A/D del microcontrolador, y son enviadas al transceptor de radio; éste a su vez intercambia los datos con el nodo coordinador vía inalámbrica a 2.4GHz. Las principales características del ATmega328 se muestran en la siguiente tabla.

Basic instructions	131 instructions
Minimum instruction execution time	50 ns(f(BLCK)=20 MHz, Vcc=3.0 to 5.5V)
Register	32x8 General Purpose Working Registers
Serial I/O	1 Channels (UART, clock synchronous serial I/O) 1 Channel (UART, clock synchronous, I²C, or IEBus)
Power supply voltage	Vcc =3.0 to 5.5 V (f(BCLK)=20 MHz) Vcc=2.7 to 5.5V (f(BCLK)=10 MHz)
Power consumption	0.2mA (Vcc 5V, f(BCLK)=20 MHz)

Tabla 3.2 Características principales del microcontrolador ATmega328.

3.2.1.2 Sensor de corriente

La monitorización de las corrientes del motor se ha realizado mediante el sensor ACS712 de Allegro MicroSystems LLC que se basa en el Efecto Hall. Se trata de una alternativa de bajo coste, precisa y con bajo offset que puede medir rangos de corrientes desde 0 a 5 A, con una sensibilidad de 185 mV/A, y alimentado con 5 V. Las características del sensor de corriente se muestran en la siguiente tabla.

Rango de medida	5 A
Tiempo de respuesta	5μs
Consumo	30 mA
Ancho de banda	80kHz
Sensibilidad	185mV/A
Linealidad	1.5%
Precisión -40°C-25°C	0.054mV/A/°C
Precisión 25°C-150°C	-0.008mV/A/°C

Tabla 3.3 Características del sensor de corriente ACS712 basado en el Efecto Hall.

De forma específica la estructura interna del sensor de Efecto Hall se muestra en la figura siguiente. El transductor de corriente mide de forma indirecta la corriente que circula a través del primario. La corriente en el secundario es lineal respecto de la corriente en el primario. Finalmente, un amplificador inversor con salida unipolar convierte la corriente sensada hacia una señal de tensión. El voltaje de salida es transformado a digital por el convertidor A/D del microcontrolador.

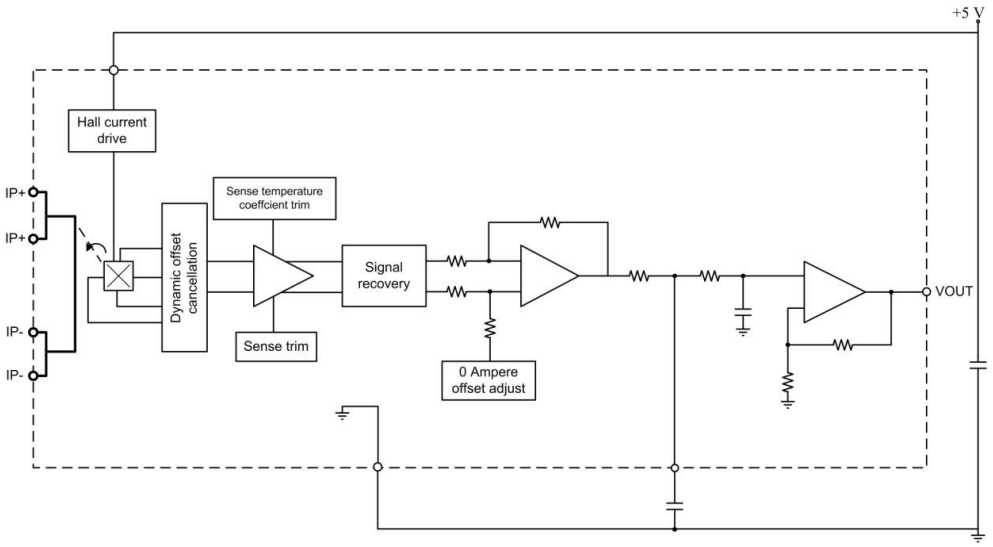


Figura 3.4 Diagrama de bloques del funcionamiento del sensor de corriente.

El sensor de corriente tiene una sensibilidad de 0.185 V/A y un offset de 2.5 V. La corriente de cada fase del motor se calcula según la siguiente ecuación:

$$I_n = \frac{analogRead - Offset}{sensitivity} \quad (3.1)$$

donde *analogRead* es la tensión de entrada en el puerto del microcontrolador.

Como el sensor de corriente no soporta el modo *sleep*, para alcanzar una estrategia de bajo consumo se ha utilizado un *switch* analógico que desconecta los sensores de corriente reduciendo considerablemente el consumo de la electrónica. Además, al tener el nodo sensor inalámbrico en su conjunto una alimentación exclusiva de baterías a 3.3 V, se ha tenido que incluir en el diseño el convertidor Boost elevador de tensión NCP1400 de Microchip de 3.3 V a 5 V para alimentar a los sensores de corriente.

3.2.1.3 Sensores de temperatura

Como sensor de temperatura se ha optado por utilizar un LM35 de Texas Instruments, que es un circuito integrado de precisión cuya salida es una tensión proporcional a la temperatura (en ◦C). Con el sensor LM35 la temperatura se puede medir con mayor exactitud que con un termistor. El sensor tiene una precisión de 0.1 °C, una capacidad de operar entre -55 °C a 150 °C y una sensibilidad de 0.01 V/°C.

Una vez que se envían los datos a la estación base, ésta realiza la siguiente conversión:

$$Temp = \frac{analogRead * 100 * 5.00}{1024} \quad (3.2)$$

La resolución del sensor se encuentra por encima de la resolución del convertidor analógico/digital del microcontrolador. El sensor posee una resolución de 10 mV/°C y el microcontrolador una resolución de 4.88 mV.

3.2.1.4 Acelerómetro

El MMA7260QT es un acelerómetro de tres ejes basado en un sistema micro-electro-mecánico (MEMS). Se trata de un sensor de bajo coste y alta sensibilidad basado en las medidas de variaciones de capacidad que puede llegar a medir aceleraciones de hasta ±6 g en diferentes rangos. El dispositivo requiere de un convertidor analógico para obtener las medidas de cada eje. Este acelerómetro posee unas características que lo convierten en una solución ideal para la aplicación en cuestión al tener un pequeño tamaño, y un muy bajo consumo cuando se encuentra en modo *sleep*. Un valor añadido es que su sensibilidad puede ser fácilmente cambiada mediante software. Sus principales características se recogen en la Tabla 3.4.

Selectable Sensitivity	1.5g/2g/4g/6g
High Sensitivity	800mV/g @ 1.5g
Voltage Operation	2.2-3.6V
Low Current Consumption	500µA

Tabla 3.4 Características del acelerómetro MMA7260QT.

La necesidad de emplear un acelerómetro de varios ejes se debe a que hay que monitorizar medidas en el motor tanto en su eje radial como en el axial (ver

Figura 3.5): unas medidas monitorizan la influencia de las corrientes mientras que las otras hacen referencia a las fuerzas; es decir, mientras que un eje se ve modificado en el caso de que existan problemas en la alimentación del motor, el otro eje tendrá modificaciones si existen problemas de fuerza tales como ejes curvados, rodamientos…

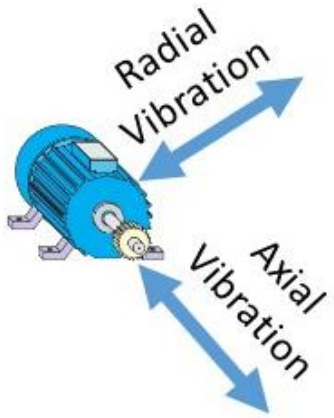


Figura 3.5 Dirección de las vibraciones a medir por el acelerómetro.

El acelerómetro basa su funcionamiento en dos electrodos de área determinada, estos electrodos se encuentran enfrentados con una masa central que se mueve entre ellos tal y como se muestra en Figura 3.6. Cuando se produce un movimiento, éste se traslada a la masa central que lo reproduce produciendo una reducción de la distancia "*d*" entre el dieléctrico y los electrodos, variando así la capacidad. La electrónica del dispositivo transforma esta variación de capacidad en una señal de salida en tensión. Esta salida de tensión es convertida por el microcontrolador de 10-bits de resolución, que a su vez envía los datos en formato entero al transceiver y éste al coordinador de la red. En el nodo se recogen y transfieren unas 2048 muestras (lo que equivale a 16,384 bytes) para la realización de la FFT (*Fast Fourier Transform*), siendo este valor de 131.072 Kb, que es aproximadamente la mitad de la tasa de datos que soporta el estándar IEEE 802.15.4.

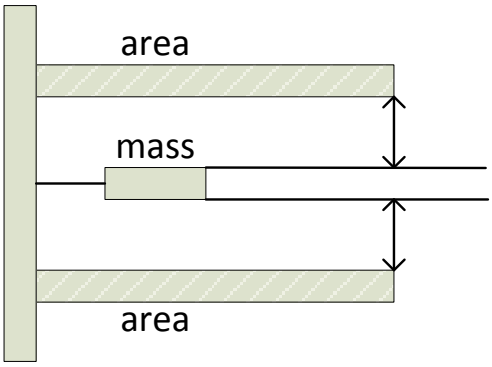


Figura 3.6 Estructura física del acelerómetro.

En el instrumento virtual creado en LabVIEW se convierten las medidas de vibraciones a milímetros por segundos (mm/s). Además, la tasa de muestreo interno del acelerómetro de 11 kHz cumple los requisitos de la mayoría de las aplicaciones relacionadas con la monitorización de vibraciones en motores, ya que la mayoría de los componentes en el espectro de la vibración en motores se encuentran al doble de la frecuencia eléctrica [Tsy13].

Los pasos para caracterizar la adquisición de la vibración son los siguientes: en primer lugar el dato entero recibido se convierte en tensión mediante la ecuación (3.3):

$$V = analogRead * \frac{5.00}{1024} \tag{3.3}$$

donde *analogRead* es un valor entero analógico que llega al microcontrolador. Una vez calculado el valor en voltios, la aceleración se calcula en g o mm/s² mediante (3.4):

$$g = \frac{V}{Sensitivity} \tag{3.4}$$

donde la sensibilidad vale 0.8 mV/g.

Una vez que el valor de g es conocido, la velocidad de muestreo del microprocesador ATmega328 se establece en 33 ciclos para 10 bits de resolución; el datasheets define la frecuencia de ciclo en 1 MHz. Por tanto, el tiempo de adquisición es 1 μs, obteniéndose un tiempo de muestreo total para la adquisición de datos de los cuatro puertos del microcontrolador como se muestra en (3.5) [LuB09]:

$$t_s = \frac{1}{Fs} * 33\ cycles * 4 \qquad (3.5)$$

resultando $t_s = 132\mu s$. A este valor hay que sumar un pequeño retardo por el envío de datos al radio transceiver de 16 μs. Por tanto, la frecuencia de muestreo se establece en $f_s = 6.756kHz$. Finalmente, el resultado final de las vibraciones, se obtiene en mm/s como sigue:

$$vibration = g * 9.81 * (1000 * t_s) \qquad (3.6)$$

3.2.1.5 Estimación del consumo de potencia

Respecto al nodo sensor, se ha realizado un estudio de consumo para determinar la capacidad de la batería, y de ahí poder estimar una autonomía. El estudio que se ha realizado hace referencia al microcontrolador ATmega328, al radio transceiver ATmega128RFA1, y los tres sensores de corriente, multiplicando el consumo por el tiempo de muestreo. El transceiver RF posee un consumo de 250nA en modo *deep sleep*, y 12.5 mA durante las transmisiones. El nodo envía paquetes de 2048 datos, lo que corresponde a 16.384 kB. La

transmisión de 8 bytes tarda 128 μs como se indica en [IEE06]. El consumo de potencia del nodo viene dado por la siguiente expresión [Cas04]:

$$P_{Node} = \frac{P_{deepsleep} * T_{deepsleep} + P_{Rx} * T_{Rx}}{T_{ib}} \quad (3.7)$$

Donde T_{ib} es el tiempo de actividad del nodo. Este tiempo se calcula por:

$$T_{idle} = T_{tb}^{min} * Bytes \quad (3.8)$$

Donde T_{idle} es el tiempo de transmisión, que está determinado por el tiempo de transmisión de un byte (T_{tb}^{min}) y el número de bytes que se desean transmitir. Según marca el estándar IEEE 802.15.4, la transmisión de ocho símbolos requiere 128 μs. Como valor medio se ha establecido un dato por hora, lo que supone 16.384 kB (cada símbolo corresponde a 4 bits). Por tanto, se puede estimar que el nodo permanece despierto por cada transmisión que se realiza en una hora.

El consumo de corriente total del nodo sensor viene dado por:

$$i_{total} = i_{accelerometer} + 3i_{ACS712} + i_{ATmega328} + i_{ATmega128RFA1} \quad (3.9)$$

Donde $i_{accelerometer}$ es 1.39x10^{-7} mAh, $i_{ATmega128RFA1}$ es 0.0497 mAh, $i_{ATmega328}$ es 0.08 mAh y $3i_{ACS712}$ 0.0083 mAh, obteniéndose un valor medio de 0.138 mAh. Estos valores permiten a una batería de 3.3 V y 2600 mAh tener una autonomía de 784 días, algo por encima de los dos años.

3.2.2 Nodo coordinador

En la red en estrella desplegada, los nodos sensores o dispositivos finales se han establecido como dispositivos de función reducida (RFD) para su capacidad de entrar en el modo de reposo (*sleep mode*) en el que el consumo puede considerarse insignificante. El nodo coordinador actúa como dispositivo de función completa (FFD). Este tipo de nodo nunca duerme ya que necesita retransmitir la información de los nodos sensores a la estación base. Por lo tanto, el coordinador también actúa como una pasarela (*gateway*) entre la red de medición y el PC, donde los datos son manejados por el instrumento virtual de LabVIEW. El nodo coordinador también es responsable de gestionar la red y asignar las direcciones a los dispositivos sensores. La Figura 3.7 muestra el diagrama de bloques del nodo coordinador. Incluye el microcontrolador ATmega128RFA1 de 8 bits y el transceptor FT231XS de FTDI. Al igual que en el nodo del sensor se ha implementado una antena en la propia PCB. El nodo coordinador es alimentado desde una fuente externa a través de USB, y se conecta directamente a un PC a través del transceptor TTL-USB. La Figura 3.8 muestra una imagen de la placa del dispositivo coordinador desarrollado.

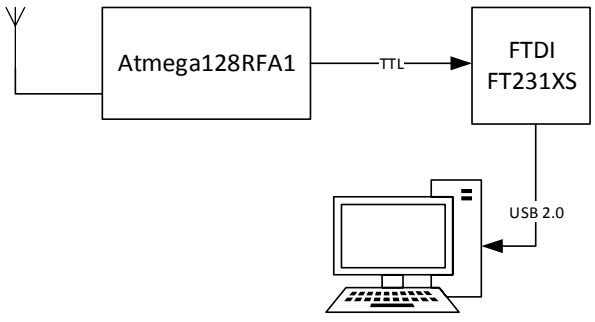
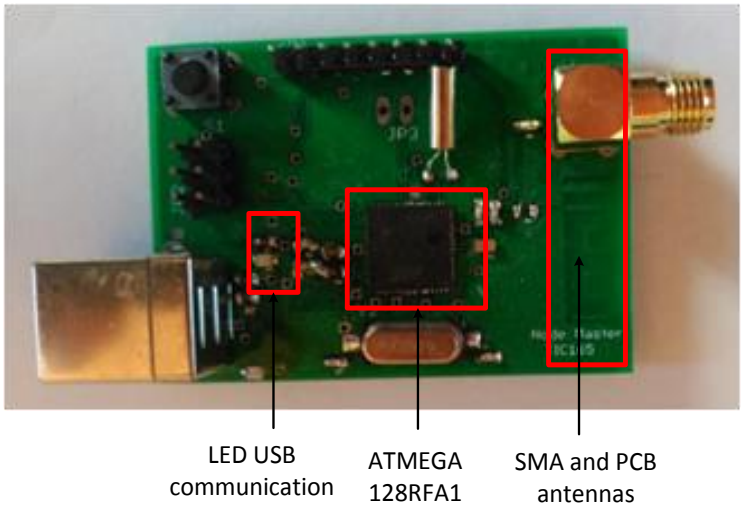
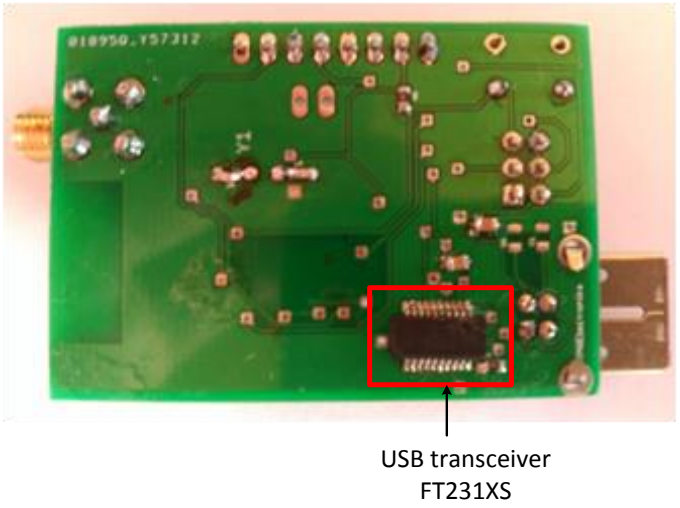


Figura 3.7 Diagrama de bloques de la conectividad del coordinador.



(a)



(b)

Figura 3.8 Vista superior e inferior del nodo coordinador.

3.3 Software

En el Capítulo 2 se realizó una introducción al estándar de comunicaciones IEEE 802.15.4, y se habló del modo de comunicaciones balizado (*Beacon Enabled Mode*) y con GTS (*Guaranteed Time Slots*). Esta modalidad proporciona una baja latencia al tener garantizado un tiempo de transmisión entre los nodos sensores y el nodo coordinador. El nodo coordinador envía periódicamente balizas a los nodos sensores asociados para sincronizarlos. En la trama de envío de datos se ubican dos periodos: uno llamado CAP (*Contention*

Access Period) y el otro CFP (*Contention Free Period*). El primer periodo es utilizado para que cada nodo transmita en el tiempo estipulado por el coordinador; por el contrario, el periodo CFP es utilizado en casos en los que hay mucho tráfico y no se puede transmitir en los periodos CAP. En el periodo CAP los nodos que quieren transmitir compiten por el acceso al medio usando el mecanismo CSMA/CA. Este mecanismo proporciona un acceso al medio en el que cada nodo selecciona aleatoriamente un intervalo de tiempo para transmitir, dentro de un rango determinado por el parámetro configurable *backoff exponent* (BE). Un valor más alto de BE proporciona un mayor rango de elección, y por lo tanto un tiempo de espera medio más alto y una menor probabilidad de colisión. Obviamente, si la transmisión no se puede completar antes del final del área CAP, esto se hará en el área CAP de la siguiente supertrama. Por lo tanto, no se puede asegurar que una transmisión pueda tener lugar dentro de una supertrama correspondiente a ese instante de monitorización. Para evitar este problema, hacemos uso del mecanismo GTS para proporcionar la monitorización de datos con una latencia predeterminada. El nodo coordinador puede asignar hasta siete de estos GTS, y un GTS puede ocupar más de un *slot*. Cada dispositivo que transmite en un GTS asegura que su transacción se complete antes del próximo GTS o al final de la CFP.

Los nodos sensores que transmiten dentro de un GTS no necesitan usar el mecanismo de acceso al medio, porque este intervalo de tiempo está reservado para un dispositivo en particular. Sin embargo, debe establecerse un protocolo de negociación entre el nodo coordinador (que envía las balizas) y el nodo sensor asociado con él. Por lo tanto, el uso de GTS permite a un dispositivo transmitir información en el mismo instante de tiempo, y a una frecuencia determinada por la periodicidad de la supertrama.

De forma más detallada la gestión y el manejo del mecanismo GTS se puede ver en el *Estudio de Tecnologías Inalámbricas*.

3.3.1 Firmware

La Figura 3.9 muestra los módulos utilizados para los mecanismos y la funcionalidad del estándar IEEE 802.15.4 desde la arquitectura MAC de Atmel para los transceptores.

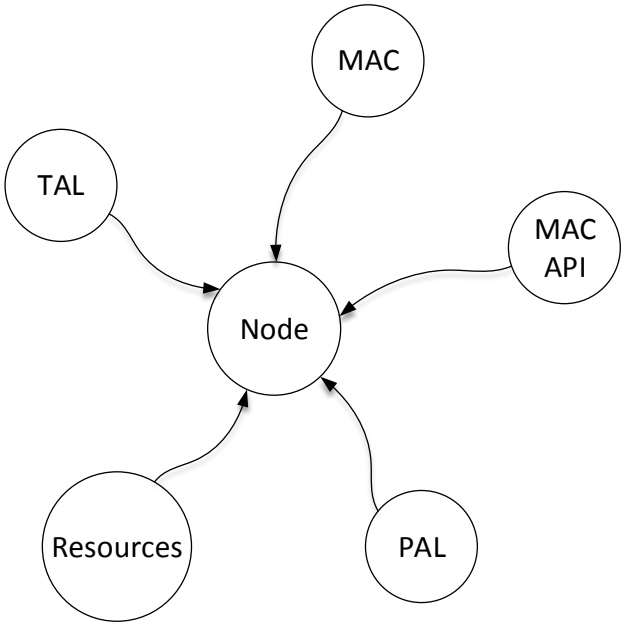


Figura 3.9 Librerías usadas tanto en el coordinador como en los dispositivos *end devices*.

La descripción de los módulos es la siguiente:

- *Platform Abtraction Layer* (PAL): es el módulo de control de las señales de comunicación TTL, interrupciones externas, *timers* y GPIO.

- El módulo MAC (*Medium Access Controls*): es la pila de protocolos para las comunicaciones IEEE 802.15.4. Aquí hay que destacar la librería MAC-API que posee aplicaciones específicas para la norma.

- *Resources*: incluye la gestión de colas y buffers.

- TAL (*Transceiver Access Layer*): este módulo contiene la funcionalidad de control del transceiver.

Los módulos anteriores son usados tanto en el coordinador como en los dispositivos *end devices*.

La programación del coordinador se inicia con la configuración de algunos parámetros tales como *BO* (*Beacon Order*), *SO* (*Superframe Order*), canal, nombre de la red, etc. Los parámetros *BO* y *SO* se encuentran definidos en el estándar y permiten determinar el tiempo entre balizas, y por lo tanto, el tiempo que tarda en transmitir los nodos sensores de la red.

La Figura 3.10 muestra el diagrama de flujo del nodo coordinador. Una vez que los parámetros están configurados para que el nodo coordinador pueda crear la red, el coordinador espera que algún dispositivo final (nodo sensor) envíe una baliza. Cuando el coordinador recibe el primer *broadcast* (baliza de asociación enviada por un nodo sensor), asocia la dirección MAC del dicho nodo a un número de motor, envía una baliza de asociación y espera la respuesta del *end devices*. Una vez que el nodo sensor acuerda estar asociado, ya está dentro del buffer de dirección del coordinador. El coordinador espera el tiempo programado de acuerdo con los parámetros *BO* y *SO* antes de enviar la baliza de petición de datos. Una vez que se envía esta baliza, pueden suceder dos eventos: el coordinador recibe el acuse de recibo desde el nodo sensor y, como éste ya está asociado, los datos pueden ser enviados a la estación base; o, por el contrario, el coordinador no recibe datos, por lo que envía otra baliza. Este último caso está programado en la pila de protocolos durante tres intentos, después de lo cual el coordinador envía a la estación base los datos $QoS = 0$.

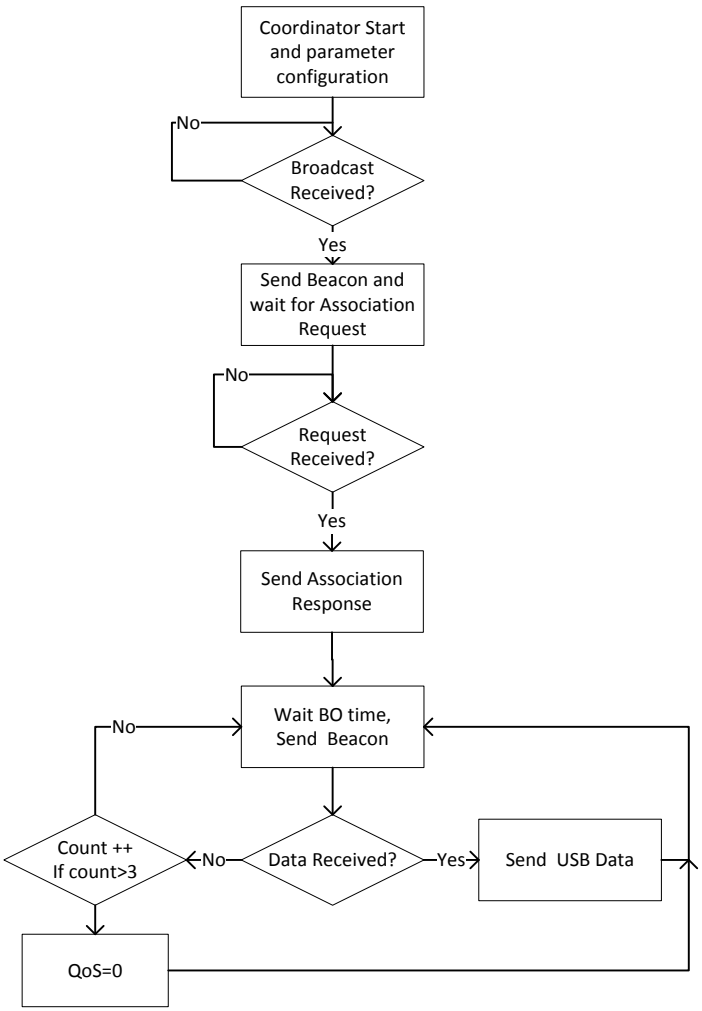


Figura 3.10 Flujo de datos del nodo coordinador.

Como algunos sensores (los sensores de corriente y el sensor de temperatura) no tienen su propia programación de bajo consumo, se desactivan y permanecen así durante el período en que no toman medidas para lograr una estrategia de bajo consumo. La figura siguiente muestra el diagrama de flujo de los nodos sensores.

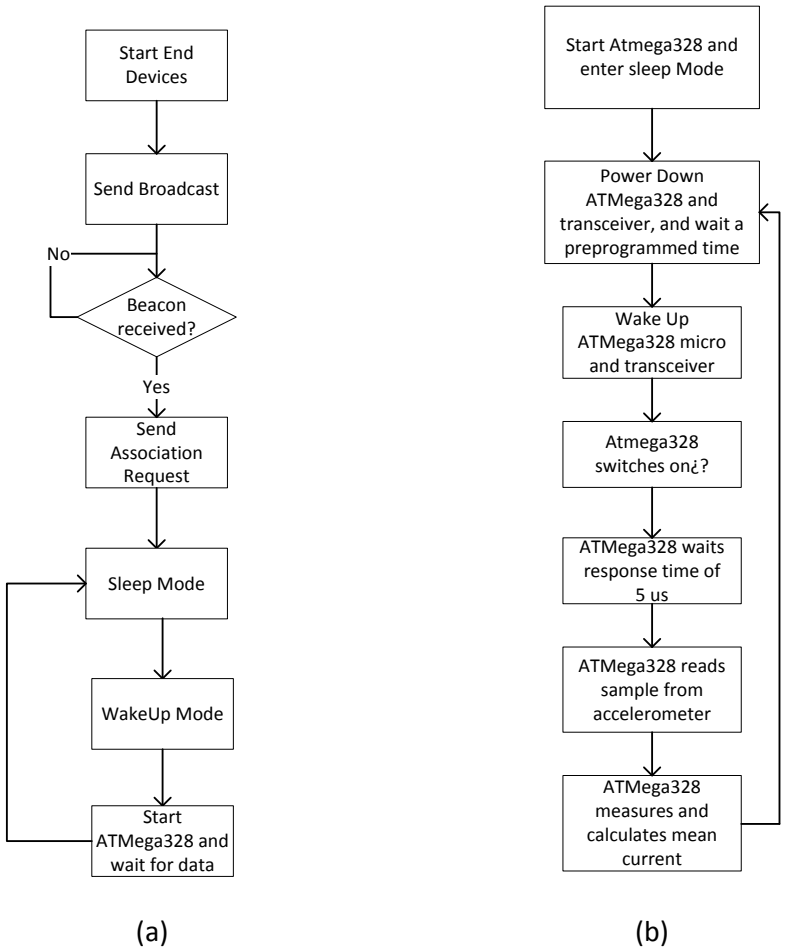


Figura 3.11 Flujo de datos de los nodos sensores. (a) Radio transceiver. b) microcontrolador ATmega328.

En primer lugar, el nodo sensor hace una asociación a la red enviando una trama de asociación conocido como *broadcast* (dentro del estándar que se define como baliza de asociación). Después de la respuesta del coordinador, el nodo sensor acepta la asociación. Algunos datos son incluidos por el nodo sensor en la trama recibida, como cuando se despierta para recibir una baliza y ser capaz de realizar la transmisión de datos.

Después del tiempo especificado, el nodo sensor se despierta y espera la llegada de la baliza. El microcontrolador también se despierta para gestionar los sensores; activa el interruptor analógico (mostrado en la Figura 3.3) y espera un tiempo razonable de 5 μs para realizar un conjunto de mediciones: en

particular, realiza 100 mediciones de corriente y envía el valor medio de corriente al transceptor. En el caso de la temperatura, realiza exactamente lo mismo que con las mediciones de corriente. Finalmente, el microcontrolador recoge las mediciones de los acelerómetros y los envía al transceptor. En este punto, el microcontrolador entra en el modo de reposo y espera en ese estado hasta que el transceptor se despierte de nuevo. El transceptor envía al nodo coordinador los datos a transmitir al instrumento virtual en la estación base.

3.3.2 Aplicación de monitorización

En Figura 3.12 se presenta la interfaz gráfica de usuario (GUI) para el seguimiento de los motores por parte del operario. La interfaz recibe los datos y los guarda en las tablas de base de datos correspondientes de acuerdo con los tipos de datos. Los datos se procesan y analizan, y luego se visualizan gráficamente en tiempo real. La información recibida es relevante para la toma de decisiones, o para tomar las acciones apropiadas. Las principales características de la interfaz gráfica se resumen a continuación. Varias etiquetas (*tabs*) en la parte superior proporcionan las opciones para cada sistema de motor, que en este caso, tienen funcionalidades comunes. Cada etiqueta incluye varios ajustes de configuración, así como los límites de los parámetros principales. Por lo tanto, para cada uno de los nodos sensores se pueden establecer los valores máximos admisibles de temperatura, vibración o desplazamiento del motor y consumo de corriente. En caso de superar estos valores, el sistema activa una alarma. La GUI también muestra los resultados gráficos en el dominio del tiempo y la potencia espectral en el dominio de la frecuencia tanto para vibraciones axiales como radiales. Además, el período de muestreo se puede modificar desde la aplicación del usuario.

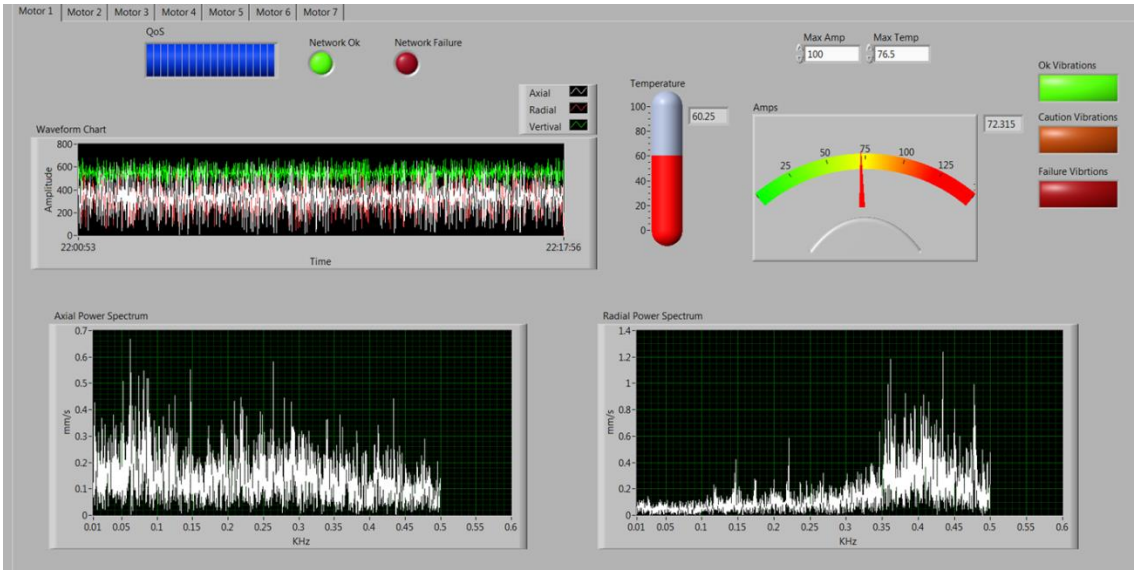


Figura 3.12 Interfaz de usuario diseñada para la monitorización de los motores.

A continuación se detalla el instrumento virtual (VI) desarrollado en LabVIEW que permite recibir y almacenar datos desde el nodo coordinador y que caracteriza la interfaz de usuario. El instrumento virtual se puede dividir en tres grandes bloques: uno que realiza la recepción de información por parte del nodo coordinador mediante el bus de datos USB: otro que es quien realiza el corte (Split) y el tratamiento de datos; y para finalizar la alarma.

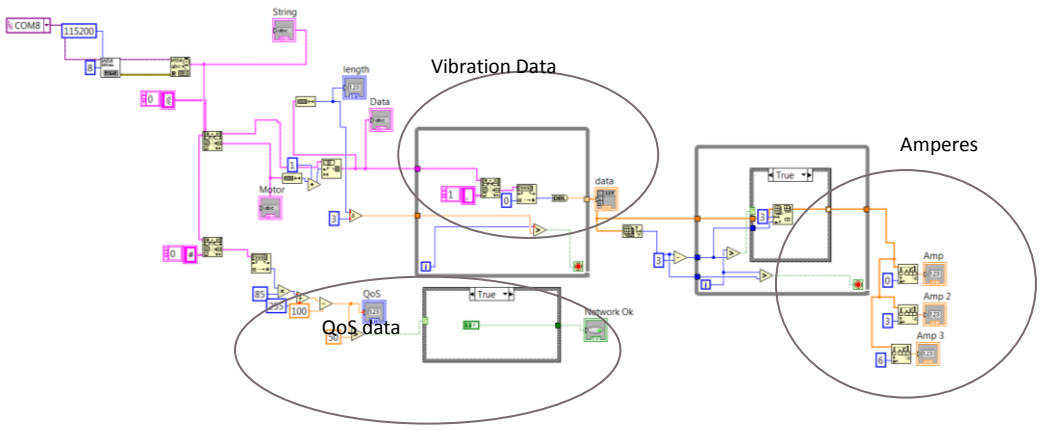


Figura 3.13 Primer Sub-VI: recepción de datos.

En este primer Sub-VI se reciben los datos mediante el puerto USB. Estos datos llegan de la forma que se muestra en la Tabla 3.5:

EngineNumber	@	Data1	,	….	Data2048	,	Amp1	Amp2	Amp3	#	QoS

Tabla 3.5 Trama de parámetros recibida por el Instrumento Virtual.

Se ha utilizado varios caracteres para dividir los datos recibidos siguiendo el siguiente orden:

- *EngineNumber*➔Hace referencia a la dirección corta de un nodo end devices; esta dirección corta dentro del coordinador se le asigna un número de motor. El coordinador envía a la estación base el número de motor.

- @ ➔Separador entre dos tipos de datos distintos. El dato anterior hace referencia al número de motor; sin embargo, los datos posteriores hacen referencia a las medidas de vibraciones.

- *Data1-Data2051*➔Son los datos recogidos y enviados en el momento de las medidas y se encuentran separados por una coma (“,”). Los últimos 4 datos son de corriente y temperatura.

- #➔determina el final de la trama tras el dato *QoS*.

- *QoS*➔dato que hace referencia a la calidad del enlace en el momento de la transmisión.

Los datos enviados son empaquetados en un *array* de vectores y lanzados al sub-VI que los debe de tratar. En la Figura 3.14 se muestra la recepción de datos por separados; una vez recibidos y troceados en vectores, se convierte de hexadecimal a mm/sg para pasar estos datos a la FFT. En la figura también se muestra la toma de decisiones que son realizadas por comparativas impuestas por los operarios a raíz de valores especificados en la hoja de características.

La emisión de alarma se realiza mediante un Express-VI; más concretamente "Play Waveform".

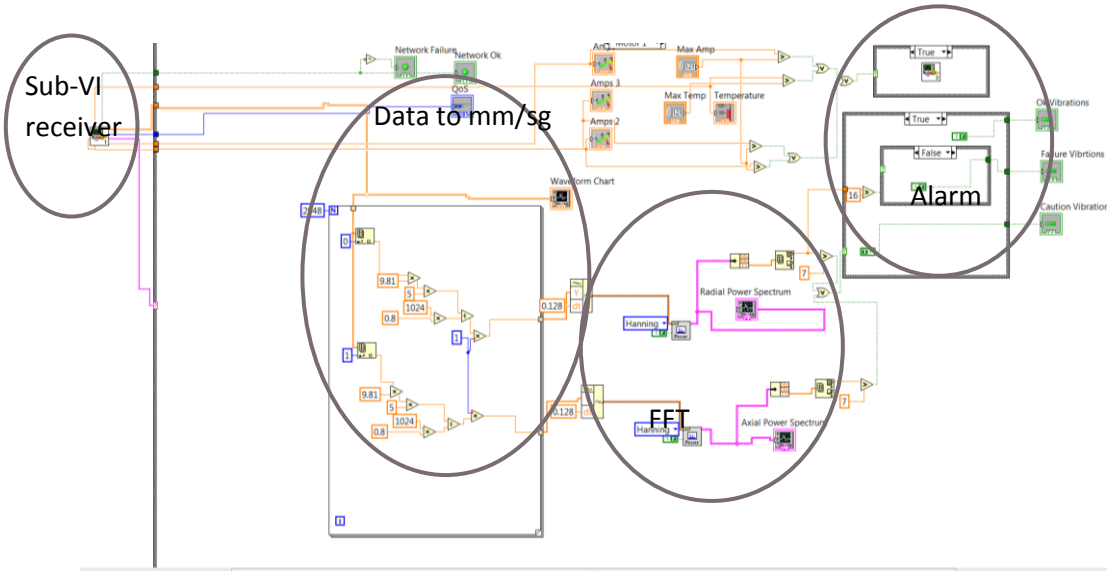


Figura 3.14 Sub-VI: tratamiento de datos.

3.4 Resultados experimentales

Se han realizado una serie de experimentos tanto en laboratorio como en campo para validar la red de sensores inalámbricos. En primer lugar, se han realizado pruebas de laboratorio para verificar la fiabilidad de la red inalámbrica, seguidas de mediciones directas de un motor de AC. Otros experimentos realizados son la medida de corriente, de vibraciones y se ha buscado una sinergia entre la tasa de datos y el correcto funcionamiento de la detección de errores por vibraciones, buscando el mínimo tiempo de transmisión posible. Finalmente, se han realizado pruebas de campo en una empresa local.

3.4.1 Fiabilidad de la comunicación inalámbrica

Debido al ruido electromagnético que producen los motores AC, y la proximidad del nodo sensor a los motores, es necesario realizar un test de

fiabilidad de la comunicación inalámbrica. Para ello se usaron antenas cerámicas integradas en el módulo inalámbrico, con una ganancia de 0 dBi.

Como primer paso se ha medido el ruido de fondo en la banda de interés en el entorno de trabajo, sin ningún motor en marcha. Para realizar el test, se ha configurado el transceiver habilitando los 16 canales en la banda de 2.4 GHz. Los resultados se pueden observar en las Figura 3.15(a). A continuación, usando la misma configuración en el transceiver, el ruido de fondo se midió colocando el nodo inalámbrico entre dos motores funcionando, como se muestra en la Figura 3.15(b). La comparación entre las dos señales de ruido se muestra en la Figura 3.16, donde ambas han sido previamente filtradas.

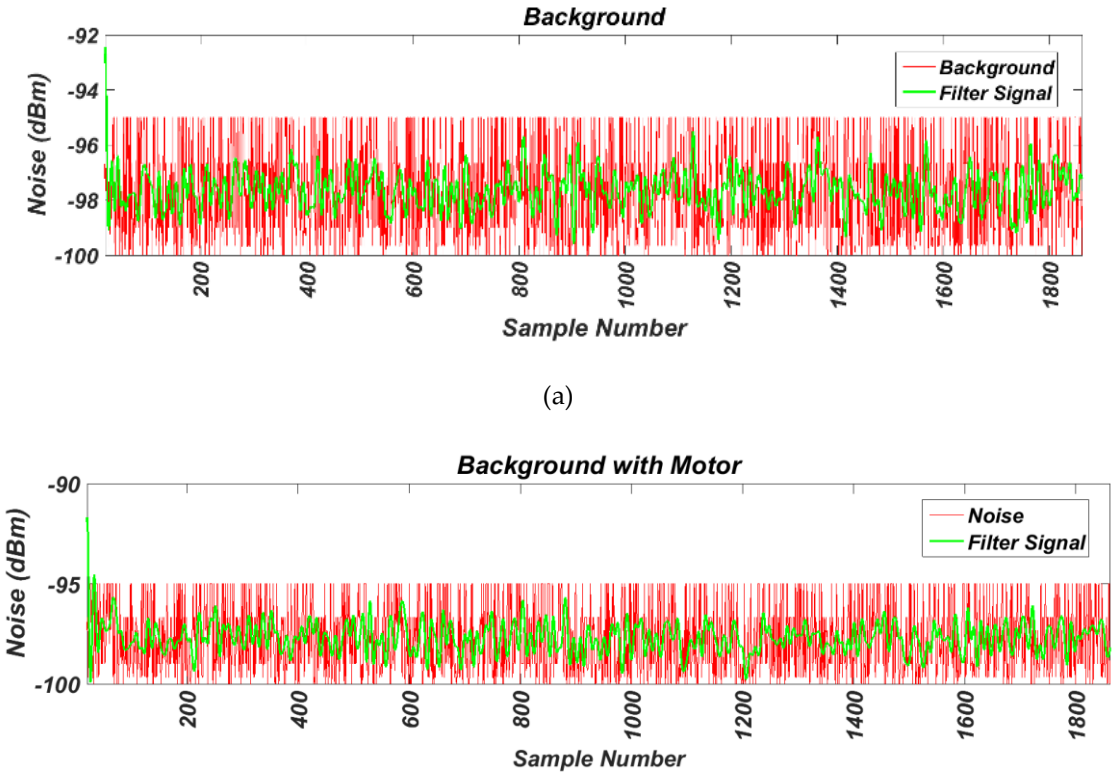


Figura 3.15 (a) Ruido de fondo sin motores. (b) Ruido con los motores funcionando.

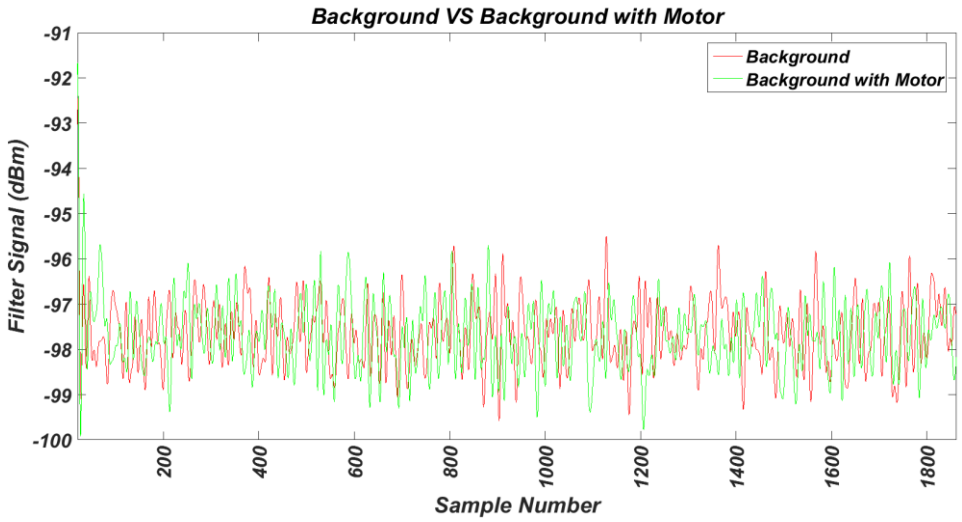


Figura 3.16 Comparación del ruido de fondo medido y filtrado con los motores apagados y encendidos.

Para evaluar los resultados, calculamos la varianza y los valores medios de las señales, obteniéndose los siguientes resultados:

$VarSignal = 34.943$

$MedSignal = -97.2688$

$VarEngine = 35.0387$

$MedEngine = -97.3068$

Como se puede observar, el ruido provocado por los motores no afecta a la comunicación inalámbrica.

3.4.2 Experimentos en laboratorio

Para estos experimentos se ha utilizado el motor trifásico 1LA7063-4AB11 de Siemens [Sie16]. El motor funciona a una tensión de 400V, con una potencia nominal de 0.18 kW, y una velocidad de 1350 rpm. Se ha realizado un test en laboratorio acorde a la norma ISO 10816-1 para la evaluación de medidas de

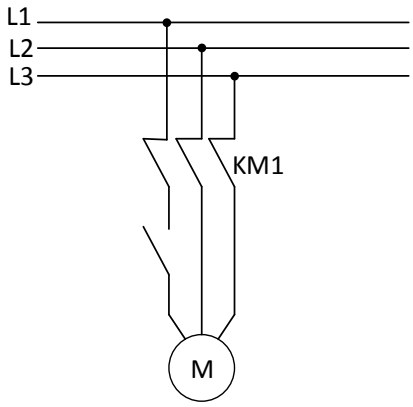
vibraciones en máquinas rotatorias [ISO10], la cual clasifica el tipo de motor y define la gravedad de las vibraciones. El motor usado es de Clase I (máquinas pequeñas), donde vibraciones superiores a 7.1 mm/s pueden dañar de forma irreversible el equipo. La metodología usada para el testeo del equipo es la siguiente:

a) Se realiza un arranque directo del motor y un muestreo de las vibraciones a 6.756 kHz.

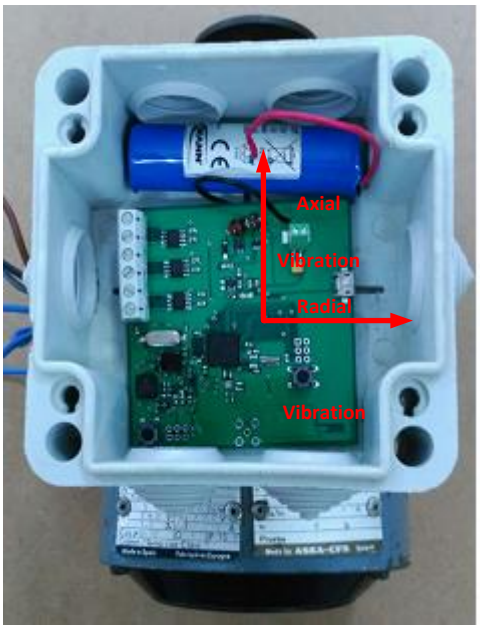
b) Se desconecta una fase del motor de forma intencionada, como se muestra en la Figura 3.17(a), provocando una avería en los devanados del motor lo que conduce a un aumento considerable de las vibraciones. El sistema genera una alarma cuando se superan los valores máximos.

La Figura 3.17(b) muestra el motor y la colocación del nodo del sensor en su parte superior. La Figura 3.17(c) muestra una fotografía de la disposición del nodo sensor dentro de una caja de protección que proporciona blindaje contra las condiciones ambientales.

En los motores AC, las vibraciones pueden ser causadas por varios tipos de fallos. Un estudio de los tipos de fallos en motores y las frecuencias naturales que llegan a alcanzar aparece recogido en [Nan05]. Las vibraciones pueden ser axiales y radiales. Las vibraciones axiales se generan en la dirección del flujo, y las vibraciones radiales ocurren en la dirección vertical al eje del motor.



(a)



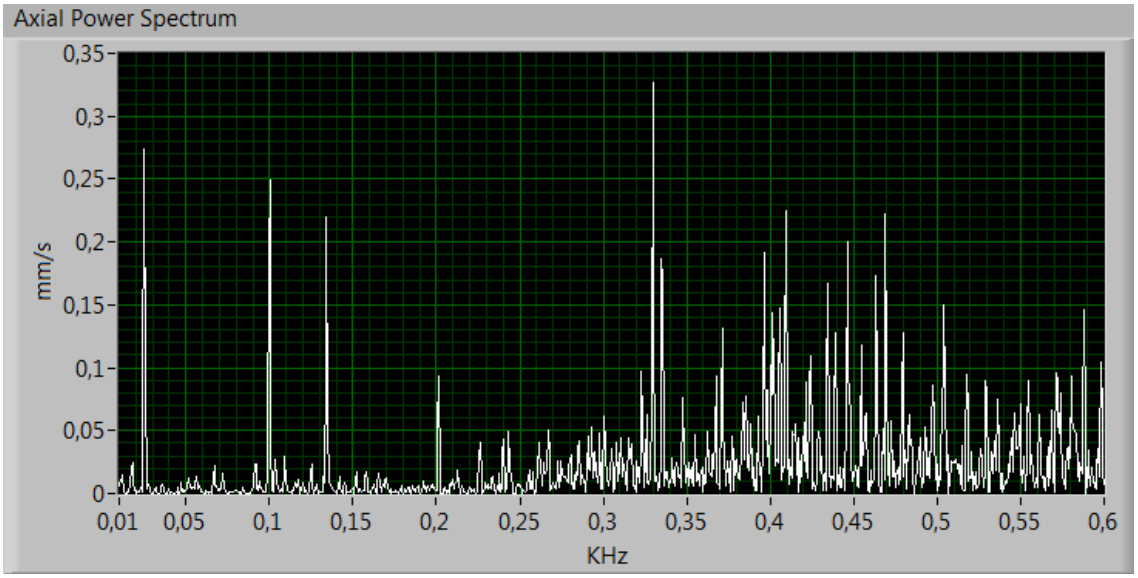
(b)

(c)

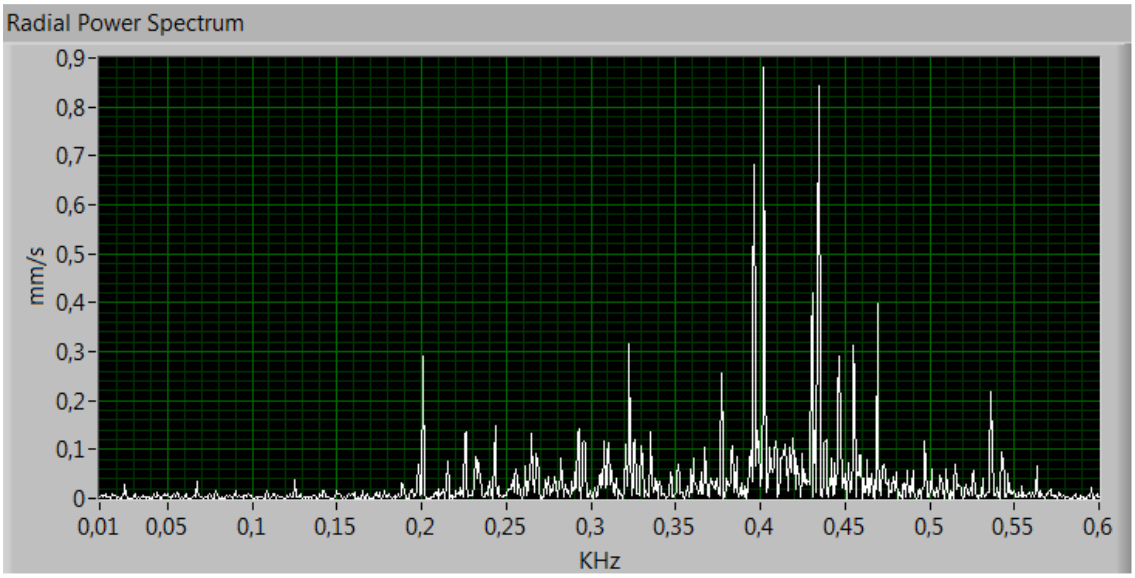
Figura 3.17 (a) Esquema sobre la eliminación de una fase en el motor. (b). Fotografía detallada del nodo sensor dentro de una caja de protección (c) Ubicación del nodo sensor en el motor.

La viabilidad general del sistema se ha validado a través de un conjunto de experimentos de laboratorio. Para medir las vibraciones se empleó el

acelerómetro de tres ejes, de los cuales sólo se utilizaron dos. La Figura 3.18 muestra el espectro de potencia de las vibraciones axial y radial medidas cuando el motor funciona correctamente.



(a)



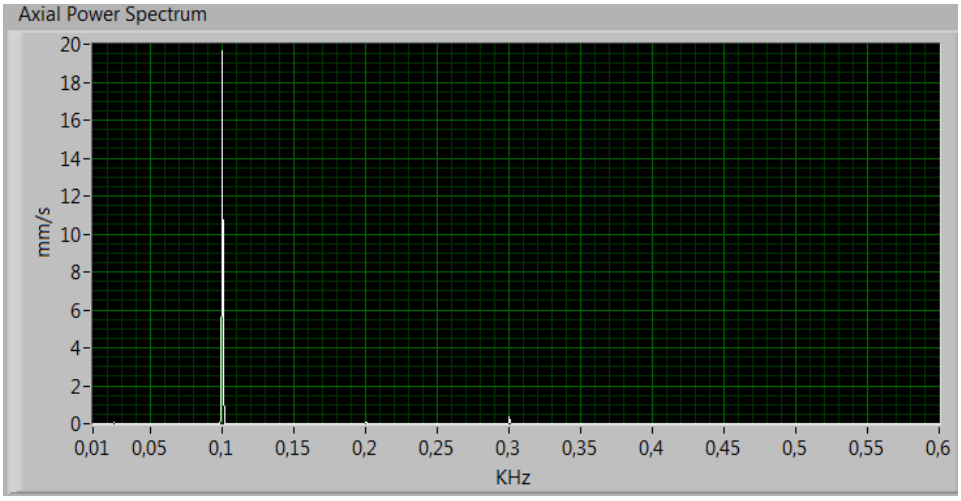
(b)

Figura 3.18 Resultados experimentales del espectro de potencia de las vibraciones cuando el motor funciona correctamente. (a) Vibraciones axiales. (b) Vibraciones radiales.

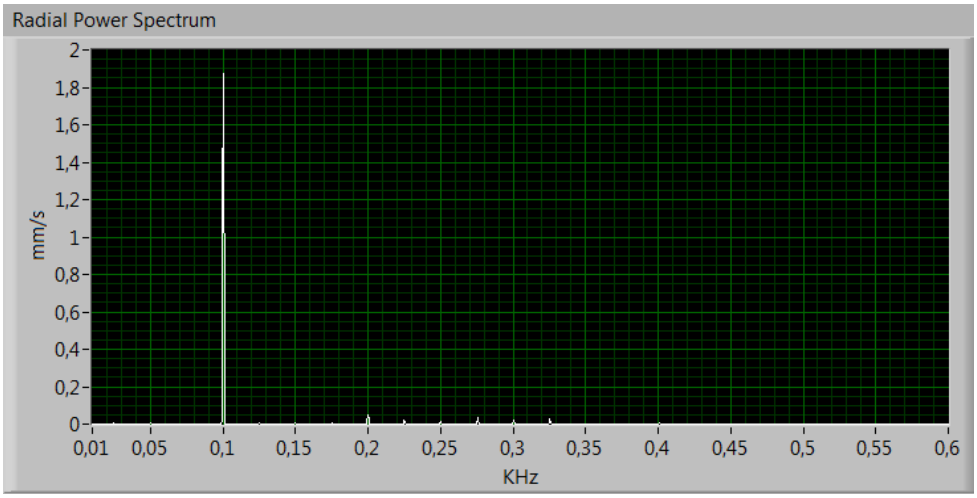
Se puede observar que ambos valores están por debajo de 1 mm/s, por lo que según [ISO10], la máquina no sufre ningún tipo de problemas. Aunque el motor no presenta ningún fallo importante, se evidencia que sí tiene algún tipo de desgaste posiblemente relacionado con desalineación o un rodamiento dañado, ya que es un motor que tiene más de 15 años y que ha sido sometido a muchos tipos de experimentos de laboratorio, algunos de ellos muy agresivos.

El siguiente test consiste en provocar un fallo en una de las líneas de alimentación del motor [Ver07]. En consecuencia, el flujo magnético de la corriente a través del devanado produce un incremento de las vibraciones en el eje axial como se observa en Figura 3.19. Se puede observar el gran incremento en las vibraciones axiales alcanzando casi los 20 mm/s, que pueden dañar el motor rápidamente, y por tanto, es necesario pararlo; por contra, las vibraciones radiales se incrementan sólo ligeramente. Teniendo en cuenta que es un motor que opera con una señal AC de 50 Hz, de acuerdo a [Tsy13], tiene una componente en 100 Hz, como se muestra en ambas figuras.

Los datos enviados al nodo coordinador de la red han sido optimizados, buscando un compromiso entre el consumo, la transmisión de datos y la fiabilidad de los datos recibidos. De ahí que los datos tomados por el acelerómetro sean de 2048 muestras para calcular la FFT.



(a)



(b)

Figura 3.19 Resultados experimentales del espectro de potencia de las vibraciones cuando el motor funciona correctamente. (a) Vibraciones axiales. (b) Vibraciones radiales.

En cuanto a los sensores de corriente, estos son capaces de detectar un fallo en cualquier fase que se muestra en la Figura 3.20. En la Figura 3.20(a) se recogen las medidas de corriente de cada uno de los sensores. Se puede observar cómo al principio las medidas de corrientes son similares en las tres fases, cuando el motor empieza trabajando correctamente. Tras pasar un tiempo en régimen permanente, se decide de cortar una de las fases (fase 3 en la figura),

observándose dos hechos: la corriente de esa fase es cero, y la corriente en las otras dos fases se incrementa significativamente. El sistema está diseñado para alertar de ambos efectos al establecer unos umbrales mínimos y máximos para la corriente en cada fase. En Figura 3.20(b) se muestra un zoom en la zona donde se ha cortado la fase 3 para poder visualizar una comparativa y ver los efectos comentados con mayor detalle.

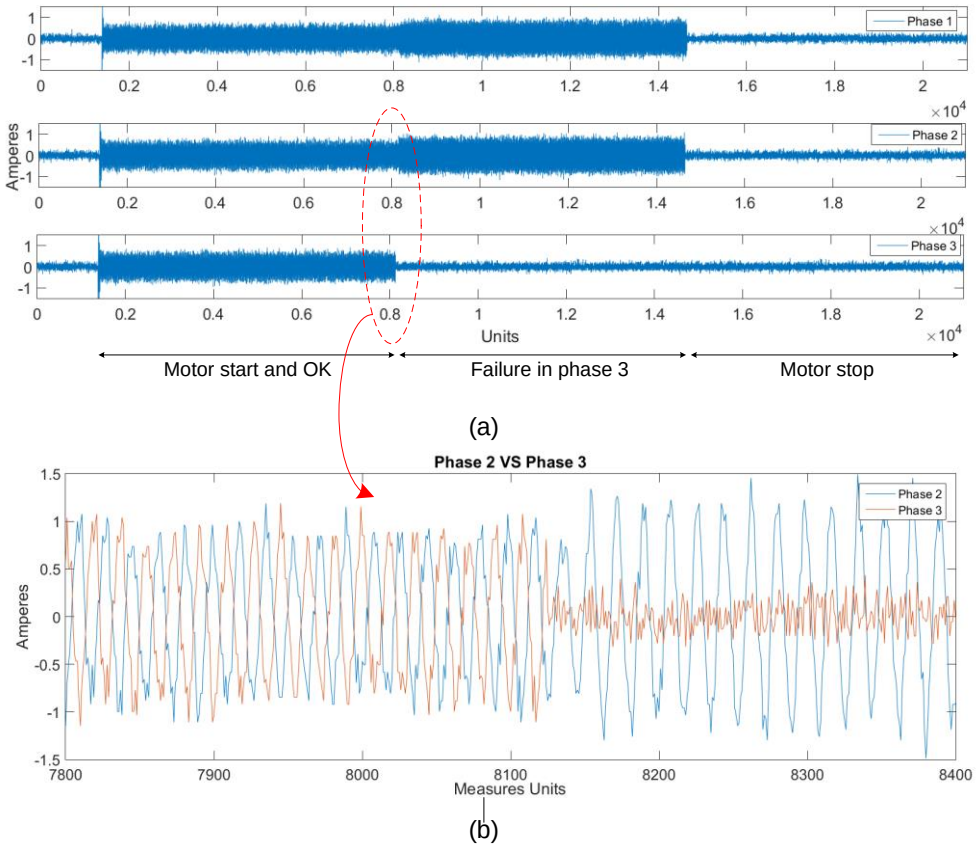


Figura 3.20 Detección de un fallo en una fase por el sensor de corriente. (a) Corriente medida para las tres fases. (b) Zoom de las corrientes de las fases 2 y 3 en el área donde falla la fase 3.

3.4.2.1 Experimentos de campo

Para probar el sistema en un entorno industrial se han realizado ensayos de temperatura y vibración (de forma no intrusiva) en la empresa Suavizantes y Plastificantes Bituminosos S.L. (S.P.B.) situada en Huévar, Sevilla, la cual posee una experiencia de cinco décadas en la industria química. Los operadores han

instalado el sistema de sensores inalámbricos en un grupo de bombeo refrigerado. La Figura 3.21 muestra cómo se coloca el sistema sobre la carcasa del motor.

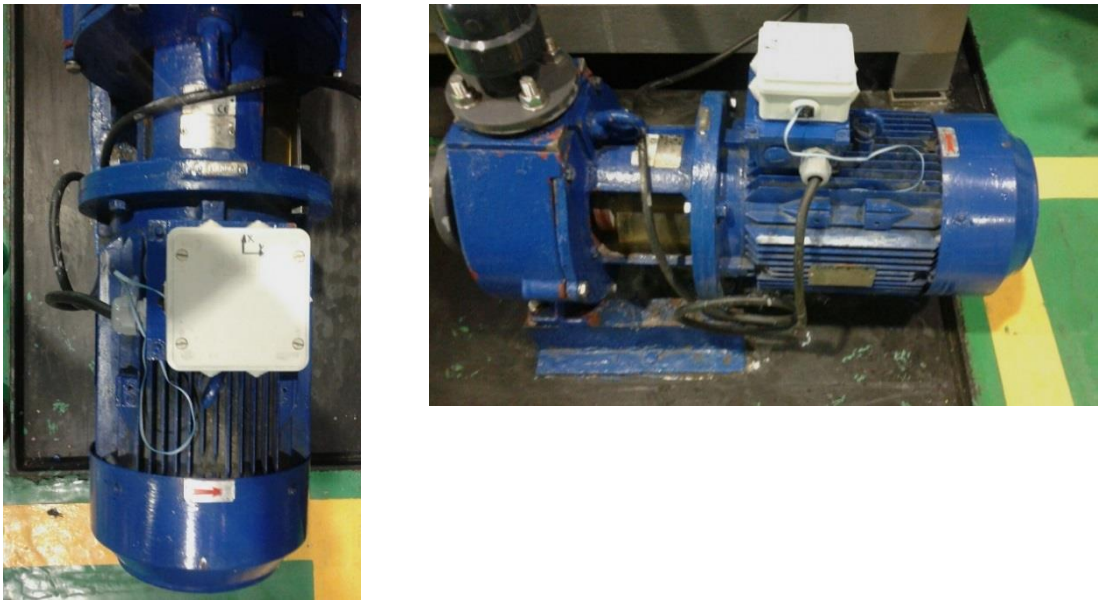
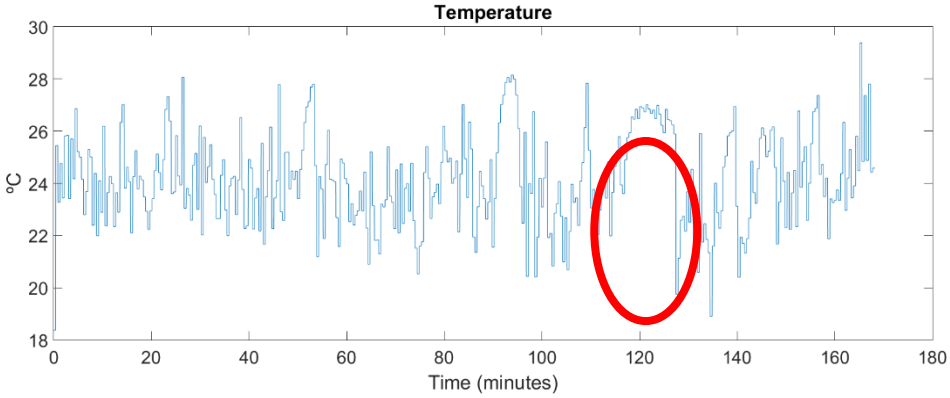
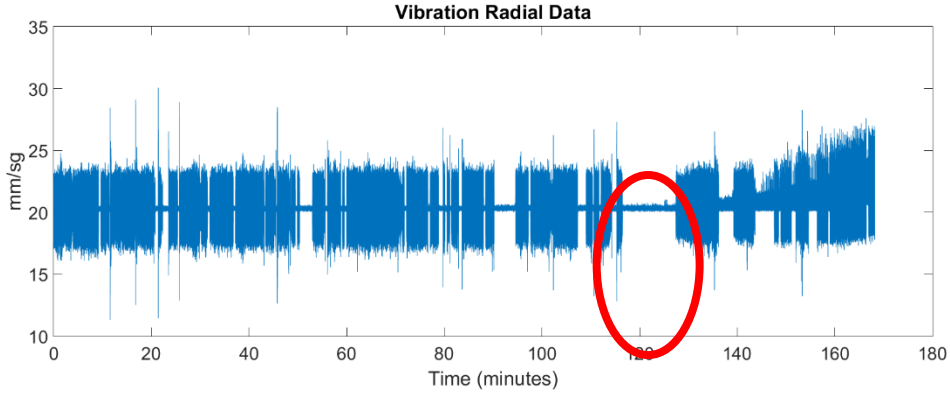


Figura 3.21 Lugar experimental para la prueba de campo en una empresa del entorno.

El tipo de motor es un 1AR132S2 de WA Motors con los siguientes parámetros: potencia nominal de 7.5 kW, tensión de 400V/690 V y velocidad nominal de 2915 rpm. Cuando se arranca el motor para bombear líquido en el proceso de fabricación de productos químicos, se utiliza un ventilador para enfriarlo. La Figura 3.22(a) muestra las medidas recogidas por el sensor de temperatura, que se encuentra físicamente por encima de la carcasa del motor. Se puede ver cómo el sistema de refrigeración del motor actúa cada vez que sube la temperatura del motor. La Figura 3.22(b) muestra las vibraciones medidas en tiempo real; se distingue claramente los períodos en los que el motor está en marcha y los que se encuentra parado.



(a)

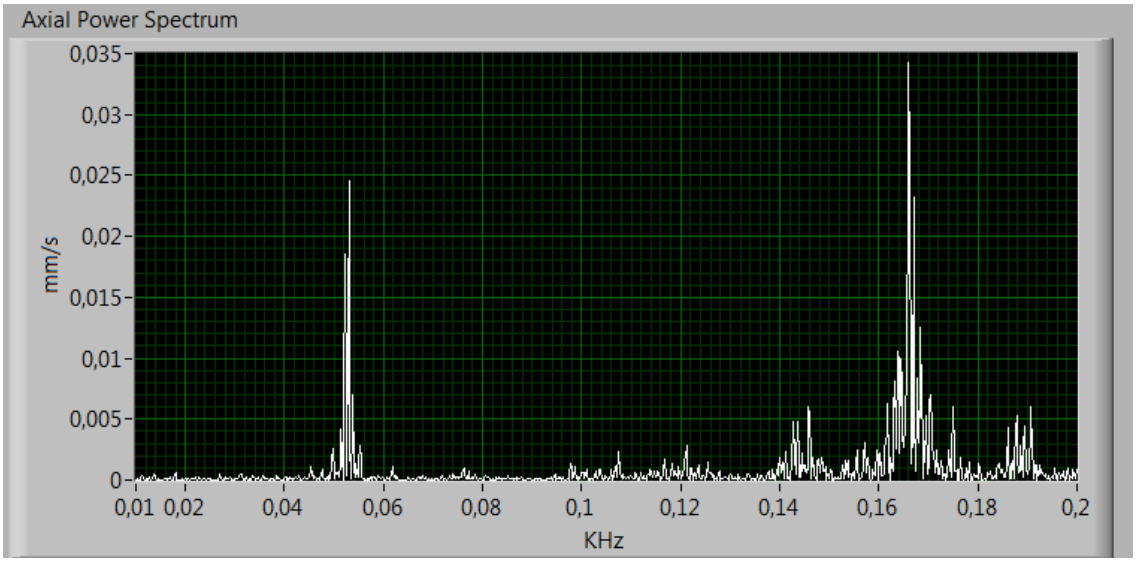


(b)

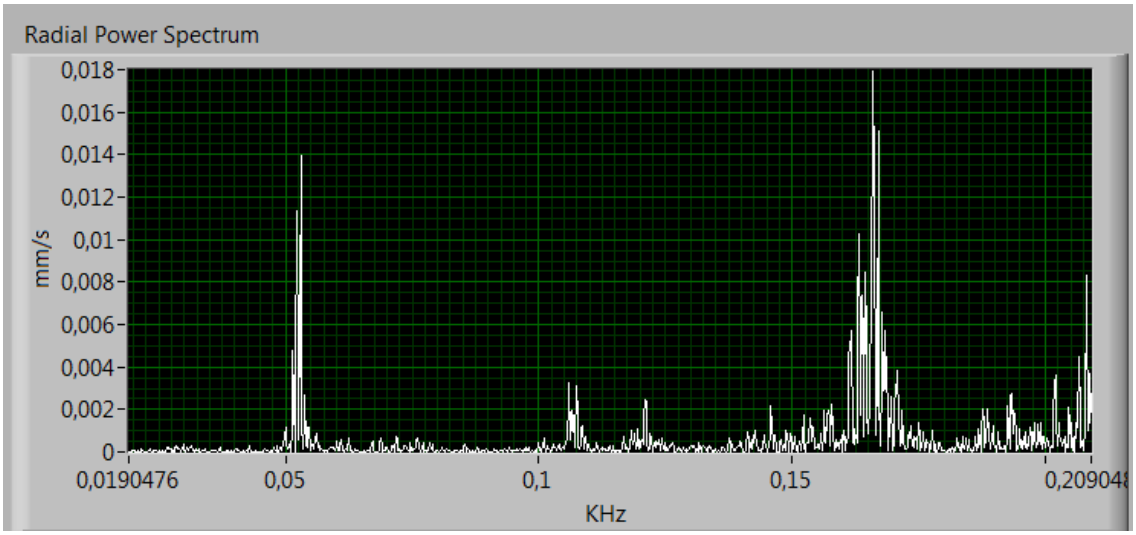
Figura 3.22 (a) Medidas experimentales de temperatura. (b) Medidas de vibraciones en el dominio del tiempo.

Comparando las figuras anteriores puede comprobarse que cuando el grupo de bombeo comienza a funcionar, su temperatura disminuye puesto que el sistema de refrigeración entra en funcionamiento. Por ejemplo, entre los minutos 117 y 127 el motor se detiene. Cuando arranca de nuevo (en el minuto 127), el sistema de enfriamiento comienza a funcionar bajando la temperatura del motor. Se aclara que cada vez que el motor se detiene, su temperatura tiende a subir a la temperatura ambiente de la habitación donde se encuentra.

La Figura 3.23 muestra la FFT realizada sobre las mediciones de las vibraciones (en el dominio del tiempo) y obtenida en LabVIEW. Se puede observar que el motor no presenta ningún tipo de fallo mecánico.



(a)



(b)

Figura 3.23 Espectro de potencia de las vibraciones medidas en las pruebas de campo. (a) Vibraciones axiales. (b) Vibraciones radiales.

3.5 Conclusiones

En el presente capítulo se ha presenta una red de sensores inalámbricos para la detección precoz y la monitorización de fallos en motores. El sistema ha sido diseñado para combinar varias mediciones de parámetros en tiempo real, mejorando la detección de fallos. La monitorización del sistema motor implica la medición de varios parámetros: vibraciones, temperatura y consumo de

corriente. Por lo tanto, en comparación con la técnica convencional que se basa exclusivamente en vibraciones, este diseño tiene dos fuentes de información más que pueden disparar una alarma. Las comunicaciones inalámbricas se basan en el estándar IEEE 802.15.4 en modo *beacon-enabled* y con slot de tiempo garantizado (GTS). Esto asegura la transmisión de datos y una adquisición síncrona, que son elementos críticos en un sistema de monitorización de la condición del motor basado en una red de sensores inalámbricos. Los datos recibidos por el nodo coordinador son almacenados y evaluados a un PC. Los datos se presentan gráficamente en tiempo real mediante un instrumento virtual desarrollado en LabVIEW. El sistema propuesto puede ampliarse fácilmente incluyendo otro tipo de sensores en el nodo de detección para la medición de otros parámetros de interés; o para añadir nuevos nodos sensores a la red inalámbrica. El sistema tiene una alta autonomía, fácil instalación y reducidos costes de mantenimiento. Las mediciones experimentales confirman la viabilidad de la implementación de la red de sensores y su utilidad para el mantenimiento preventivo en maquinaria rotativa trifásica.

Capítulo 4. Plataforma experimental para el estudio de la vulnerabilidad hardware de los robots móviles

4.1 Introducción

En la última década, el número de aplicaciones robóticas utilizadas para solucionar problemas cotidianos o vinculados a tareas de seguridad ha crecido considerablemente. Así, está fuera de toda duda la utilidad de los robots en multitud de campos de aplicación: desde los relacionados con la vigilancia y el rescate de personas [Ryb00], [Lim03],[Gar07], hasta su utilización en el ámbito del hogar para tareas asistenciales o de limpieza [Bar02], [Tak06],[Jar08]. En este sentido, no hay que olvidar la ya contrastada eficiencia de soluciones robóticas en el mundo de la industria, donde realizan un amplio rango de actividades, muchas de ellas de naturaleza crítica [Mor12],[Val15].

Es por esto que, paralelamente a dicho desarrollo, se han publicado numerosos trabajos que estudian y proponen arquitecturas para aumentar la fiabilidad y la tolerancia a fallos de los sistemas robóticos tales como [Lad04], [Bas04], [Fer11], [Cañ14]. No obstante, este hecho contrasta con el relativamente bajo número de publicaciones que abordan el estudio de la vulnerabilidad de los robots.

En contraste con la calificación de fiable, un sistema se considera vulnerable si el fallo de alguno de sus componentes pone en riesgo su integridad. La vulnerabilidad engloba tanto los fallos generados por causas estocásticas, como aquellos que pueden ser provocados de forma intencionada. Por tanto, el análisis de la misma abarca un espacio de problemas más amplio que el de aquellos que tradicionalmente son considerados como fallos más probables.

Aunque la vulnerabilidad ha sido estudiada en lo que se refiere al diseño del software [Aro08],[Hee11], la investigación sobre la vulnerabilidad robótica no ha sido abordada con mucha asiduidad; de hecho, apenas se ha considerado el efecto de fallos o anomalías hardware que supongan un riesgo para el propio robot. En este sentido, cabe destacar algunos trabajos pioneros que estudian la vulnerabilidad del hardware de robots frente a ciertos tipos de ataques externos como son [Nob12],[She17].

En [Mag17] se aborda de una forma más exhaustiva posibles escenarios en los que se puede intentar ataques a sistemas robóticos, más concretamente a manipuladores industriales. En función de la forma del ataque, los autores hacen una clasificación de los mismos en cinco categorías:

- Ataques del tipo I. Son ataques dirigidos a modificar los parámetros del lazo de control.

- Ataques del tipo II. Son ataques destinados a modificar los parámetros de calibración.

- Ataques del tipo III. Son ataques destinados a la manipulación de las tareas que debe ejecutar el robot.

- Ataques del tipo IV. Son ataques destinados a alterar la comunicación desde el robot al usuario, de tal forma que el usuario no conozca el estado real del robot.

- Ataques del tipo V. Son ataques destinados a alterar el estado real del robot, de tal forma que se pierda el control del mismo.

Aunque los autores basan estos modelos en ataques software, dicha clasificación puede ser aplicada a cualquier ataque independientemente de la naturaleza del mismo (ya sea software o hardware).

Debido a la estructura de los sistemas robóticos modernos, las vulnerabilidades más críticas afectan al hardware de control y actuación. En la actualidad, prácticamente todas las plataformas robóticas se implementan en base al uso de microprocesadores. Aunque, las vulnerabilidades de los sistemas basados en microprocesadores se han analizado en diversos trabajos como son [Hua03], [Bru05], [I10], [Kar13], el estudio específico de la vulnerabilidad de los sistemas de control y actuación de bajo nivel no ha sido suficientemente abordado.

Recientemente se han publicados trabajos que consideran las consecuencias de ciertos ataques hardware en sistemas robóticos [Gom15],[Gom16]. Estos trabajos abordan el estudio de los ataques a la señal de reloj en el proceso de comunicación con los controladores de bajo nivel. Sin embargo, en ellos se contempla exclusivamente la viabilidad y efectos que puede tener la inserción intencionada de fallos sin describir el procedimiento necesario para hacer realidad la inserción, ni los detalles de la plataforma utilizada para ello. En contraste, el objetivo de este trabajo se centra en la descripción de una plataforma "*hardware configurable*" utilizada para el estudio

de la vulnerabilidad y el correspondiente comportamiento de los módulos que permiten emular la inserción de fallos.

Por "*hardware configurable*" se entiende que todos los elementos hardware que intervienen en el proceso pueden ser monitorizados, y, si es el caso, alterado su comportamiento durante el funcionamiento del robot, simulando de esta forma una situación de fallo. Dentro del concepto de controladores de bajo nivel se engloban todos aquellos elementos responsables de asegurar que los sistemas de actuación del robot se comportan de acuerdo con lo especificado por elementos de jerarquía superior.

Con este fin, se ha diseñado un sistema instrumental y una plataforma que emula la cinemática de un robot diferencial. La plataforma está configurada alrededor de un dispositivo FPGA (*Field Programmable Gate Array*) que implementa los módulos más críticos que deben ser convenientemente monitorizados. Por su parte, el sistema instrumental posibilita la realización de múltiples medidas digitales y analógicas que permiten caracterizar los comportamientos anómalos de los elementos vulnerados. Más concretamente. En el capítulo se presenta un caso de estudio en el que se analiza la vulnerabilidad del bus I2C en situaciones en las que la señal de reloj se ve alterada. Hasta ahora, han sido pocos los trabajos que han abordado el estudio de la tolerancia a fallos en el bus I2C [Fuk04],[Alk06]. En particular, la vulnerabilidad propuesta en los siguientes puntos no ha sido analizada con profundidad, y mucho menos, el efecto de la misma en la ejecución de tareas de navegación en robot móviles. Sin embargo, como quedará demostrado, la repercusión de ésta puede ser dramática, lo que justifica ampliamente que su estudio merezca la atención de la comunidad científica.

4.2 Descripción de la plataforma

La plataforma desarrollada tiene como objetivo emular lo más fielmente posible el comportamiento de un robot móvil, permitiendo, durante el proceso de navegación, la interacción y/o alteración los elementos de control de bajo nivel. En esta sección se describen los distintos componentes de la plataforma detallando: las características de la arquitectura de control empleada; el controlador de alto nivel; la implementación de los módulos de bajo nivel y el sistema de instrumentación y medida.

4.2.1 Arquitectura

La arquitectura de control que ha inspirado el diseño de la plataforma mostrada en Figura 4.1, sigue las líneas tradicionales que han sido propuestas y ampliamente utilizadas en la bibliografía sobre robótica móvil [Oll94], [Gom01], [Min04], [Cue04].

En ella, un controlador de alto nivel es el encargado de tomar decisiones con cierto grado de abstracción, mientras que otro u otros controladores, que funcionan de forma concurrente con el primero (controladores de bajo nivel), son los responsables de asegurar que los motores sigan el perfil cinemático correspondiente.

El bucle de control se cierra en ambos niveles: los controladores de bajo nivel consideran individualmente los valores de las velocidades de cada una de las ruedas, mientras que un módulo de localización considera distintos tipos de información, con el fin de determinar la posición y orientación del robot y asegurar el cierre del bucle en alto nivel.

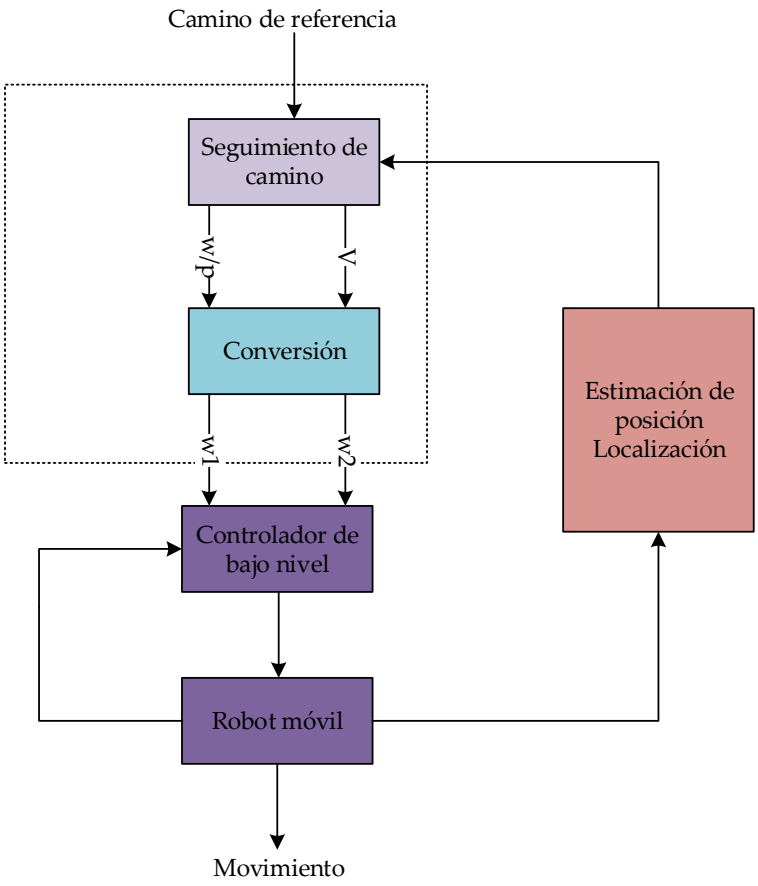


Figura 4.1 Arquitectura tradicional de control de un robot móvil.

La plataforma se divide en tres zonas diferentes: un ordenador personal (PC) que ejecuta un programa, responsable entre otras cosas de actuar como controlador de alto nivel; un dispositivo FPGA para implementar los módulos controladores de bajo nivel; y un dispositivo I2C esclavo estándar, para tareas de verificación, responsable de controlar el movimiento de los motores, y por tanto, de las ruedas del robot.

En Figura 4.2(b) se muestra una fotografía de la plataforma real sobre la que se han realizado los experimentos (PC, FPGA, y controlador de motores, al que se le ha unido el sistema de potencia), mostrando, en negro, el flujo básico de información entre ellos. En primer lugar, el PC es el encargado de generar la trayectoria que debe seguir el robot enviando los valores de velocidades para las ruedas. Estos valores son enviados a la placa FPGA mediante una conexión

RS232, los cuales serán procesados para su envío al controlador de los motores a través de un protocolo I2C. Dicho controlador actualiza el valor de las velocidades de los motores, y envía de vuelta el valor de los encoders para conocer la nueva posición exacta del robot. Dicha posición es enviada a la placa FPGA vía I2C para ser procesada y enviada al PC vía RS232. El PC, con la posición real, obtiene los valores de las nuevas velocidades para seguir la trayectoria deseada. Este flujo de información se mantiene hasta que se haya completado la trayectoria.

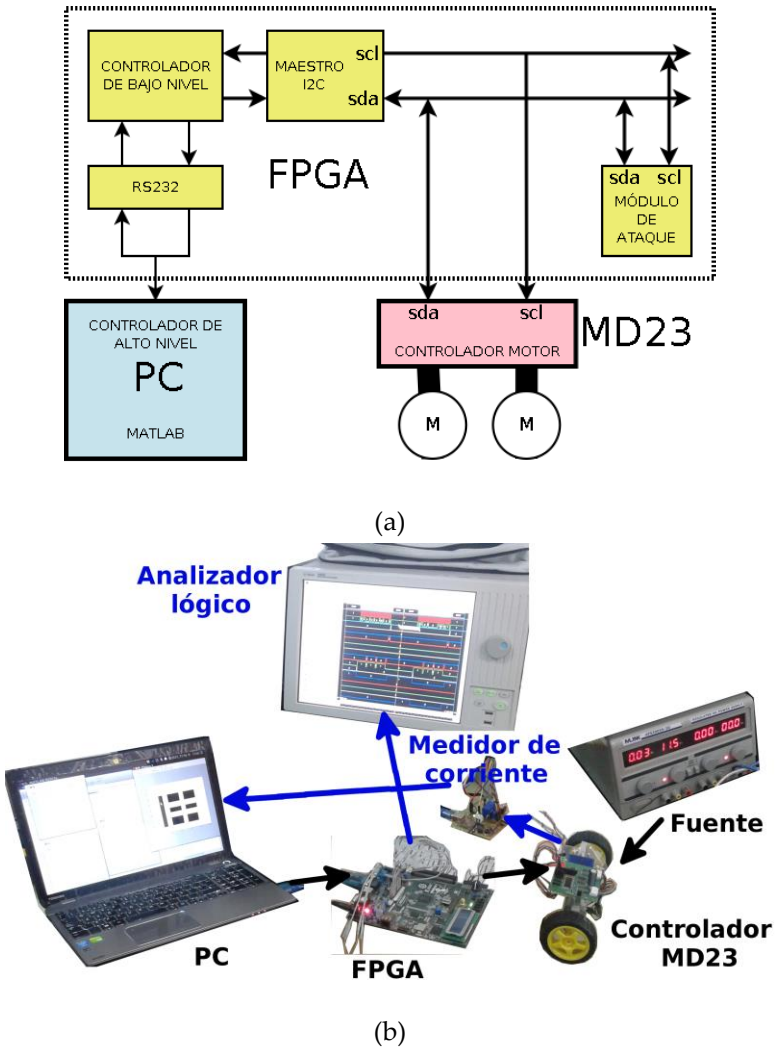


Figura 4.2 (a) Esquema de la plataforma experimental. (b) Fotografía de la plataforma experimental.

La principal misión de la plataforma es la verificación y monitorización del proceso de navegación. Luego, ha sido necesario para tal fin la inclusión de los elementos necesarios para la monitorización y medición de dicho proceso. La monitorización de las transiciones lógicas ha sido fundamental para establecer el procedimiento de inserción de fallos y validar el correcto funcionamiento de los módulos diseñados. Asimismo, ha sido importante el estudio del consumo del sistema durante la inserción de fallos, con el fin de confirmar si dicho procedimiento afectaba o no, de forma nociva, al funcionamiento interno del robot. Por todo ello, los elementos de medida y verificación utilizados han sido: un analizador lógico para monitorizar el bus I2C y las principales señales generadas en el dispositivo FPGA y un sistema medidor de corriente que permite estimar la potencia consumida por los motores.

Dada la cantidad y el volumen de la instrumentación incluida en el sistema, cabe destacar que no se consideró conveniente la implementación de ésta sobre una plataforma móvil. De hecho, como podrá comprobarse en la sección de experimentación, la realización de los experimentos en movimiento real habría puesto en peligro la integridad del sistema. Por este motivo, las ruedas de la plataforma no han reposado en el suelo y la plataforma ha permanecido estática. No obstante, el movimiento se ha estimado aplicando técnicas de odometría, a partir de las medidas de los encoders. Esta situación no invalida para nada las conclusiones alcanzadas, pues lo que se muestra en este trabajo son las situaciones de vulnerabilidad y los puntos débiles del hardware de control utilizado, por lo que no ha sido necesaria la ejecución real del movimiento de la plataforma.

4.2.2 Controlador de alto nivel

Dado que el objetivo fundamental de estudio ha sido la vulnerabilidad del hardware de bajo nivel, se ha elegido una aplicación que, utilizando la información procedente de dicho hardware, efectuará una tarea de navegación y a la vez permitiera identificar claramente los efectos de las vulnerabilidades detectadas. Así, se han descartado aplicaciones elaboradas donde la mayor complejidad de procesado podría haber dificultado la interpretación de los resultados. Concretamente, para el control de alto nivel se ha utilizado el algoritmo de persecución pura (*pure pursuit*) [Oll95],[Gom01]. Se trata de un algoritmo de seguimiento de caminos ("*path following*") cuyo objetivo es asegurar que el robot sigue un camino predeterminado con la precisión adecuada. La elección de este algoritmo no restringe los resultados obtenidos en este trabajo. Conclusiones similares podrían haberse alcanzado con la elección de otras aplicaciones basadas en navegación reactiva, topológica, SLAM, etc. si bien habría sido necesario utilizar un entramado sensorial más complejo que no habría aportado nada a la problemática vinculada al control de bajo nivel.

El algoritmo "*pure pursuit*" ha sido programado en un PC sobre una aplicación que se ejecuta en Matlab. En ella se integran tanto el cálculo de referencias de control como la gestión de la interacción con la FPGA. En la Figura 4.3 se muestra el diagrama de flujo que ilustra el funcionamiento de dicha aplicación, la cual se ejecuta en paralelo con las rutinas propias de la FPGA.

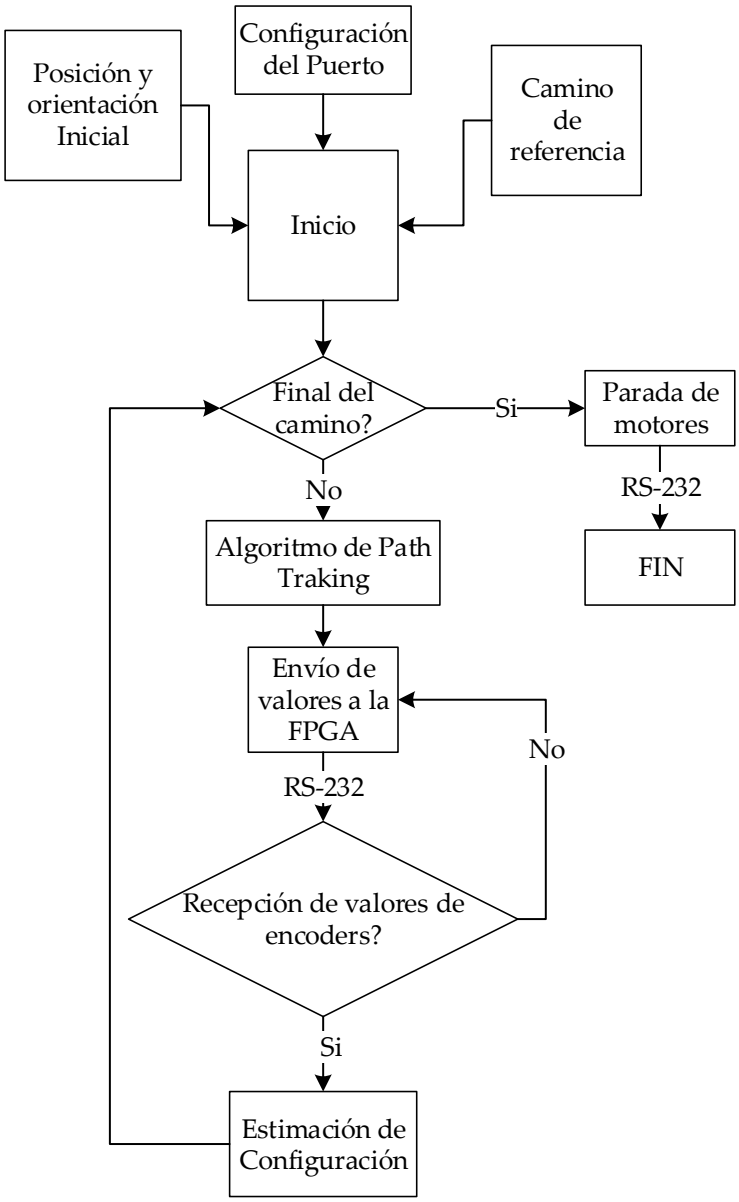


Figura 4.3 Diagrama de flujo que describe el funcionamiento del programa que corre en el PC.

El programa que hace de controlador de alto nivel y ejecutado en Matlab realiza las siguientes tareas. En primer lugar, el controlador de alto nivel inicializa la operación del sistema. Dicha inicialización conlleva la inicialización de la comunicación serie por el puerto RS-232, la determinación de la trayectoria de referencia que debe seguir el robot, y la posición y orientación inicial del mismo (la exactitud de estos datos no debe tener excesiva relevancia ya que serán corregidos cuando el robot envíe los datos reales). Una vez

inicializado el controlador, éste entrará en un bucle del cual sólo saldrá cuando finalice la trayectoria. El bucle comienza con el cálculo de las velocidades de los motores para alcanzar el siguiente punto de la trayectoria. Dicho cálculo será realizado por un algoritmo de seguimiento de camino, y son enviadas (vía RS-232) a la placa FPGA para su posterior envío a los motores. A continuación se leen los encoders de los motores para poder determinar la nueva posición del robot. Esta nueva posición es determinada a través de un filtro de Kalman extendido (EKF) como pre-procesamiento previo a técnicas de odometría. Finalmente, el bucle se vuelve a ejecutar si no se ha alcanzado el final de la trayectoria. En el caso de alcanzar el final de la trayectoria, el controlador envía una orden de parada a los motores para detener el robot.

Aunque en esta plataforma, el posicionamiento del robot se ha obtenido únicamente mediante técnicas de odometría, la inclusión de otros tipos de técnicas de posicionamiento (como la utilización de GPS) no alteraría las conclusiones alcanzadas, porque las anomalías estudiadas no afectan a este procedimiento.

4.2.3 Implementación en FPGA

El concepto de "hardware configurable", expresado anteriormente, puede materializarse de muchas maneras. La más arcaica de todas sería la utilización de componentes discretos, lo que facilitaría el acceso a la monitorización y manipulación de todos los elementos que conforman la plataforma. Sin embargo, esta opción adolece de falta de flexibilidad, muy necesaria a la hora de solventar problemas o rediseñar la arquitectura del sistema. Por este motivo, se optó por la utilización de una FPGA. Gracias a ella, es posible tener un mayor control sobre las señales más críticas de la plataforma aprovechando la flexibilidad que dan los sistemas hardware programable [Pri07]. La placa de

desarrollo utilizada es la placa Spartan 3AN Starter Kit Board [Cha06], que entraría en la categoría de dispositivos de bajo coste. Los módulos desarrollados en dicho dispositivo son:

- Una unidad UART basada en el protocolo RS-232 para comunicar la FPGA con el PC.

- Un controlador de bajo nivel. Este módulo está compuesto por: un controlador de tipo PID para cada rueda; el interfaz que transforma los datos interpretables por el maestro I2C y viceversa; y el estimador de velocidades a partir de los valores leídos de los encoders.

- Un maestro I2C, para controlar la comunicación a través del protocolo I2C.

- Un esclavo I2C, para comprobar el mecanismo de transmisión dentro de la FPGA.

- Un módulo de inserción de fallos, para implementar las anomalías consideradas en las comunicaciones I2C.

La utilización de un dispositivo FPGA como plataforma de implementación ha motivado el uso de un lenguaje de descripción de hardware de alto nivel, concretamente VHDL [Ash90], para el diseño de los diferentes bloques y su conexionado. La metodología de diseño utilizada para los diferentes bloques ha sido su implementación en una máquina de estados algorítmica [Bro81]. En esta metodología, la implementación se divide en dos grandes bloques: un procesador, que realiza las operaciones del sistema; y un controlador, que realiza la secuencia de las operaciones. De esta forma, la

implementación dispone de señales cuyo único significado son datos, y otras cuyo único significado son fases de operación.

A continuación se describen las características de los elementos más significativos dentro de la FPGA: controlador de bajo nivel; módulo de inserción de fallos y el sistema de monitorización de señales. La unidad UART, y los módulos maestros y esclavo han sido implementados siguiendo los estándares tradicionales definidos en la norma RS-232 y en la del bus I2C.

4.2.3.1 Controlador de bajo nivel

La Figura 4.4 se muestra un diagrama funcional del controlador de bajo nivel, mientras que en Figura 4.5 se ilustra el funcionamiento secuencial del sistema.

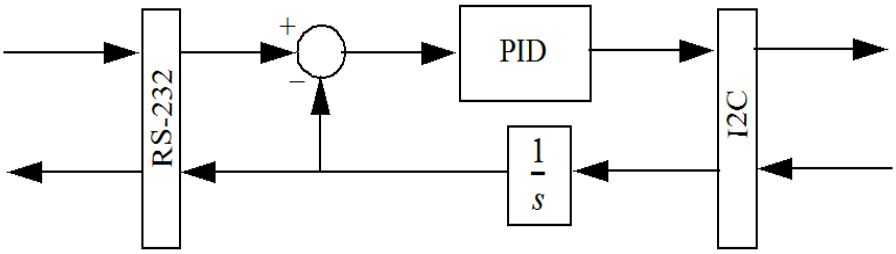


Figura 4.4 Esquema funcional del controlador de bajo nivel.

Tras la inicialización comienza un bucle sin fin. Dentro del mismo, en primer lugar, se atiende al buffer del puerto serie, con el fin de conocer los valores de referencia para las velocidades de las ruedas. Si no hay datos en el buffer se mantienen las referencias anteriores, si los hay, las referencias se actualizan. Seguidamente, el módulo PID calcula el error entre los valores deseados y los medidos con anterioridad y determina los valores de la acción de control a ejecutar sobre las ruedas. Estos valores se envían a través del bus I2C. A continuación, se solicita al esclavo la lectura de los valores leídos por los

encoders. Merece la pena resaltar que los encoders proporcionan información incremental, es decir, informan de número de pulsos leídos desde la última vez que fue solicitada la lectura. Los valores de las velocidades angulares de los motores se calculan dividiendo el número de pulsos por el tiempo transcurrido desde la última medida. De ello se encarga el mismo módulo del control de bajo nivel, que posee un contador de tiempo que determina el lapso temporal entre dos medidas consecutivas. Los valores de las velocidades así estimados son enviados al PC a través del puerto RS-232, a la vez que serán utilizados por el PID para establecer futuras acciones de control.

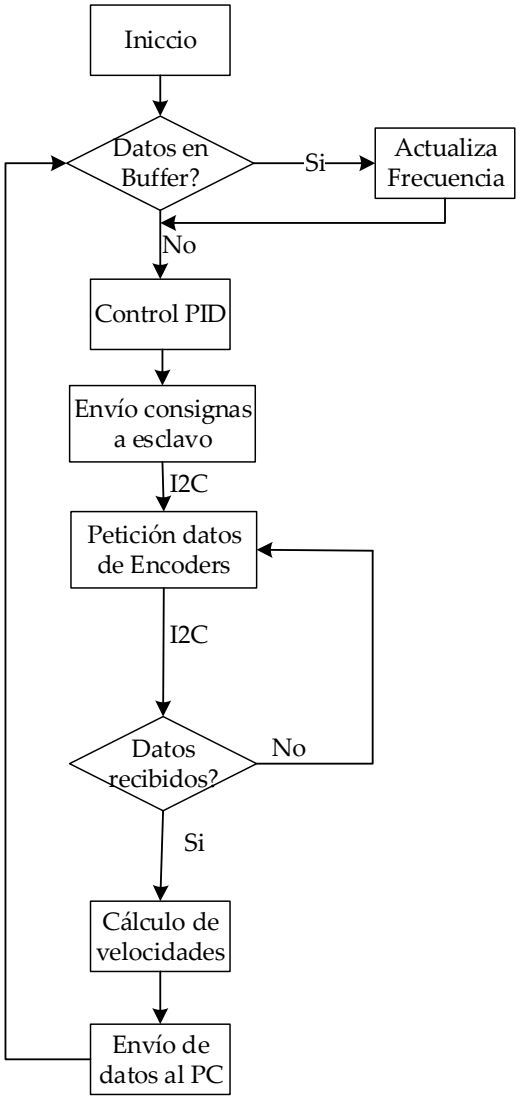


Figura 4.5 Diagrama de flujo que describe el controlador a bajo nivel.

4.2.3.2 Módulo de inserción de fallos

El módulo de inserción de fallos se ha diseñado para provocar de forma controlada y reversible un mal funcionamiento en el sistema de bajo nivel. Se trata de un módulo reconfigurable, para permitir probar en el futuro distintas anomalías que pudieran desvelar más vulnerabilidades.

En la Figura 4.6 se muestra el diagrama de flujo que presenta el funcionamiento de este módulo, el cual ejecuta su tarea en paralelo con el resto de módulos que contiene la FPGA. Tras la inicialización de todos los componentes del bloque, la ejecución comienza con un bucle infinito, que mantiene el sistema a la espera de recibir la orden de ejecución de la perturbación. Una vez recibida la orden de ataque, se procede a configurar el mismo. Dicha configuración consiste en determinar el destino de la comunicación que se quiere perturbar, es decir, la dirección del esclavo y la del registro dentro de dicho esclavo. La perturbación que se va a ejecutar consiste en la inhabilitación de un determinado proceso de comunicación.

Seguidamente, el sistema se mantiene escuchando y a la espera de detectar que se ha realizado una transmisión con el esclavo y el registro seleccionado. En ese momento, el módulo procede a insertar la perturbación.

Más concretamente la perturbación provocada consiste en transmitir a las líneas del bus I2C un estado de baja impedancia. Cuando la línea perturbada es la línea de sincronización, es decir, la línea SCL, el esclavo pierde la sincronía de la comunicación. Por lo tanto, el elemento esclavo, cuya comunicación ha sido perturbada, no es capaz de identificar ningún dato válido. En la Figura 4.7 se muestra un esquema funcional de este módulo. Dentro se presentan las etapas de salida de los buffer triestado que permiten la escritura. Las perturbaciones

así provocadas son totalmente reversibles, ya que, según la norma, las líneas del bus I2C trabajan en configuración *pull-up* y una baja impedancia no supone un cortocircuito en la línea.

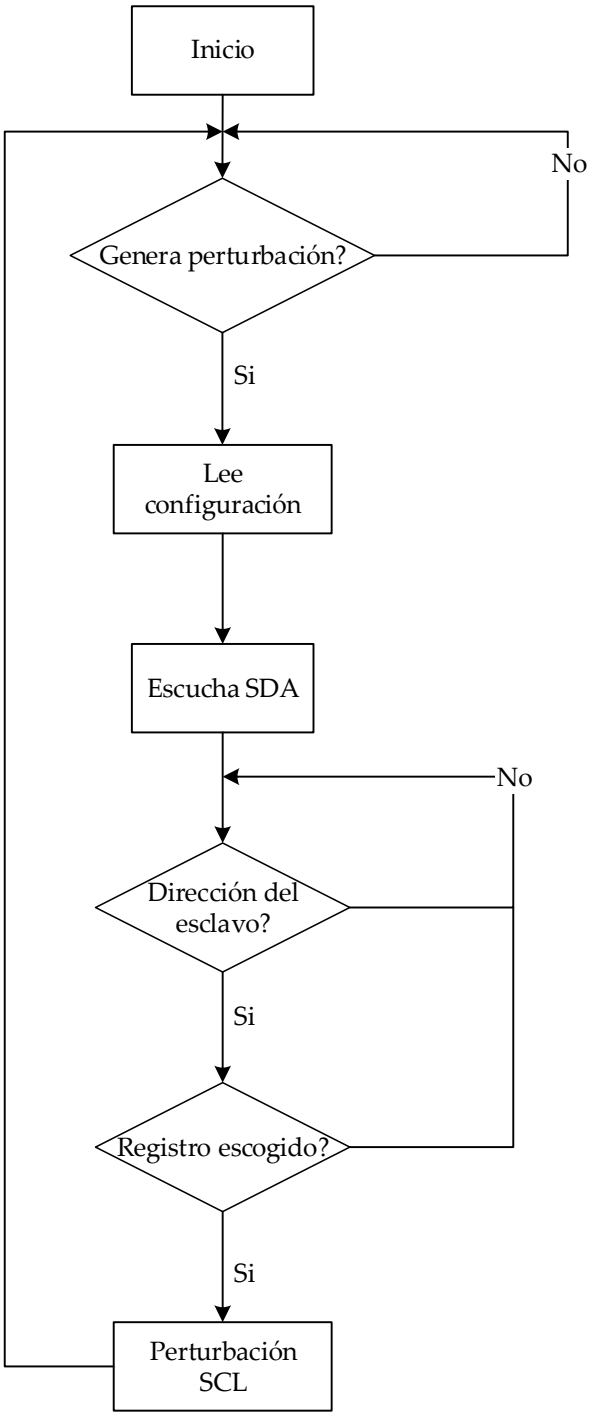


Figura 4.6 Diagrama de flujo que describe el módulo de inserción de fallos.

En definitiva, el módulo de inserción de fallos tiene un comportamiento parecido al de un módulo maestro, con la peculiaridad de que al escribir, fuerza en las líneas un nivel bajo durante el tiempo que dura la anomalía provocada.

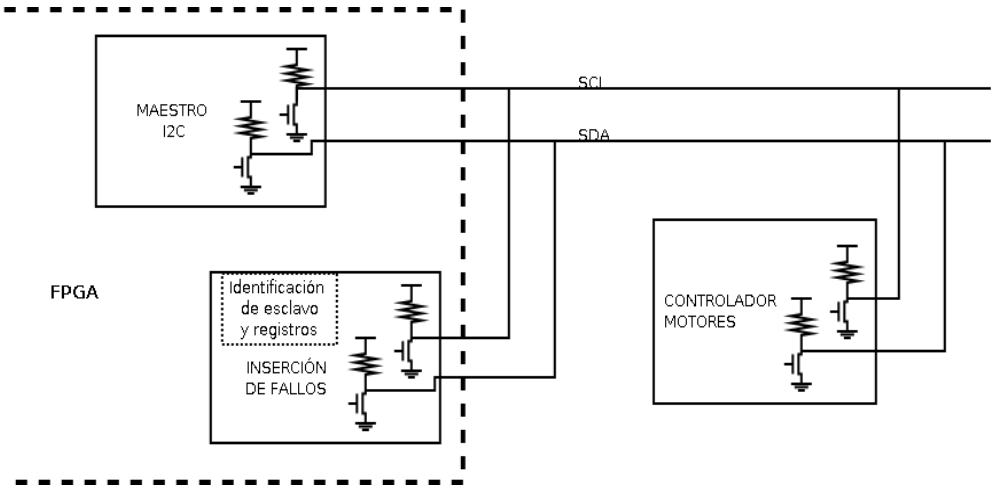


Figura 4.7 Esquema funcional del módulo de inserción de fallos.

4.2.3.3 Sistema de interconexión y monitorización de señales

En la Figura 4.8 se muestra como se interconecta la FPGA con el resto de elementos de la plataforma. En primer lugar, se distinguen las conexiones que permiten programar y alimentar la placa. En segundo lugar, destacadas en rojo, se señalan las utilizadas para la comunicación con el controlador de alto nivel, y con el esclavo I2C estándar. Finalmente, las zonas destacadas en azul se identifican las conexiones y elementos utilizados para la monitorización de señales.

En este sentido, uno de los principales objetivos de la plataforma es facilitar la monitorización de las señales involucradas en el proceso de control, con el fin de verificar el comportamiento obtenido. Para ello, se han utilizado dos tipos diferentes de monitorización: las señales cuyo cambio es realizado a alta frecuencia (señales de comunicación entre el dispositivo FPGA y el resto de componentes, que visualizarán utilizando el analizador lógico); las señales cuyo

cambio es realizado a baja frecuencia (los estados de configuración del sistema, que se visualizarán utilizando LEDs).

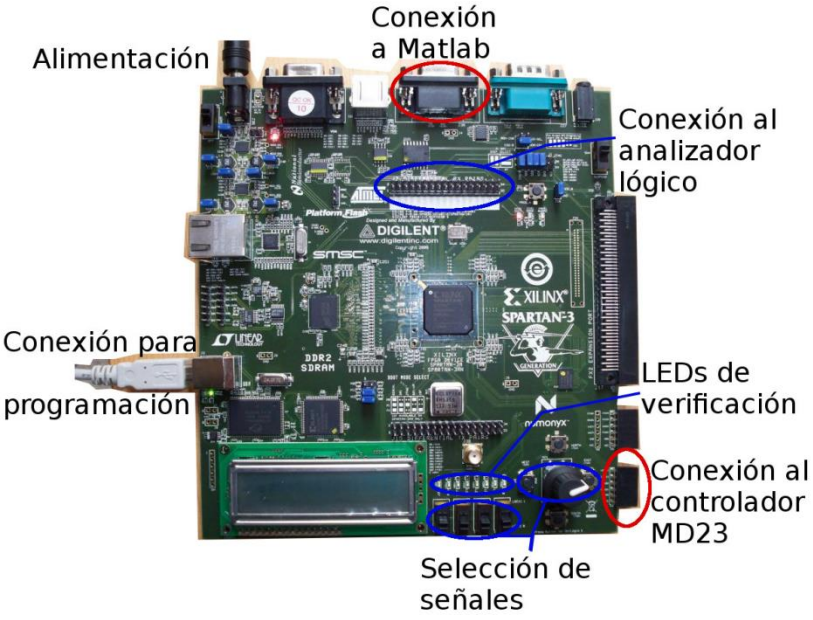


Figura 4.8 Conexión de la placa de desarrollo con los elementos restantes de la plataforma de experimentación.

Las fases de operación consideradas han conllevado la monitorización de los grupos de señales de alta frecuencia, como se observa en la Tabla 4.1.

Mux.	Señal de disparo
X"F"	Señal de finalización de recepción de RS-232: rx_done
X"C"	Orden de ataque transmitido por RS-232: estado_ataque
X"A"	Estado de captura de encoder de la comunicación RS-232: estado_esp_valor
X"9"	Estado de finalización del proceso de transmisión RS-232: estado_fin
X"0"	Estado de captura de dato de la comunicación RS-232: estado_captura_dato

Tabla 4.1 Monitorización de fases de operación en el analizador lógico.

que se corresponden con:

- Inserción de vulnerabilidades. El módulo de inserción de fallos va a provocar una perturbación sobre la señal de sincronización del

protocolo I2C, por lo que las señales que se van a monitorizar son las del protocolo I2C (tanto las generadas por el maestro como la señal afectada), las órdenes de ataque (tanto para atacar a la señal de sincronización como la generación de los reconocimientos), y los estados de dicho módulo.

- Adaptación de la comunicación RS-232 al protocolo I2C. Existe una conversión de protocolos desde el RS-232 (que se utiliza para la comunicación entre el controlador de alto nivel y bajo nivel) al I2C (que se utiliza para la comunicación entre el controlador de bajo nivel y los actuadores). Por lo que las señales que se van a monitorizar la señal de finalización de la transmisión serie, los datos independientes (que vienen de la transmisión RS-232) y los estados de dicho módulo.

- Comportamiento del maestro del protocolo I2C. Este protocolo será utilizado para la comunicación con un módulo estándar, en concreto unos motores. Las señales que serán monitorizadas son las señales del protocolo I2C y el estado de dicho módulo.

- Comportamiento de la comunicación RS-232 con el PC. Debido a que el número de señales involucradas es muy elevado, la monitorización de dicho comportamiento se ha dividido en dos. Las señales a monitorizar en este caso son las señales típicas del protocolo I2C, y del protocolo R-232 (incluyendo la sincronización tanto de la transmisión como de la recepción), así como la dirección del esclavo con el que se quiere establecer la comunicación.

- Comportamiento del protocolo I2C. En este caso, se van a monitorizar las líneas del bus, así como todas las órdenes que deben ser traducidas a las líneas del bus (como son las órdenes de lectura/escritura, la orden de reinicio, ...).

Con las configuraciones anteriores puede ser monitorizada la operación de captura de encoders. Dicha operación no tiene su sección independiente debido a la gran cantidad de señales involucradas (cuatro paquetes por cada motor) y a que los valores no son fácilmente replicables para poder saltar de una configuración a otra y disponer de una visión de conjunto.

Además, la utilización del analizador lógico implica la generación de una señal de disparo ("trigger"). Dicha señal entra dentro de la categoría de alta frecuencia, y se utiliza para indicar al analizador el momento de comienzo de captura de datos, de forma que éste registre el número de eventos adecuados, monitorizando el comportamiento completo de las fases anteriores.

No obstante, una de las principales limitaciones de los dispositivos FPGAs es su limitado número de conexiones, el cual está aun más limitado en las placas de desarrollo. Como consecuencia de ello, no es posible monitorizar al mismo tiempo todas las señales enumeradas anteriormente. En consecuencia, se ha utilizado una estrategia de multiplexación mediante selección, para permitir el acceso a la monitorización de las señales según se hayan escogido. Esto se logra mediante un proceso de configuración que involucra el uso de dos pulsadores y cuatro conmutadores. Un primer pulsador se utiliza para activar el valor de la señal de disparo (mostrado en la Tabla 4.1). Un segundo pulsador se utiliza para activar la selección de las señales a monitorizar (mostrado en Tabla 4.2). Dicha selección se configura mediante los cuatro conmutadores, que conforman el valor de 4 bits. Este valor determina: el grupo de señales a

monitorizar y el momento de activación de la señal de disparo. Dicha configuración es monitorizada correspondientemente por los diodos LEDs.

Mux.	Operaciones a monitorizar	Señales
X"F"	Inserción de vulnerabilidades	scl_ataque, ack, scl_master, sda_master, ataque_replay, ataque_clk, estados_ataque
X"E"	Adaptación entre RS-232 e I2C	rx_done, transmisión, ack, inicio_repetir, espera_tray, estados_trayectoria
X"C"	Maestro I2C	scl_master, sda_master, done_master, inicio_repetir, estado_maestro
X"6"	Comunicación RS-232 I	rx_int, relj_rx, rx_done, scl_master, sda_master, ack, done_master, tx_int, reloj_tx, tx_done, error_ack, sda_in
X"5"	Comunicación RS-232 II	rx_int, reloj_rx, rx_done, dir(7:0), scl_master, sda_master
X"0"	Comunicación I2C	scl_master, sda_master, ack, done_master, error_ack, libre_master, transmisión, inicio_repetir, rw, ini_trayectoria, tx_int, tx_done

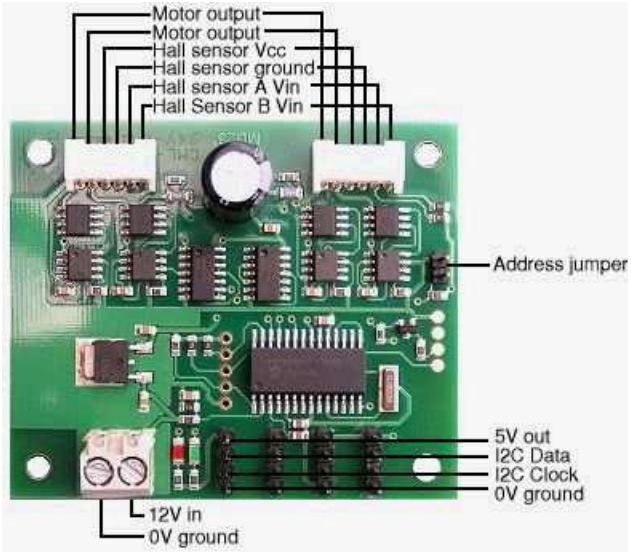
Tabla 4.2 Monitorización de fases de operación en el analizador lógico.

4.3 Esclavo I2C estándar y motores

En la Figura 4.9(a) se puede observar el controlador de motores MD23 utilizado en la plataforma experimental, que ha sido diseñado para interactuar con el modelo de motor EMG30 (Figura 4.9(b)). Dicho controlador dispone de una única dirección I2C pudiendo controlar hasta dos motores, correspondiente a cada una de las ruedas que posee la plataforma, izquierda y derecha, respectivamente.

La propiedad más destacable del motor EMG30 es que integra un sistema de encoders que proporcionan 306 cuentas por vuelta. Dichos sensores están unidos a una caja reductora de 30:1. Las especificaciones más interesantes de dicho motor son: alimentación 12 V; velocidad comprendida entre 1.5 rpm y 200

rpm: par máximo de 1.5 kg/cm; corriente sin carga 150 mA; corriente con carga 530 mA.



(a)

(b)

Figura 4.9 Control de bajo nivel. (a) Controlador MD23. (b) Motor EMG30.

Cabe destacar que la interacción desde el dispositivo FPGA hacia el controlador MD23 (y por tanto a los motores) se realiza mediante el acceso a registros internos del controlador. Concretamente, cada motor tiene asociado un registro, de tal forma que, para enviar una acción de control sobre un motor u otro se escribe en la dirección del registro correspondiente; igualmente se procede para la lectura de los encoders. Los registros con los que se ha interactuado en la experimentación son los indicados en Tabla 4.3.

Registro	Misión
0x00	Velocidad del motor 1
0x01	Velocidad del motor 2
0x02-0x05	Lectura de los cuatro bytes del encoder del motor 1
0x06-0x09	Lectura de los cuatro bytes del encoder del motor 2
0x10	Deshabilitar la parada a los dos segundos

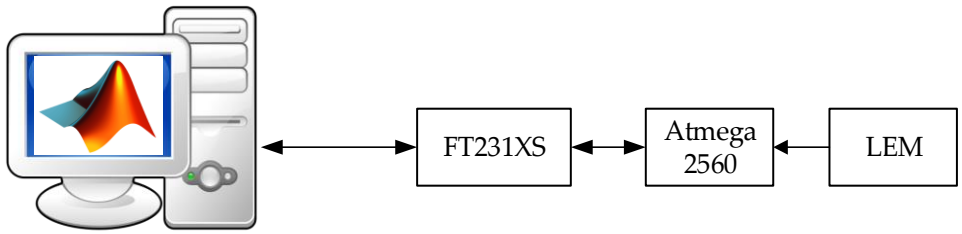
Tabla 4.3 Registros del controlador MD23 utilizados en la plataforma experimental.

4.4 Sistema de instrumentación

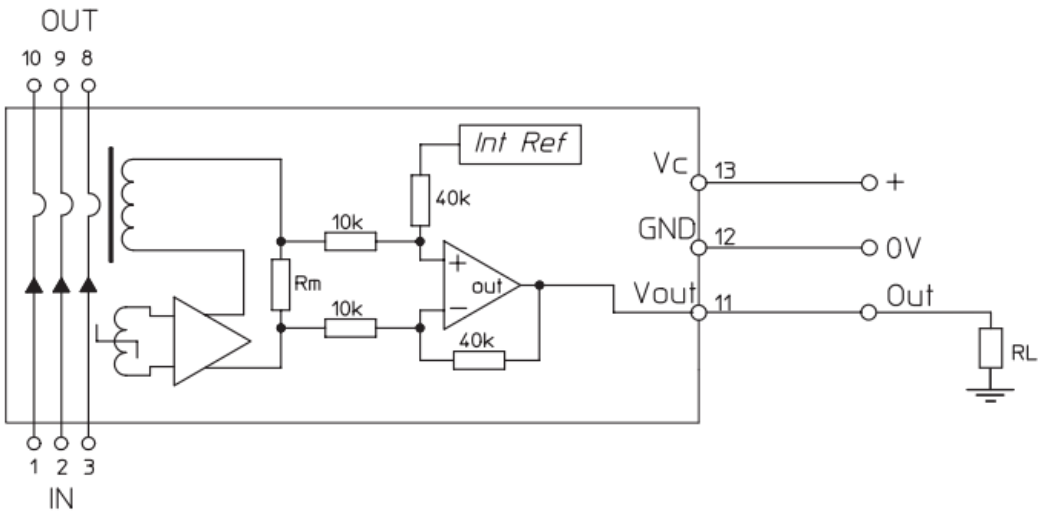
El sistema de instrumentación utilizado se puede dividir en dos grandes grupos: instrumentación digital y analógica. El bloque de instrumentación digital está compuesto por un analizador lógico y la electrónica que da acceso a las señales que deben ser monitorizadas. Este bloque ha sido explicado en las secciones 4.2.3.2 y 4.2.3.3.

El sistema de medición analógico se ha implementado con objeto de tomar medidas del consumo de ambos motores. Con ello se ha pretendido supervisar el efecto que cualquiera de los fallos insertados pudiera provocar en el desempeño del sistema de tracción. Aunque el controlador MD23 permite la medición de la potencia consumida, dicha medida no puede ser continua en el tiempo, puesto que requiere la lectura de los registros adecuados. Por tanto, se ha diseñado un sistema de instrumentación externo que captura las señales del consumo, siendo éstas registradas por el PC a la misma vez que se ejecuta el programa principal. Dicho sistema está compuesto por un microcontrolador ATmega2560, un transceiver (FT231XS) y un transductor de corriente (LEM CAS-6NP), como se puede observar en la Figura 4.10(a).

El transductor de corriente es un dispositivo electrónico que posee una entrada, que alimenta el primario de una bobina, y una salida en baja impedancia, que suministra una corriente proporcional y amplificada a la que circula por el primario. El esquema eléctrico del transductor se muestra en Figura 4.10(b). El primario de cada transductor se conecta en serie con el circuito de alimentación de cada motor, de forma que en cada experimento es posible registrar el consumo de ambos motores.



(a)



(b)

Figura 4.10 Instrumentación analógica: (a) diagrama de bloques; (b) esquema del transductor de corriente.

4.5 Caso de estudio: Vulnerabilidad del bus I2C

En esta sección se presenta el estudio realizado sobre la comunicación entre el sistema de control de bajo nivel y los motores, presentándola como ejemplo de uso de la plataforma para el estudio de una vulnerabilidad concreta.

Especialmente, con esta investigación se ha pretendido analizar los efectos que tiene la inyección de fallos en la señal de reloj que controla la comunicación en el bus I2C.

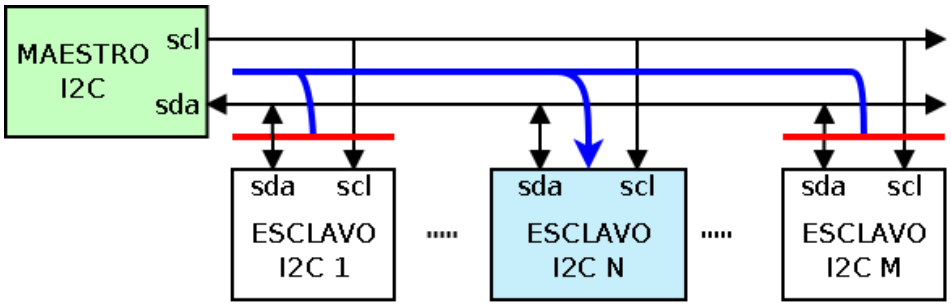
Esta interferencia puede aparecer en diversas fases del proceso de comunicación. De todos ellas, la transmisión de órdenes desde el maestro hacia el esclavo es la fase en la que, sin duda, este tipo de perturbación puede tener un efecto más negativo. Más concretamente, en la plataforma diseñada se han interferido los comandos enviados por parte del maestro I2C al esclavo controlador de motores. Con la idea de estudiar la aparición de fallos, tanto fortuitos como provocados intencionadamente, se han explorado varias posibilidades, interfiriendo la comunicación con un solo motor, con los dos motores, de forma temporal o de forma permanente.

Como se muestra en la sección 4.6, según sean las condiciones en que se producen estas perturbaciones, el curso del robot puede verse afectado en mayor o menor medida, poniendo en peligro la integridad de la aplicación robótica.

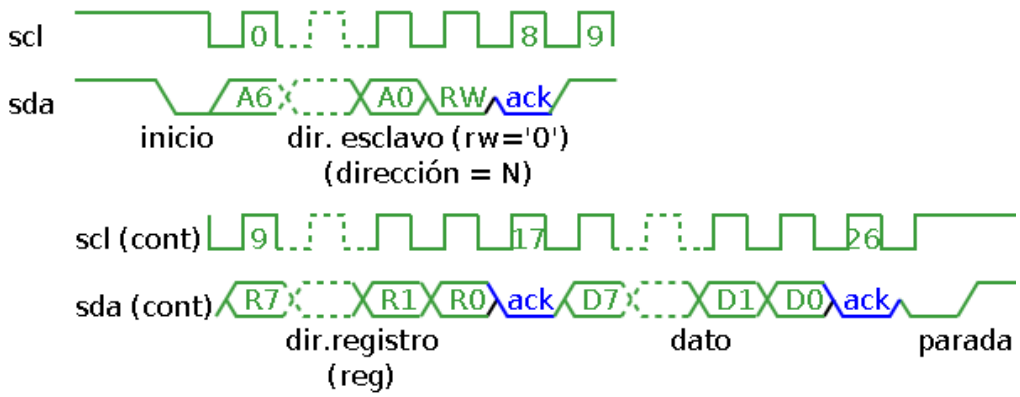
4.5.1 Vulnerabilidad del bus I2C

El protocolo I2C se basa en la transmisión de dos señales, generalmente llamadas SDA (línea de datos) y SCL (línea de reloj) y eventualmente una señal de tierra (todos los elementos deben tener la misma referencia de voltaje). Todos los módulos incluidos en el sistema de comunicaciones están conectados a las mismas líneas del bus, como se muestra en Figura 4.11(a). Existen dos tipos de módulos: un módulo principal o maestro que genera la señal SCL y controla las transmisiones; y uno o varios módulos esclavos que serán la fuente (operaciones de lectura) o destino (operaciones de escritura) de la información. El comportamiento de una operación de escritura usando un protocolo I2C se

ilustra en Figura 4.11(b) y está ampliamente descrito en la literatura científica y técnica [Ham13]. La señal SDA es generada tanto por el maestro como por los módulos esclavos cuando se produce la comunicación. Con el fin de que diferentes módulos sean capaces de manipular una misma señal sin problemas (colisión de información), los valores lógicos en el protocolo I2C son: tierra, para el nivel bajo (común a todos los módulos); y alta impedancia, para el nivel alto.



(a)



(b)

Figura 4.11 Bus I2C. (a) Arquitectura de un sistema basado en I2C. (b) Señales del procedimiento de escritura en el bus I2C.

La utilización de alta impedancia tiene una doble implicación. En primer lugar, en caso de colisión de información (ocurre cuando simultáneamente un nodo trata de escribir un '1' lógico y otro nodo un '0' lógico) resulta dominante el '0' lógico (evitando un valor lógico inespecificado o incluso fuera de los márgenes de ruido). En segundo lugar, los diferentes módulos no tienen que

utilizar la misma fuente de polarización y, por lo tanto, los módulos que necesitan alta tensión se pueden conectar al mismo bus que los módulos necesitados de baja tensión.

Desde el punto de vista del análisis de la vulnerabilidad, estas implicaciones representan un inconveniente manifiesto. Si una colisión de información (bien fortuita o intencionada) tiene lugar en la señal SCL, la comunicación con el controlador esclavo queda totalmente inhibida. Aun cuando ésta se detectase (por cualquiera de los controladores de alto o bajo nivel), sería imposible la comunicación, ya que éste no tendría referencia temporal para monitorizar los diferentes bits de la transmisión.

Con el fin de estudiar la repercusión de esta situación de fallo en el proceso de navegación del robot, se ha utilizado el módulo de inserción de fallos presentado en la sección 4.2.3.2. Mediante el mismo, se ha provocado, de forma controlada, una colisión de información en la línea SCL, en los momentos en que el maestro I2C se comunica con el esclavo para transmitir una acción de control sobre uno o ambos motores. Un fallo de inserción típico se puede apreciar en la Figura 4.12. La gráfica se encuentra dividida en tres sectores:

- Cronograma de estados del módulo de Inserción de fallos (fondo gris).

- Seña de inserción de fallos (fondo crema): CLK_Insert (activación de fallo en SCL).

- Señales del bus (fondo blanco): SDA y SCL MASTER (generadas por el maestro); SCL CONTROLLER (señal leída por el módulo MD23).

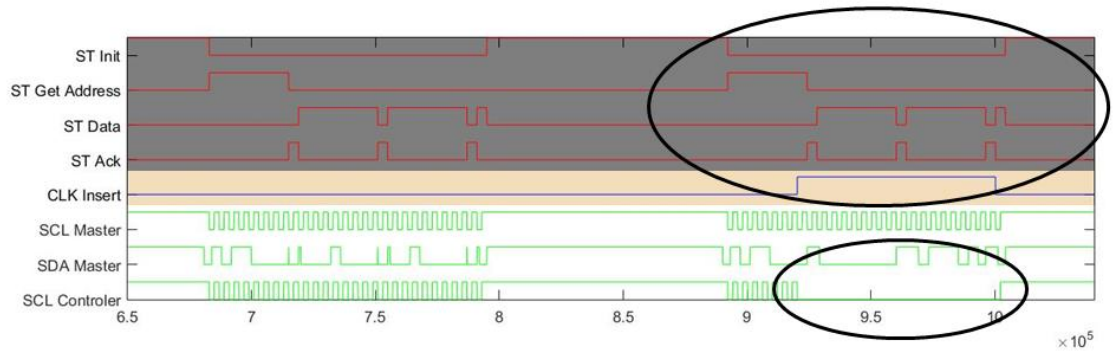


Figura 4.12 Gráficas de las señales obtenidas con el analizador lógico durante el estudio de la inserción de un fallo.

En un principio, y tras ser activado, el módulo de inserción se encuentra en un estado inicial (*St Init* a nivel alto) a la espera de la llegada de una comunicación. Cuando esto sucede, el módulo pasa al estado de lectura de la dirección del esclavo (*ST Get address*).

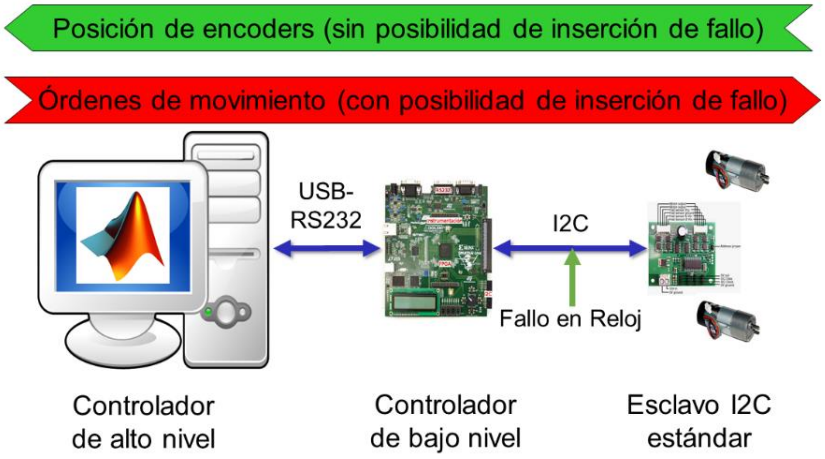
Una vez el maestro escribe dicha dirección (ver señal *SDA Master*) el módulo de inserción la lee, y evoluciona al estado de identificación de datos (*St Data*) y posteriormente al de identificación de reconocimiento (*St Ack*). Cuando llega el fin de la transmisión, el módulo vuelve al estado inicial (*St Init* vuelve a nivel alto). La inserción de fallo se producirá cuando la dirección del esclavo incluido en la transmisión sea la misma que la previamente seleccionada. Esta situación se observa en la zona marcada de la Figura 4.12, cuando la señal *CLK Insert* pasa a nivel alto. Puede observarse que en la primera transmisión no se ha activado el fallo, mientras que en la segunda sí. En la zona marcada inferiormente, se observa el comportamiento ante el fallo. El maestro genera las señales de forma adecuada (*SDA Master* y *SCL Master*), mientras que la señal *SCL* recibida por el esclavo (*SCL controller*) no es la adecuada ya que se mantiene a nivel bajo durante la mayor parte de la transmisión, instante en el que el maestro trata de transmitir el valor de la velocidad al motor.

4.5.2 Metodología experimental

Los mecanismos utilizados para realizar la verificación experimental de la plataforma ha sido la siguiente: A partir de un escenario particular se ha definido una trayectoria que el robot ha de seguir con precisión. Dicha trayectoria es entregada al programa de control de alto nivel. Este programa se ejecuta según lo detallado en la sección 4.2.2.

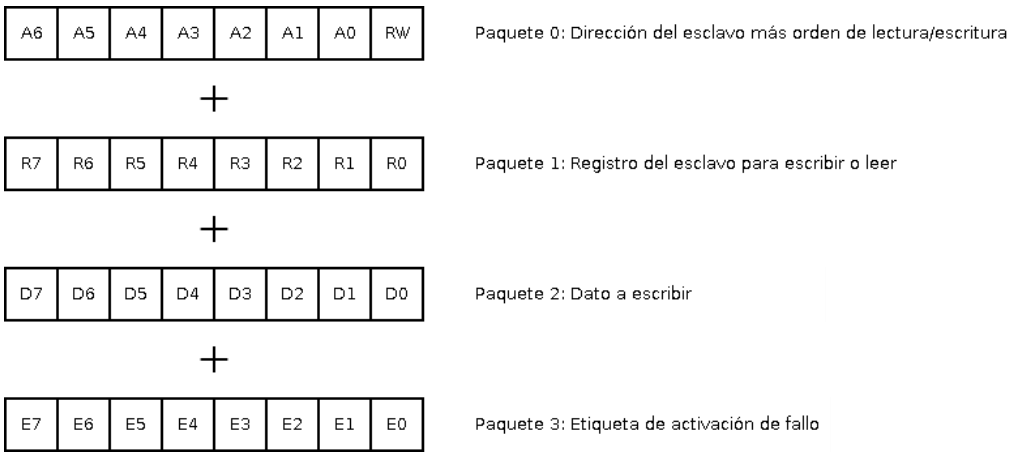
Como se observa en la Figura 4.13(a), el flujo de la comunicación va en dos sentidos diferentes. Las órdenes de escritura van desde el controlador de alto nivel hasta el esclavo (pasando por el controlador de bajo nivel) con el fin de controlar las velocidades de los motores, y de esta forma seguir la trayectoria requerida. Las órdenes de lectura van desde el esclavo hasta el controlador de alto nivel para conocer el número de giros realizados por cada rueda, y de esta forma poder determinar la posición del robot en cada momento utilizando técnicas de odometría.

Aunque la inserción de fallos se ha podido realizar en ambas transacciones (órdenes de escritura y de lectura), se ha optado por incluirla únicamente en las órdenes de escritura, es decir, cuando el controlador de alto nivel envía las velocidades a los motores; disponiendo siempre de las órdenes de lectura sin fallos. Esto ha sido necesario para poder observar por donde iría el robot ya que la plataforma no es móvil debido al sistema de instrumentación, como se comentó en la sección 4.2.1.



(a)

(b)



(c)

Figura 4.13 (a) Dirección de la información. (b) IDE implementado en Matlab para realizar la labor de ataque. (c) Datos enviados al bus de comunicaciones.

El mecanismo de ejecutar la orden de inserción de fallos ha sido incluido en el PC a través de una simple interfaz mostrada en la Figura 4.13(b). La operación básica de dicho interfaz consiste en incluir una etiqueta a la orden para activar o no el módulo de inserción de fallos. De esta forma, cuando la interfaz no ha sido accionada el módulo de inserción de fallos no es activado, por lo que permanece en un estado "durmiente" y el resto del sistema no se ve afectado por su operación. Cuando se acciona el interfaz, las órdenes de escritura son etiquetadas para ser atacadas activando el módulo de inserción de fallos. Eso implica una inserción de fallos tal y como se comentó en la sección 4.2.3.2. Por tanto, la comunicación real RS-232 desde el PC consta de cuatro paquetes de 8 bits, como se muestra en la Figura 4.13(c):

- El primer paquete indica la dirección del esclavo con el que se desea establecer la comunicación, así como su tipo, es decir, si se trata de una lectura o de una escritura.

- El segundo paquete indica el registro del esclavo con el que se desea realizar la comunicación.

- El tercer paquete indica el valor que se desea escribir. En el caso de operaciones de lectura, este campo no tiene ningún efecto.

- El cuarto paquete indica la etiqueta que va a tener dicha comunicación, a efectos de activar el módulo de inserción de fallos cuando sea necesario.

La inserción de fallos propuesta puede tener distintos orígenes. Bien puede ser provocada por una causa fortuita (sería el caso de una avería que provoque un cortocircuito en la conexión del esclavo a la línea *SCL*). O incluso puede ser generada de forma intencionada, con el fin de implementar un ataque

hardware sobre el robot. Esta situación se describe en [Gom15],[Gom16]. En el primero de los casos, la anomalía tendría muchas probabilidades de ser permanente, lo que contribuiría a una fácil detección de la misma. Por el contrario, en el segundo caso la anomalía sería de carácter temporal, lo cual dificultaría la caracterización del problema y además aparecería, con alta probabilidad, en aquellos momentos en los que la vulnerabilidad afectara con mayor grado a la fiabilidad de la aplicación robótica. Ambas posibilidades se exploran en la sección 4.6 de resultados experimentales.

En los experimentos propuestos, tras insertar el fallo, el controlador de alto nivel identifica que el robot no sigue la trayectoria correcta. Éste siempre tratará de corregir el comportamiento anómalo. No obstante, mientras dure la inserción estas correcciones nunca llegan a los motores. Un posible intento de corrección por parte del mismo MD23 está descartado, porque dicho elemento al no recibir señal de reloj, no identifica que se quieren comunicar con él. Por tanto, la detección del comportamiento anómalo, bien sea por parte del controlador de alto nivel o por otro módulo externo al esclavo, no permite contrarrestar las consecuencias negativas del fallo insertado. No parecen ser productivas las soluciones en ese sentido; por el contrario sí parece más apropiado el diseño de un esclavo, con cierta inteligencia, que sea capaz de detectar este tipo de fallo.

En la próxima sección se muestran las distintas situaciones en las que se ha procedido a insertar este tipo de fallo. Como se verá, la vulnerabilidad propuesta afecta con distintos grados de intensidad a la fiabilidad del robot, de manera que las conclusiones alcanzadas, van más allá de lo que una simple predicción pudiera prever.

4.6 Resultados experimentales

Se han realizado numerosos experimentos con varios escenarios. La idea ha sido caracterizar un conjunto de circunstancias en las que la inserción de fallos provoca en el robot un comportamiento no deseable, poniendo de manifiesto la envergadura de la vulnerabilidad estudiada.

Para mostrar los resultados obtenidos, se va a considerar el escenario mostrado en la Figura 4.14 compuesto por varios obstáculos. Del mismo modo, también se muestra la trayectoria planificada que debe recorrer el robot, que consiste en cuatro líneas rectas, dos curvas hacia la derecha y dos curvas hacia la izquierda. En dicha figura se presentan: la trayectoria de referencia (azul continuo), la trayectoria seguida por el robot (verde continuo) en el entorno definido por el mapa, así como el contorno del robot en distintos puntos del experimento. Como era de esperar, la trayectoria seguida prácticamente coincide con el camino de referencia.

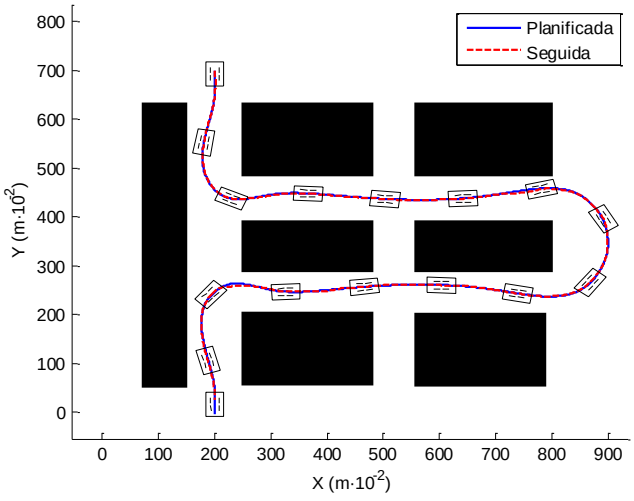


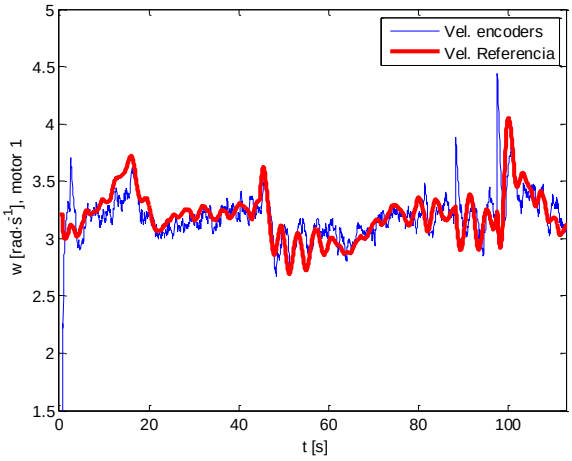
Figura 4.14 Trayectoria realizada por el robot sin inserción de fallos.

En la Figura 4.15 se ilustran para cada motor: las referencias de velocidad angular entregadas al controlador de bajo nivel (línea gruesa roja); la evolución de dichas velocidades medidas a partir de las señales de los encoders (línea fina

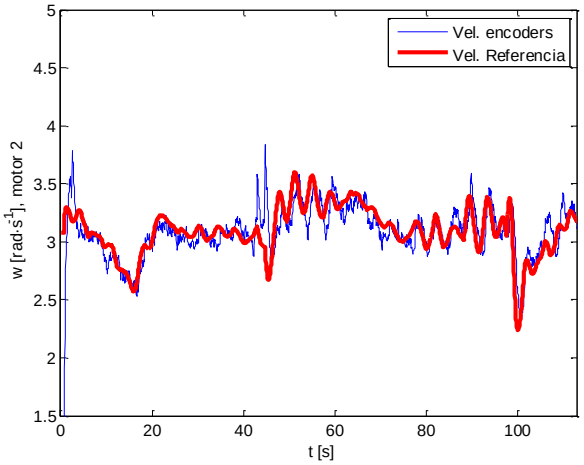
azul); el consumo de intensidad de cada motor. Nótese que, al no haber inserción de fallos, la evolución de la velocidad de las ruedas sigue con bastante precisión la referencia generada por el controlador de alto nivel.

Obviamente, la inserción de un fallo permanente en la señal de reloj durante el proceso de escritura en ambos motores, tiene una repercusión dramática sobre la integridad del robot. A continuación se mostrarán varias situaciones de inserción de fallos sobre el escenario y trayectoria mostrada en la Figura 4.14. En concreto se considerarán la inserción de fallos en una zona de la trayectoria donde el robot deba ir en línea recta, ya sea de forma permanente, o de forma temporal, o ya sea a un solo motor como a los dos motores.

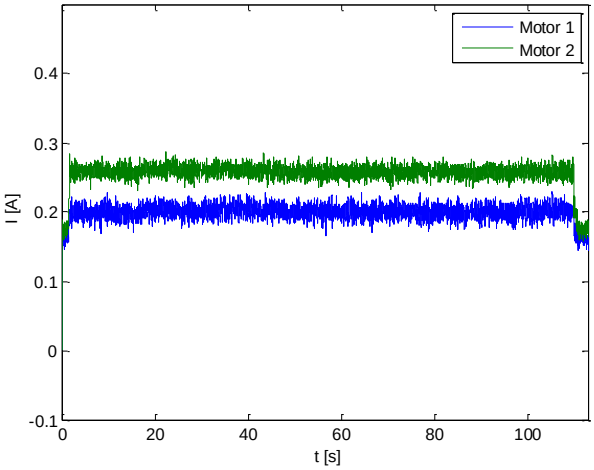
En primer lugar, se va a considerar la inserción de fallos de forma permanente en diferentes partes de la trayectoria donde el robot debería ir en línea recta. Como se muestra en la figura, el robot no sigue ninguna línea recta real, sino que se producen curvas para realizar una adaptación de salidas de curvas y entradas de curvas. Esta adaptación tiene la finalidad de aumentar el radio de las curvas y evitar curvas demasiado cerradas, debido a la problemática que implican. Por lo tanto, la falta de control sobre los motores, incluso en estos puntos, generará una nueva trayectoria que generalmente provocará una colisión con los obstáculos del escenario. Esta situación se muestra en la Figura 4.15, donde se han insertado fallos a la salida de la primera curva (Figura 4.15(a)) y a la entrada de la primera curva (Figura 4.15(b)). Como se puede observar, la colisión es inevitable por el carácter permanente de la inserción del fallo. El tiempo en que tarde en producirse (mayor o menor) dependerá de la curvatura que tuviese el robot en el momento de la inserción.



(a)



(b)

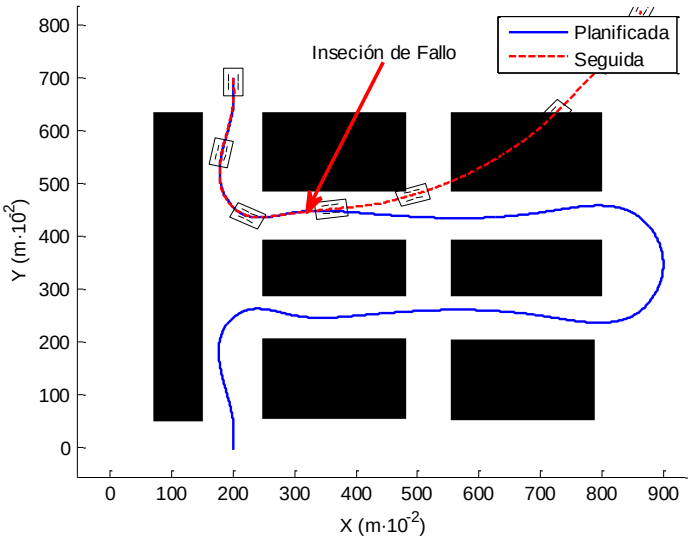


(b)

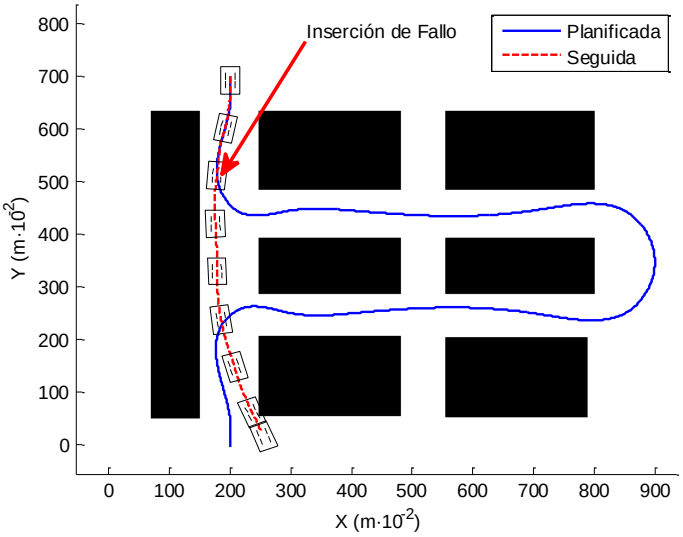
Figura 4.15 Experimento sin inserción de fallos. (a) Referencia y velocidad angular del motor 1. (b) Referencia y velocidad angular del motor 2. (c) Intensidades de ambos motores.

En la Figura 4.17 se representan la evolución de las velocidades medidas de los motores y sus referencias para el experimento de Figura 4.16(b).

Adicionalmente se incluye la evolución del estado del sistema de inserción de fallo (línea negra gruesa). Obsérvese que esta última señal se mantiene a bajo nivel hasta que, en un determinado instante, adquiere un nivel lógico alto (los valores asociados a los niveles lógicos se han elegido para facilitar la representación), activando el módulo de inserción de fallos.



(a)

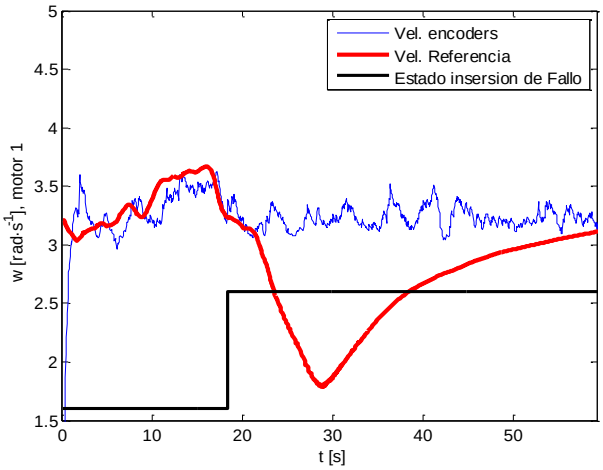


(b)

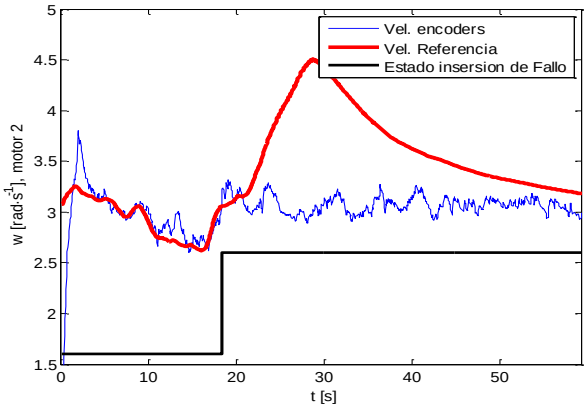
Figura 4.16 (a) y (b). Experimentos con fallos permanentes al escribir en ambos motores.

A partir del momento en que se produce la inserción del fallo, las Figura 4.17(a) y (b) muestran cómo ambos motores dejan de seguir la referencia

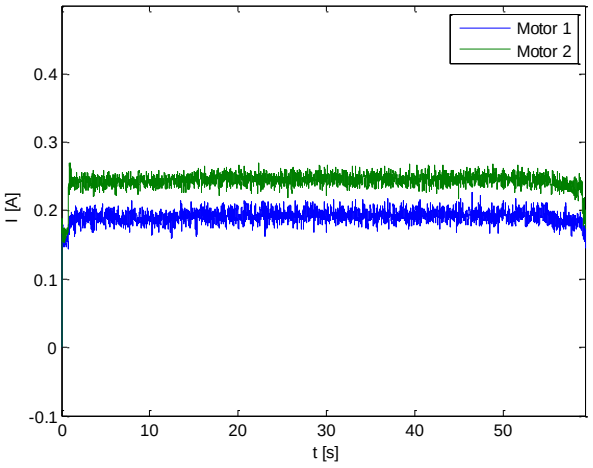
procedente del controlador de alto nivel, manteniendo aproximadamente el valor de velocidad que tenían en el momento de aparecer el fallo en la línea del reloj.



(a)



(b)



(c)

Figura 4.17 Experimento con inserción de fallo en ambos motores (a) velocidad, referencia e inserción de fallo en motor 1; (b) velocidad, referencia e inserción de fallo en motor 2; (c) intensidades consumidas por los motores.

Lo que ocurre, es que durante el tiempo en que tiene lugar la inserción, el esclavo del bus I2C mantiene la velocidad de los motores con los últimos valores que ha registrado, entendiendo que todo funciona correctamente ya que al no recibir señal de reloj, piensa que no hay comunicación en curso.

Otro dato importante a resaltar es que las intensidades consumidas por los motores (Figura 4.17(c)) no presentan ninguna anomalía, ni comportamiento diferente antes o después de la inserción del fallo. Lo que indica, que esta medida no puede ser utilizada para detectar el fallo estudiado en este trabajo.

Una situación diferente y muy interesante aparece cuando la inserción de fallo afecta sólo a un motor. Esta situación podría darse si cada motor estuviese controlado por un MD23 distinto o el módulo de inserción de fallos sólo se activase con la dirección del registro con el que se quiere comunicar (en lugar de con la dirección del esclavo), y una anomalía apareciera en la sección de la línea SCL que establezca la comunicación con un motor en particular.

La Figura 4.18 muestra la evolución del robot cuando se inserta un fallo en uno solo de los motores, en nuestro caso particular en el motor 2. En este caso de estudio, se observa que el robot sigue la trayectoria deseada, pero con una desviación mayor que en el caso sin fallos. Esto ha sido posible porque al controlador de alto nivel le llegan los datos reales de los encoders de ambos motores (sin incluir en las órdenes de lectura fallos). De esta forma, el controlador de alto nivel puede ajustar la velocidad del motor sin fallo (motor 1) tomando como referencia la velocidad del motor con fallo (motor 2).

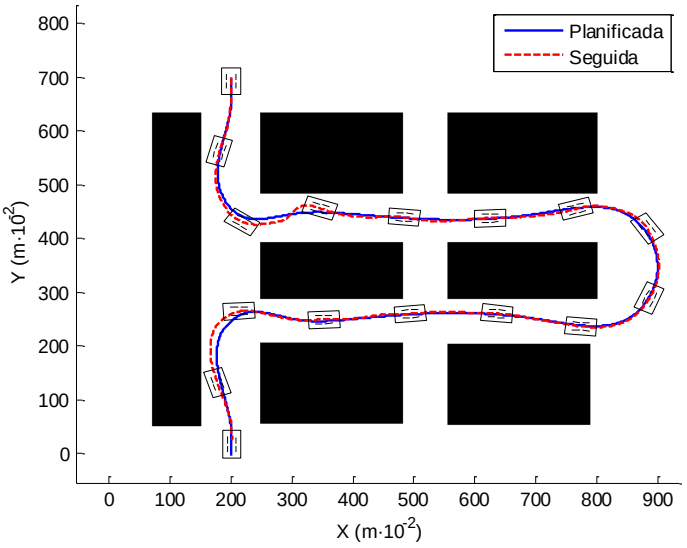
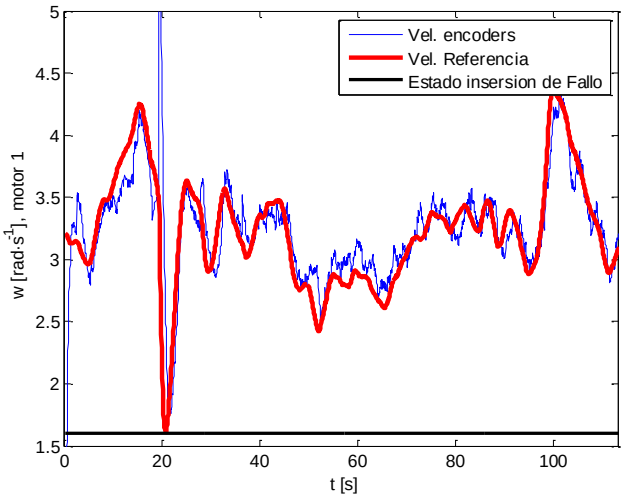


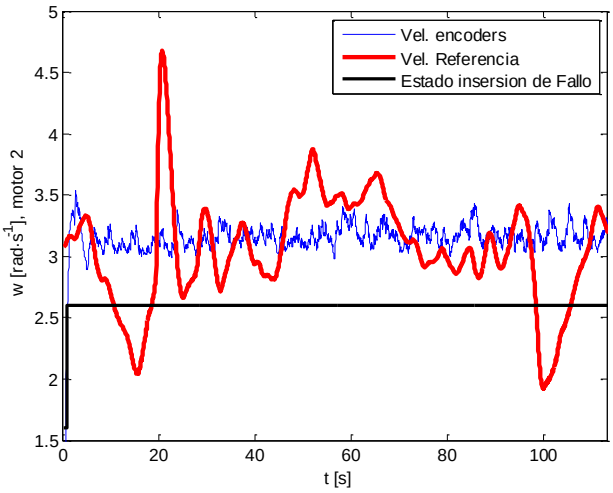
Figura 4.18 Trayectoria realizada por el robot insertando un fallo permanente en motor 2.

En la Figura 4.19 se puede observar cómo la velocidad del motor 1 evoluciona siguiendo su referencia y cómo el motor 2, que presenta un fallo insertado de forma permanente (desde el principio del experimento), mantiene un valor de velocidad aproximadamente constante, sin obedecer a las consignas de la referencia. Por lo tanto, el controlador de alto nivel puede ajustar la trayectoria con dos implicaciones básicas. En primer lugar, la desviación con respecto a la trayectoria planificada no es despreciable, pero no llega a ser lo suficiente como para que no se pueda seguir. En segundo lugar, el tiempo necesario para completar la trayectoria será diferente al caso sin inserción de fallos, puesto que las velocidades de los motores serán diferentes debido a la adaptación del motor1 al motor 2.

Estos resultados permiten concluir que disponer de un esclavo distinto para cada motor puede suponer una ventaja, ya que la aparición de fallos en solo uno de los motores no impide la ejecución de la navegación, aunque la precisión de la misma se devalúe.



(a)



(b)

Figura 4.19 Fallo en un solo motor: (a) Referencia y velocidad angular en motor 1 (sin fallo); (b) Referencia y velocidad angular motor 2 (con fallo).

La última situación que se va a considerar en este capítulo será la inserción de fallos sobre ambos motores, pero temporalmente. Esta situación podría ser consecuencia de un ataque hardware tal y como se comentó anteriormente. En este caso, la inserción dura unos breves instantes, y es efectuada en lugares estratégicos. Si la perturbación es efectuada con precisión, es posible modificar el curso del robot de forma que, sin causar daños en el mismo, y sin que aparentemente exista causa externa, el robot (por ejemplo) evite visitar una zona determinada. Para ilustrar esta situación se ha escogido el mismo

escenario que en los experimentos anteriores, y el ataque pretenderá que l robot no realice ninguna curva eludiendo su paso por toda la trayectoria original.

La Figura 4.20 muestra la trayectoria original (en color azul) y la seguida por el robot (en color rojo). Además, también se muestran los puntos en los que se ha activado y desactivado la inserción de fallos. Como se puede observar la trayectoria real es muy diferente de la trayectoria original, cumpliéndose el objetivo del ataque, es decir, que el robot entre en el escenario por la parte superior y salga por la parte inferior sin ejecutar ninguna de las curvas planificadas.

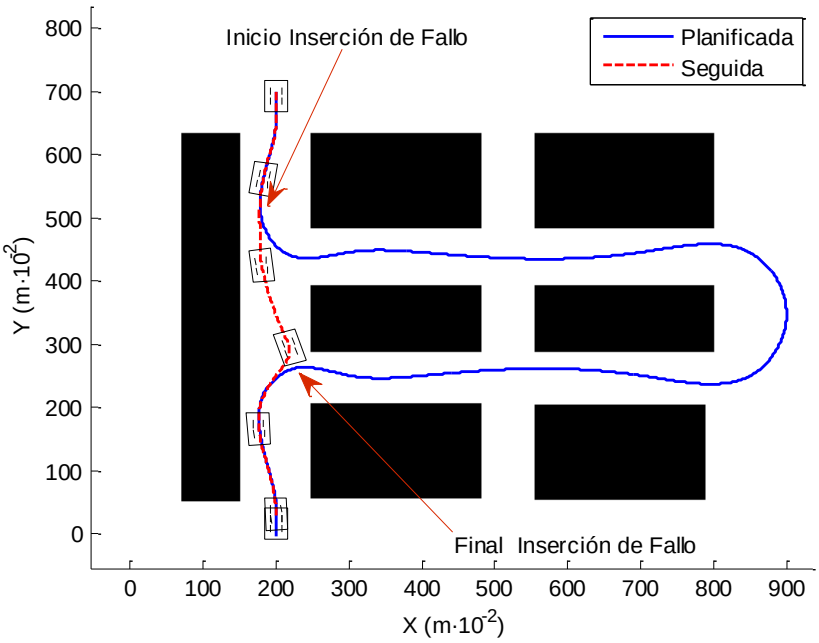
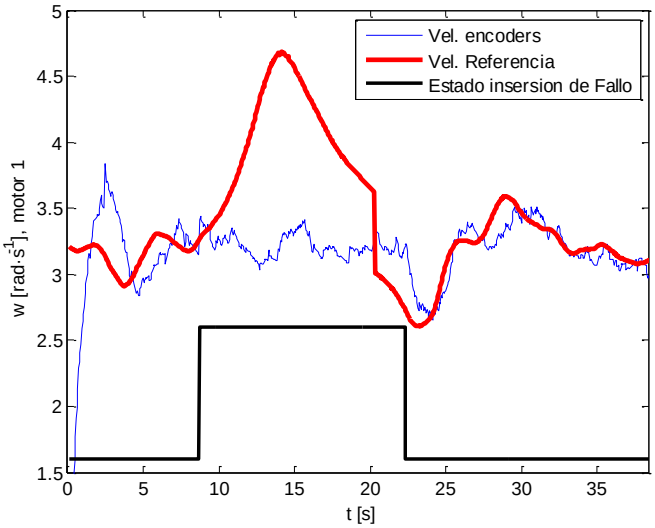


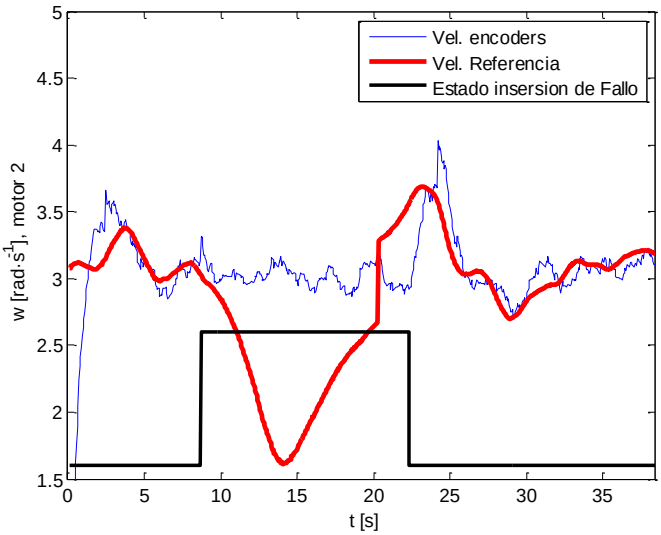
Figura 4.20 Trayectoria del robot con una inserción temporal selectiva.

En la Figura 4.21(a) y (b), se muestra cómo la señal de control generada para los dos motores (ambas gráficas en rojo) intenta obligar a la plataforma a cambiar el sentido de la marcha. Esto lo realiza el controlador de alto nivel bajando el valor de la velocidad deseada para el motor 2 y aumentando de forma considerable el valor de la velocidad deseada para el motor 1. Sin embargo, los motores hacen caso omiso. Posteriormente, cuando el fallo

insertado desaparece, el controlador de alto nivel es capaz de hacer que el robot converja hacia la sección del camino original que queda más próxima al vehículo. Es por este motivo que el robot se mueve hasta alcanzar la sección final del mismo. Si esta perturbación hubiese sido generada de forma intencionada, habría conseguido que el robot visitara una parte de la trayectoria de referencia. Obsérvese en las Figura 4.21(a) y (b), que a partir del cese de la inserción del fallo, las velocidades de los motores siguen con bastante precisión la señal de referencia.



(a)



(b)

Figura 4.21 Inserción temporal selectiva: (a) referencia y velocidad del motor 1 (con fallo temporal); (b) referencia y velocidad angular del motor 2 (con fallo temporal).

4.7 Conclusiones

En este capítulo se presenta con detalle el desarrollo de una plataforma para el estudio de la vulnerabilidad hardware y el comportamiento de los controladores de bajo nivel en el ámbito de la robótica móvil. La plataforma ha sido diseñada en base al concepto de hardware configurable, de tal forma que los elementos hardware que intervienen en el proceso pueden ser monitorizados, y, si es el caso, alterado su comportamiento durante el funcionamiento del robot, simulando de esta forma una situación de fallo. La plataforma dispone de un sistema de instrumentación que permite realizar múltiples medidas digitales y analógicas que posibilitan la caracterización de los comportamientos anómalos de los elementos vulnerados. Particularmente, se presenta un caso de estudio en el que se analiza la vulnerabilidad del bus I2C en situaciones en las que la señal de reloj se ve alterada. Los resultados experimentales alcanzados con la plataforma ponen de manifiesto la importancia de la vulnerabilidad estudiada. Un descubrimiento destacable consiste en la bondad de controlar cada motor con un esclavo distinto, de manera que las perturbaciones sufridas por uno puedan ser compensadas por el control realizado por el otro.

Capítulo 5. Sensor para detectar ataques hardware en aplicaciones robóticas

5.1 Introducción.

En los últimos años, el estudio de la vulnerabilidad en dispositivos electrónicos ha captado la atención de la comunidad científica [I10],[Bru05], [Kar13],[I10],[Bru05],[Kar13]. Se puede confirmar que entre los dispositivos electrónicos más usados están las plataformas robóticas. En la actualidad, los robots son usados para ejecutar multitud de tareas críticas tales como rescate, vigilancia, procesos industriales y otras operaciones cotidianas en general [Mar07], [Gar07], [Mor12] y [Par13],[Mar07],[Gar07],[Mor12],[Par13].

El control implementado en la plataforma robótica sigue una arquitectura tradicional, basada en PC, debido a esto, el estudio de seguridad en relación con la vulnerabilidad informática presenta un asunto relevante para el desarrollo de robots más seguros. Las acciones que buscan modificar el comportamiento del sistema se conoce como ataques, y la debilidad del sistema a ser explotado por dichas acciones es conocida como vulnerabilidad.

Muchos trabajos realizan investigaciones basándose en la vulnerabilidad software de las computadoras [Aro08],[Hee11],[Sin13],[Are06], otros en la

vulnerabilidad hardware [I10],[Bru05],[I10],[And96]. Los ataques para explotar las vulnerabilidades hardware son conocidos como ataques hardware.

El estudio de los efectos de un ataque software en sistemas robóticos es algo muy extendido. Sin embargo, los efectos producidos por un ataque hardware deben de considerarse de forma muy particular, teniendo en cuenta la cantidad de sensores y actuadores que hay en el campo de la robótica [I10], [Bru05], [Kar13],[And96],[Nob17].

El análisis de seguridad de un sistema es la búsqueda de vulnerabilidades para que el sistema tenga una respuesta adecuada. La respuesta puede ser dividida en dos partes. Cuando no hay una situación de ataque, el comportamiento del sistema debe ser el mismo para el que ha sido diseñado. Cuando hay una situación de ataque, el comportamiento debe ser tal que el objetivo del atacante no sea alcanzado. Este análisis da lugar a un flujo de diseño referente a la seguridad, mostrado en la Figura 5.1. En primer lugar, el sistema debe ser analizado para detectar vulnerabilidades, y cómo éstas pueden ser aprovechadas para atacar al sistema, lo cual implica el desarrollo de una serie de modelos de ataques al sistema. Una vez detectadas las vulnerabilidades y su forma de aprovechamiento, hay que desarrollar una serie de políticas de seguridad para eliminar las vulnerabilidades y/o evitar su aprovechamiento. Finalmente, dichas políticas deben ser implementadas en una serie de mecanismos.

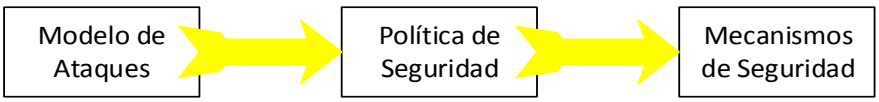


Figura 5.1 Porción del flujo de diseño referente a la seguridad de sistemas.

Uno de los modelos de ataques más utilizados es el que involucra a los sistemas de comunicaciones, ya que estos canales suelen estar ubicados en

zonas no seguras. Esta vulnerabilidad suele ser inherente al diseño, y su eliminación suele estar descartada. Los atacantes pueden hacer uso de esta vulnerabilidad de diferentes formas: monitorizando el contenido de las transmisiones, o suplantando la identidad del emisor o receptor. La política de seguridades más utilizadas en este campo es la encriptación de la información para que el contenido de la información transmitida no sea accesible. Esta política es implementada utilizando algoritmos de encriptación, ya sean de índole matemático [Sin13] o mediante sincronización [Are06]. Otras políticas utilizadas son la detección de sistemas externos atacantes, como los módulos troyanos [I10]; o la detección de mensajes fraudulentos [Bru05].

Este capítulo considerará la vulnerabilidad de un canal de comunicaciones no seguro. Como canal de comunicaciones se ha utilizado un canal I2C, que es muy utilizado por los sistemas robóticos para la comunicación entre el controlador central, y los actuadores y sensores del sistema. El estudio realizado seguirá el mismo flujo mostrado en la Figura 5.1. En primer lugar se estudiará una vulnerabilidad en el proceso de comunicación, que consiste en el libre acceso a la señal de reloj del canal de comunicación, y se mostrará como un atacante puede hacer uso de ella. En segundo lugar se discutirá la política y la implementación de un mecanismo de defensa para dar respuesta a un posible ataque.

El caso de estudio presentado está relacionado con la navegación autónoma de un robot móvil. Es decir, se supone que aplicando un algoritmo de seguimiento de trayectoria, el sistema debe ser capaz de visitar las áreas definidas por el usuario. En aplicaciones típicas de robótica móvil, se utiliza este tipo de estrategias [Mar07], [Gar07], [Mor09] y [Cue04], [Mar07], [Gar07], [Mor09], [Cue04]. En este contexto se estudia la posibilidad de atacar el proceso de comunicación I2C entre el controlador de alto nivel y el controlador de bajo

nivel, de esta forma se puede evaluar la defensa del sistema propuesto. Para realizar este estudio se hará uso de la plataforma descrita en el capítulo anterior, con la adición del mecanismo de defensa.

5.2 Señal de reloj: una posible fuente de ataques

La importancia del ataque a la señal de reloj, radica en la importancia que tiene el reloj en las operaciones del sistema. Más concretamente, la perturbación puede consistir en modificar la amplitud del pulso, la frecuencia o el periodo de la señal de reloj.

En la Figura 5.2 se observa una serie de ataques a la señal de reloj. En dicha figura, la señal "*CLK*" es la señal de reloj sin ataques (es decir con un periodo correcto), y la señal "*Ataque clk*" es la señal de reloj atacada (en estos casos se ha modificado la señal de reloj).

En Figura 5.2(a), se observa un ataque a la señal de reloj que aumenta de forma considerable el periodo. El resultado de este ataque consiste en la disminución de la velocidad de operación. Estas perturbaciones pueden ser usadas, para posibilitar la monitorización de señales transcendentales en el sistema, como por ejemplo en ingeniería inversa como se muestra en [McL08]. En la Figura 5.2(b), de tal forma que el periodo de la señal de reloj sea inferior al mínimo que garantiza la operación del sistema. Este decremento provoca un malfuncionamiento global del sistema, ya que no dispone del tiempo suficiente para realizar sus operaciones. En el caso de la Figura 5.2(c), se muestra un ataque que disminuye el periodo de reloj en una operación determinada. En este caso, el ataque consiste en evitar el funcionamiento de cierta operación. Si se considera un sistema microprocesador o microcontrolador, un ejemplo de estas operaciones críticas son las instrucciones de salto, ya que alterarían el

funcionamiento normal del programa. Este ataque es conocido de forma genérica como "*clock glitching*".

Éste último ataque se puede usar en circuitos encriptados que usan algoritmos de cifrados. El objetivo del ataque es evitar un ciclo completo del cifrado, lo que causa una vulnerabilidad en el circuito. Este hecho se puede lograr de forma independiente en la plataforma de implementación ya que todos requieren señal de reloj. Esta situación ha sido discutida en otros momentos [Dut11].

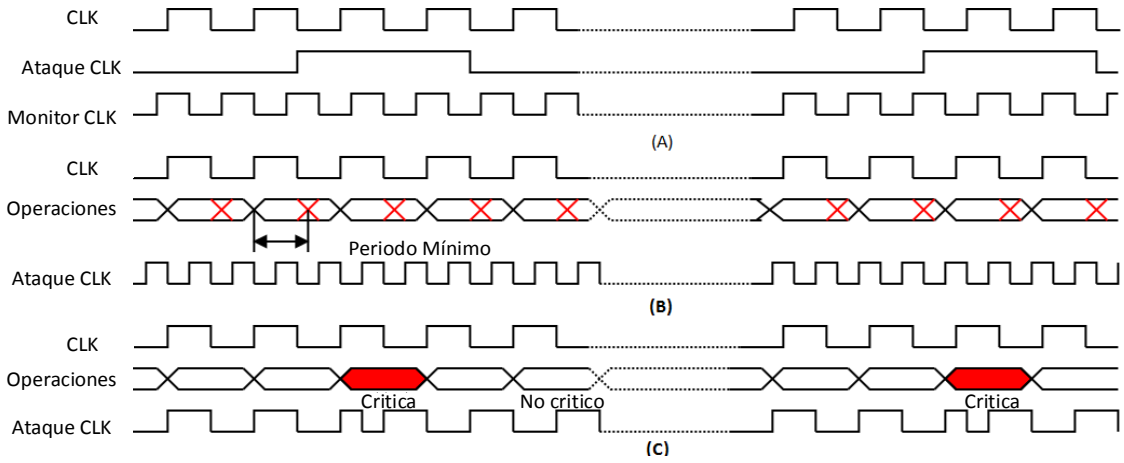


Figura 5.2 Ejemplo de un ataque a la señal de reloj. (a) Atacando incrementando el periodo de reloj. (b) Ataque disminuyendo el periodo de reloj. (c) Ataque hacia algunas instrucciones de reloj.

5.2.1 Ataque al protocolo I2C

La vulnerabilidad del protocolo de comunicaciones I2C no ha sido muy tratada en la bibliografía científica [Fuk04], [Alk06]. Este es el protocolo de comunicaciones más usado en plataformas robóticas para interconectar los sensores, motores y microprocesadores. Esta sección está dedicada al estudio de ataques "*clock glitching*" en el protocolo de comunicaciones I2C. Esto es debido a la facilidad de implementación, y la dificultad de detectar dichos ataques, por lo

que es importante estudiar el efecto y las soluciones de esta amenaza al protocolo de forma frecuente.

El protocolo usa dos líneas de señales: la señal *scl* que sincroniza la información y la señal *sda* que envía los datos de información. El proceso de comunicación involucra tres tipos distintos de datos: dirección del esclavo con un máximo de 128 esclavos (o 1024 esclavos para un direccionamiento extendido); dirección de registros, que indica el registro con el se va a interactuar; y valores en los registros, que indica el valor transmitido.

La capa física del protocolo es una de sus características principales e indica la correspondencia entre valores lógicos y valores de tensión. En este sentido, el nivel lógico alto equivale a una situación de alta impedancia en la línea del bus; mientras que el valor lógico bajo equivale a una conexión a tierra. De esta forma, cualquier módulo (ya sea maestro o esclavo) puede modificar el contenido de dichas líneas sin que exista colisión de información. Este hecho será el utilizado por el atacante puesto que el bus puede ser alterado tanto por módulos del sistema como por módulos externos al sistema (en este caso, un módulo de inserción de fallos).

En la Figura 5.3(a) se puede observar el comportamiento del protocolo I2C en una operación de escritura (en color negro se observa los datos transmitidos por parte del maestro, y en azul se identifican los datos transmitidos por el esclavo). La comunicación es controlada por el maestro del bus, es el encargado de iniciar la comunicación mediante la condición de "*inicio*" (la condición de *inicio* se identifica por una transición a nivel bajo de la señal *sda*, mientras la señal *scl* se mantiene en nivel alto). Seguidamente a la condición de inicio, se envía la dirección del esclavo (señales desde *A6-A0*), tras lo cual se introduce los valores *RW* (donde se identifican las señales de escritura mediante un nivel bajo

en la señal de datos). Seguidamente a las tramas proporcionadas por el maestro de la red, le sigue una trama de acuse de recibo o "*Acknowledgement*" por parte del esclavo. Esta operación es identificada por el nivel bajo de la señal *sda* en el pulso número 8 de la señal *scl*, indicando que el esclavo forma parte del proceso de comunicación. Una vez ha respondido el esclavo a los datos proporcionados por maestro en el bus, el siguiente paso por parte del maestro es, proporcionar los registros del esclavo a los que se quiere tener acceso (señales desde *R7-R0*), la respuesta de acuse de recibo por parte del esclavo, en este caso consiste en el pulso número 17 de la señal de reloj. Finalmente, el maestro envía el valor a escribir en el registro del esclavo, y éste último vuelve a enviar un *Ack*, pero en el pulso número 26 de la señal *scl*. Una vez finalizado el proceso de comunicaciones, el maestro envía una condición de "*stop*", esta condición se identifica por una transición de la señal *sda* a un nivel alto, mientras la señal *scl* se encuentra en un nivel alto.

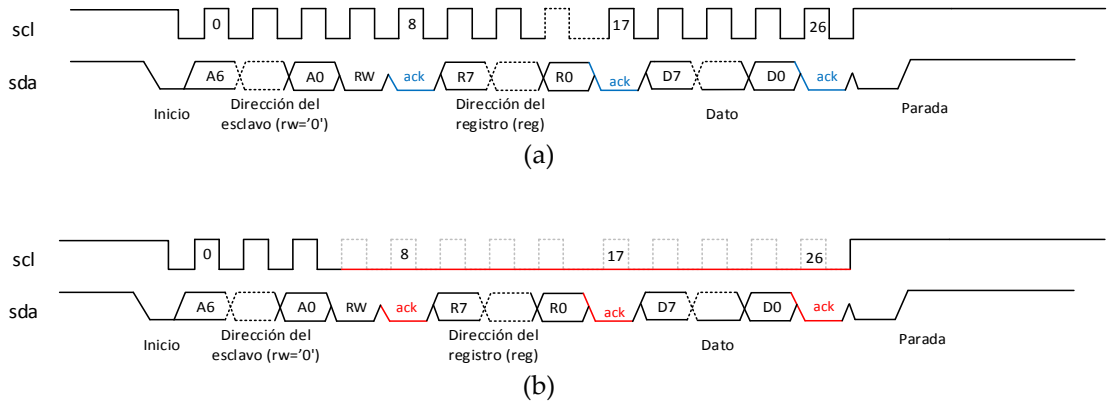


Figura 5.3 Comportamiento del protocolo I2C. (a) Operación de escritura. (b) Ataque a la operación de escritura.

Las características de este protocolo nos permiten realizar ataques focalizados, un ejemplo, atacar de forma particular la comunicación centrando estos ataques en algunos esclavos o en algunos registros específicos del esclavo. Los ataques involucran una variación en el periodo de la señal *scl* del protocolo. Un ejemplo de ataque hardware es el que se muestra en Figura 5.3(b), en color

negro se identifica el valor de la escritura de maestro en la red, mientras que en color rojo se identifica los valores de escritura en los ataques. En estos procesos, están involucrados tres módulos: el maestro del bus, el esclavo y finalmente el módulo de ataque.

Cuando se inicia un proceso de comunicación, se realiza con la transmisión de dirección del esclavo con el que se quiere comunicar. En ese momento, el módulo que realiza el ataque, monitoriza lo que sucede en el bus de comunicaciones, para realizar el ataque cuando lo crea oportuno. El ataque se realiza sobre la señal de reloj, no sobre la señal *sda*. En el momento que se observe un dato susceptible a ser atacado, la señal *scl* se coloca a nivel bajo. En el caso mostrado, el ataque es realizado a un esclavo en particular, y a cualquier registro del mismo. Por lo tanto, cuando la dirección del esclavo es completada (con la llegada de la señal *A0* en la línea *sda*), se activa el ataque colocando la línea *scl* a nivel bajo. Como el esclavo no detecta la finalización de la trama, no envía la señal de reconocimiento (señal *ack*); siendo en este caso generada por el módulo de ataque para mantener la integridad de las comunicaciones, y que el maestro no detecte nada anómalo. Luego, el maestro sigue enviando la trama sin que sea recibida por ningún esclavo, contestando únicamente el módulo de ataque suplantando la identidad del esclavo.

5.3 Sistema de detección. Sistema contra-medidas

Si un sistema es responsable de una tarea crítica, este sistema es susceptible de recibir diversos ataques, por lo que se debe de incluir algún tipo de estrategia para defender el mismo. El objetivo principal de esta estrategia debe consistir en frustrar los objetivos del sistema atacante. Para este caso particular, la política de seguridad más utilizada es la detección de variaciones en el periodo de la señal de reloj, la cual es implementada con un sensor de

frecuencia. Para nuestro estudio se va a utilizar como base el sensor de frecuencia citado en [Jim13]. Como se menciona en [Köm99], una solución estándar se basa en filtros analógicos. La comparación y el uso del sensor de frecuencia son presentados en [Jim13]. Los resultados no afectan ni modifican la implementación del sensor en la nueva aplicación.

Aunque el sensor puede ser materializado mediante técnicas VLSI [Jim13], en este capítulo se ha realizado un prototipo rápido implementando el sensor en una FPGA, con el objetivo de incluirlo en la plataforma experimental del capítulo anterior. La arquitectura del sensor se puede observar en la Figura 5.4(a). El sensor se encuentra compuesto por cuatro bloques: un detector de transiciones, que se encarga de detectar una nueva transición durante la monitorización de la señal e inicia las medidas de frecuencias; un oscilador local, para evitar ataques similares en el sensor; bloque de medidas, que realiza las medidas de los pulsos del oscilador local; y un bloque de salida, que genera la respuesta en el sensor. La respuesta generada en el bloque de señal es realizada cuando la frecuencia se encuentra fuera de rango. Aunque el concepto de sensor presentado en [Jim13] puede usarse en la aplicación Figura 5.4, existen varias razones por lo que no es aconsejable su uso:

- Los estados de espera en el proceso de comunicaciones, con una frecuencia de la señal *scl* igual a 0, serían identificados como ataques. Ello implicará una modificación en el detector de transiciones.

- El posible uso de relojes de baja frecuencia implicaría un elevado número de recursos en la generación del oscilador local. Ello implicará una modificación en el oscilador local.

- La respuesta del sensor en [Jim13] consiste en poner a cero la señal de reloj para que el sistema haga caso omiso a la situación de ataque. No obstante, esta misma situación es producida por el ataque considerado. Ello implicará una modificación en el bloque de salida.

Por todo ello, se ha variado ligeramente la arquitectura del sensor, así como la construcción de los diferentes bloques (excepto el bloque comparador, que no requiere ninguna modificación). Dicha arquitectura es mostrada en la Figura 5.4(b).

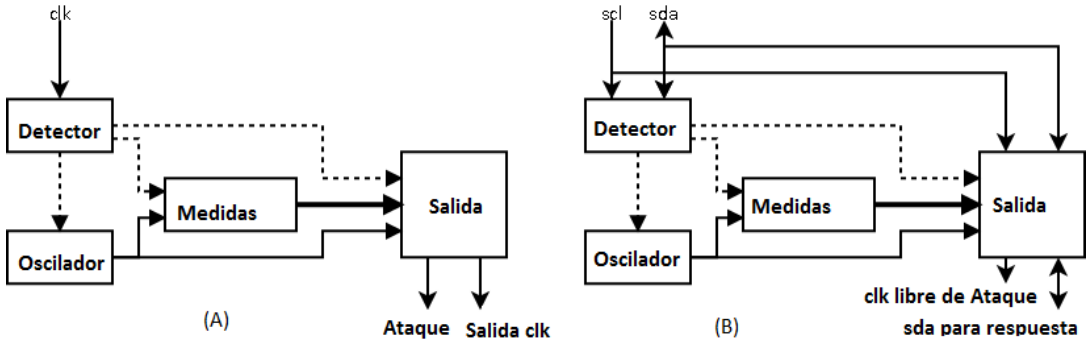


Figura 5.4 Arquitectura del sensor de frecuencia como contramedida a un ataque hacia la señal de reloj. (a) Arquitectura básica de [Jim13]. (b) Nueva arquitectura adaptada al protocolo I2C.

5.3.1 Detector de transiciones

La operación principal del detector de transición es identificar cuando ha comenzado una nueva operación, lo que ocurre en las transiciones de la señal *scl*. Una vez detectada dicha transición, el detector envía las órdenes de comienzo de operación al resto de bloques, como se muestra en la Figura 5.5. Dichas órdenes se envían a través de las señales *reset_osci*, *reset_salida* y *reset_medidas*.

El comportamiento del detector durante las transiciones se muestra en la Figura 5.5. El proceso de comunicación comienza con un estado de inactividad determinada por la situación de las señales *scl* y *sda* a nivel alto. Durante este estado, el sensor se encuentra deshabilitado con la activación de las señales de inicialización del resto de bloques. La llegada de una condición de *inicio* en el protocolo de comunicaciones habilita al sensor para que comience la medida del periodo en la señal *scl*. Dichas medidas son reiniciadas con la llegada de una nueva transición de subida en la línea *scl*. El ciclo anterior continúa hasta que se finaliza la comunicación o hasta que se detecta un ataque (que es la situación mostrada en la Figura 5.5). Una vez que se ha detectado el ataque, porque el periodo está fuera del rango permitido, se activa la señal *busy* indicando que el esclavo ha cortado la comunicación con el bus y está ejecutando la respuesta programada. Cuando finaliza la respuesta, el esclavo no vuelve a estar operativo hasta que no existe una nueva comunicación en el bus.

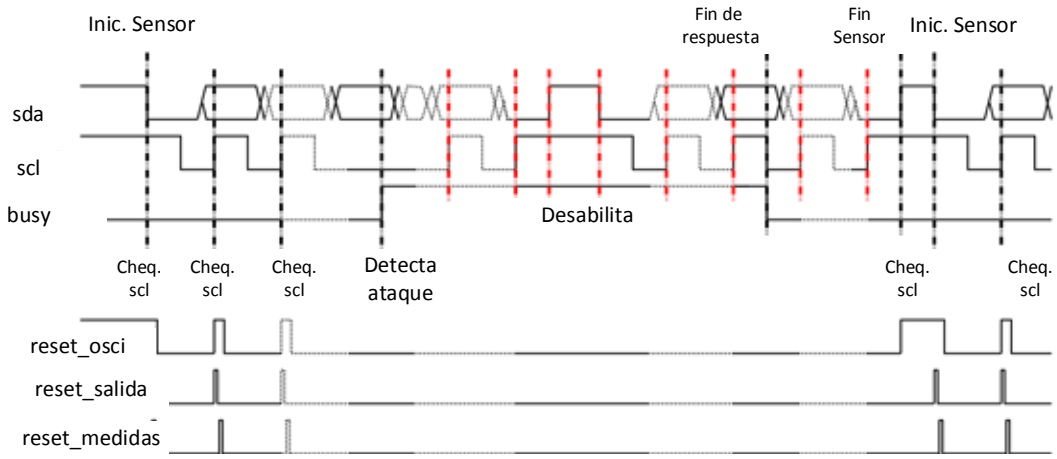


Figura 5.5 Comportamiento del detector de transición. Se considera los siguientes casos: una condición de inicio; Un ataque detectado; Una condición de parada; y una nueva condición de inicio.

Debido a que la señal monitorizada es la señal de reloj, el detector de transiciones será implementado utilizando una estrategia asíncrona (puesto que no se puede usar la señal de reloj). En la Figura 5.6, se ilustra los tres bloques que forman el detector de transiciones. En primer lugar, se dispone de un

circuito de reset, cuya principal función es la inicialización del bloque. En esta sección se identifica la llegada de un nuevo evento en la señal *scl* y las condiciones de "*start*" y "*parada*". En segundo lugar, una cadena de elementos de retraso provoca la temporización adecuada entre las inicializaciones del resto de bloques (oscilador local, bloque de medida y bloque de salida). En tercer lugar, una sección que genera las señales de inicialización para el resto de bloques en una secuencia adecuada.

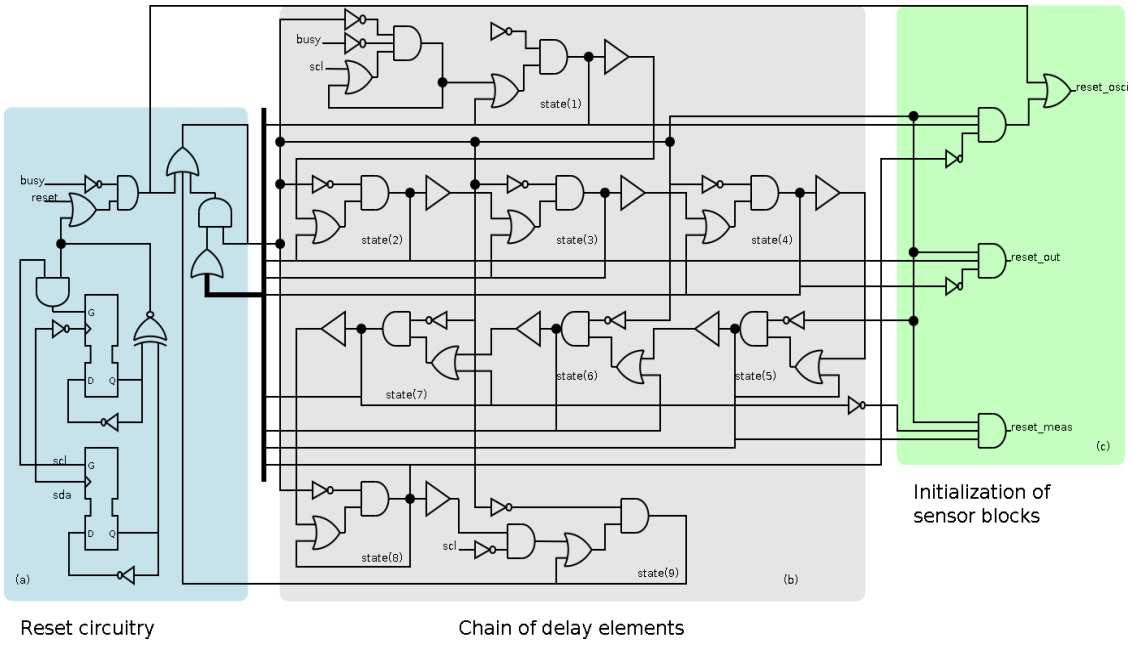


Figura 5.6 Esquema del detector de transición, identificado por tres secciones. (a) Circuito Reset. (b) Cadena de elementos de retardo. (c).Bloque que inicializa los sensores (oscilador local, bloque de medida y bloque de salida).

Merece la pena destacar que la cadena que garantiza el retraso, proporciona una secuencia correcta cuando los sensores son inicializados. Esta secuencia se detalla en [Jim13], lo que realiza en primer lugar es un reset y desactivar el oscilador, luego inicializa y activa el bloque de salida, más tarde inicializa y activa el bloque de medición para finalmente activar el oscilador. El componente principal que realiza este retraso es un buffer como se observa Figura 5.6(b).

5.3.2 Oscilador local

Una de las implementaciones más utilizadas para el diseño de un oscilador digital es una configuración en anillo. En esta configuración, existe un camino de realimentación positiva con un determinado retraso. El valor de dicho retraso es el que determinará el periodo del oscilador. En una implementación basada en FPGA, los elementos de retraso del camino de realimentación pueden ser implementados con flip-flops configurados como latches transparentes.

Aunque la función de este bloque es independiente de la aplicación, el uso de sistemas que operan a baja frecuencia requiere un excesivo uso de recursos (básicamente flip-flops). En la Tabla 5.1, se puede observar una comparación de recursos hardware para la implementación del sensor para detectar una señal de 12.5 kHz, variando el periodo del oscilador local.

El número total de flip-flops es determinado por la suma de los siguientes bloques: el oscilador local, el detector de transición, el bloque de medida y el bloque de salida. En el caso del oscilador local el número es igual a la relación entre el periodo y el retraso generado en un elemento. En el caso del detector de transición, se trata de un número fijo (puesto que su función es independiente de la referencia temporal utilizada), es 8. En el caso del bloque de medida, el dato es igual a la relación comprendida entre la mayor velocidad permitida en el periodo y la frecuencia del oscilador. En el caso del bloque de salida, el número es igual a la mitad de los flip-flops del bloque de medida más los elementos utilizados para implementar el bus I2C, este dato también es fijo.

En la Tabla 5.1 se muestra la implementación de un número mínimo de flip-flops que implica un compromiso entre una baja frecuencia, y la frecuencia más alta del oscilador local. Como consecuencia de este razonamiento, se

pueden utilizar dos alternativas: un oscilador local de alta frecuencia con un elevado número de flip-flops en los bloques de medida, o un oscilador local de baja frecuencia, con un número elevado de flip-flops en el oscilador local. Sin embargo, la solución más óptima de equilibrio se encuentra en una alta frecuencia en el oscilador local.

Oscillator Period	Flip-Flops				Total FF
	Oscillator	Detector	Measurement	Output	
4 ns	3	8	20,000	10,000	30,003
40 ns	21	8	2000	1000	3021
90 ns	48	8	889	445	1390
200 ns	105	8	400	200	713
400 ns	210	8	200	100	518
900 ns	472	8	89	45	614
4 us	2095	8	20	10	2133

Tabla 5.1 Estudio del número de flip-flop considerando el oscilador de periodo con un retardo de 1.91 ns. El límite superior del periodo es 80 μs (12.5 kHz).

La solución óptima se encuentra fijando la frecuencia del oscilador local a 400 ns, requiriendo un total de 518 flip-flops. La Tabla 5.1 muestra un estudio comparativo de la ocupación de un sensor en su configuración óptima (obtenido del estudio mostrado en la Tabla) con respecto a la cantidad total de recursos en diferentes familias de FPGA (todas ellas de la empresa Xilinx). Las familias consideradas son las familias actuales (Spartan 7, Artix 7, Kintex 7 y Virtex 7) y otras más establecidas (Spartan 3E, Spartan 3AN y Spartan 6). De todas ellas se han elegido dos modelos, los cuales tienen la menor y mayor cantidad de recursos hardware de la familia. En dicho estudio se puede apreciar que los recursos hardware necesarios para la implementación de un sensor para los modelos con menos recursos de las familias Spartan 3E, Spartan 3AN, Spartan 6 y Spartan 7 requieren un número de flip-flops superior al 5% del total. Entendemos que dicha situación es prohibitiva ya que implicaría una limitación en los recursos sobre el sistema de procesamiento que se pretenda

implementar. Adicionalmente, si el sensor tiene que actuar sobre frecuencias más bajas, esta limitación se hará más restrictiva.

Familia de FPGA	Modelo de FPGA	Flip-Flops		% ocupación en FPGA
		FPGA	**Sensor**	
Spartan 3E	XC3S100E	1920	518	26,98
	XC3S1600E	29504	518	1,76
Spartan 3AN	XC3S50AN	1408	518	36,79
	XC3S1400AN	22528	518	2,3
Spartan 6	XC6SLX4	4800	518	10,79
	XC6SLX150T	184304	518	0,28
Spartan 7	XC7S6	7504	518	6,9
	XC7S100	128000	518	0,4
Artix 7	XC7A12T	16000	518	3,24
	XC7A200T	269200	518	0,19
Kintex 7	XC7K70T	82000	518	0,63
	XC7K480T	597200	518	0,09
Virtex 7	XC7V585T	728400	518	0,07
	XC7V2000T	2443200	518	0,02

Tabla 5.2 Estudio del % de ocupación (con respecto al número de flip-flops) de un sensor, considerando la configuración óptima, para diferentes modelos de FPGA.

Por ello se ha buscado una solución que reduzca drásticamente los recursos del sensor, tomando un compromiso entre dichos recursos y la precisión (frecuencia del oscilador local) del mismo. Esta nueva solución propuesta consta de dos estrategias diferentes: un oscilador en anillo y un contador binario como divisor de frecuencia. Como se muestra en Figura 5.7, el oscilador en anillo generará con precisión un periodo, mientras que el divisor de frecuencias multiplicará el periodo sin necesidad de utilizar muchos recursos hardware. Por lo tanto, la nueva implementación utilizará dos parámetros de configuración: el número de elementos de retardo en el oscilador en anillo (*número_retardos*) y el número de flip-flops en el divisor (*número_multiplos*).

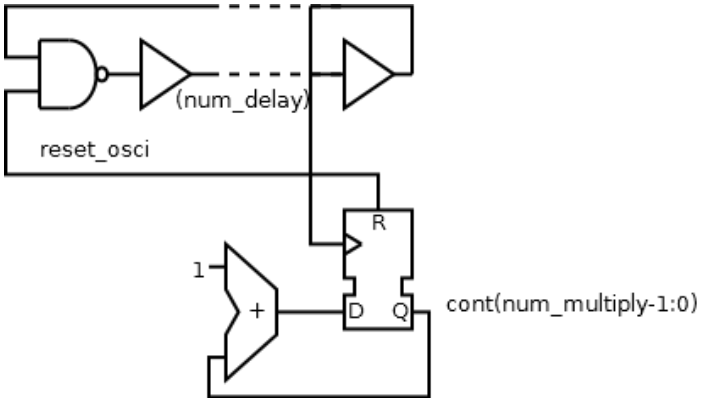


Figura 5.7 Nueva implementación del oscilador basado en un anillo oscilador y divisor de frecuencia.

El periodo del oscilador local (*POL*), en función de los parámetros anteriores, se muestra en la ecuación (5.1), donde $delay_{element}$ es el retraso de un elemento y $delay_{routing}$ es el retraso del conexionado.

$$POL = \left[\left(delay_{element} + delay_{routing}\right) * num_{delay}\right] * 2^{num_{multiply}}$$
$$number\ of\ flipflops = num_{delay} + num_{multiply} \quad (5.1)$$

La primera ecuación muestra el periodo del oscilador local. En ella se puede observar el periodo del oscilador en anillo, correspondiente a los retrasos del elemento y del conexionado, multiplicado por el número de elementos del anillo. Igualmente, se puede observar la contribución del divisor de frecuencia, correspondiente a la potencia de dos. La segunda ecuación indica el número de flip-flops del oscilador local. En ella se puede observar la contribución del oscilador en anillo (correspondiente al número de elementos del anillo), y la contribución del divisor de frecuencia (correspondiente al número de elementos del divisor).

El siguiente paso consiste en realizar un estudio comparativo entre la nueva estrategia basada en un oscilador en anillo seguido de un divisor de frecuencia, y un único oscilador en anillo. Dicho estudio comparativo se

muestra en la Tabla 5.3, en función de los parámetros *num_delay* y *num_multiply*. En dicha tabla se muestran los siguientes parámetros:

- *Nom. Period*: El periodo que se desea alcanzar con la implementación del oscilador (ya sea la configuración en anillo o la configuración mixta).

- *Num_delay* (en ambas configuraciones) y *num_multiply* (únicamente en la configuración mixta): Representarán los recursos hardware (en nuestro caso de flip-flops) necesarios en la implementación.

- *Period.*: El periodo alcanzado con la implementación específica del oscilador local, teniendo en cuenta tanto su configuración como sus elementos externos.

- *Error Period.*: Desviación existente entre el periodo deseable y el periodo alcanzado.

La comparativa ha sido realizada considerando alcanzar tres periodos diferentes (4 ns, 400 ns y 4 μs). El último de los periodos ha sido determinado porque el sensor debe monitorizar una señal de 80 μs (entendiendo que 20 ciclos es una resolución aceptable). En el caso de la configuración mixta (y siempre que sea posible) se han considerado tres versiones. La primera va a dar prioridad a la configuración en anillo, minimizando el parámetro correspondiente al divisor de frecuencia; la segunda va a primar al divisor de frecuencia, minimizando el parámetro de la configuración en anillo; y la última va a primar de forma conjunta ambas configuraciones. El estudio comparativo muestra que la implementación en anillo muestra la mínima desviación entre los periodos deseables y alcanzados. No obstante, dicha desviación mínima es obtenida a cambio de un número muy elevado de recursos hardware. En el caso

de la configuración mixta se obtiene que la mínima desviación se obtiene primando la configuración en anillo sobre la división de frecuencia, pero a costa de un número elevado de recursos hardware (como sucedía en el caso anterior). Cuando se prima la división en frecuencia, se produce una reducción de recursos a costa de una mayor desviación. Finalmente, se puede alcanzar un compromiso entre ambas opciones de tal forma que se reduzca la desviación sin aumentar significativamente los recursos hardware (en nuestros casos de estudio, siempre por debajo del 10%). De este modo, se obtiene la situación deseada, es decir, reducir los recursos hardware sin incrementar en demasía la desviación del periodo.

Nom. Period (ns)	Ring-Divisor Oscillator			Error Period (%)	Ring Oscillator		Error Period (%)
	num_delay	*num_multiply*	Period (ns)		*num_delay*	Period (ns)	
4	1	1	3.82	4.5%	2	3.82	4.5%
400	105	1	401.1	0.3%	210	401.1	0.3%
	1	8	489.0	22%			
	3	6	366.7	8.3%			
4000	1048	1	4003	0.07%	2095	4001.5	0.04%
	1	12	7823	96%			
	4	9	3912	2.2%			

Tabla 5.3 Comparativa del estudio del número de flip-flop que varía considerando el oscilador de periodo con un retardo de 1.91 ns. El límite superior del periodo es 80 μs (12.5 kHz).

5.3.3 Bloque de salida

La función principal del bloque de salida es generar la respuesta del sensor, que se trata de un caso particular para cada aplicación. En este caso, la respuesta trata de dividir en dos las diferentes acciones. En primer lugar, cuando no detecte ningún ataque, el sensor no debe emitir ningún tipo de respuesta, es decir no debe de realizar ninguna acción sobre las señales *sda* y *scl*. En segundo lugar, cuando se detecta algún tipo de anomalía en el bus producido por un ataque, la respuesta del sensor será desconectar al esclavo del canal atacado *sda* y *sda* e incluir en el esclavo un comportamiento de defensa para evitar posibles objetivos. La respuesta debe ser tal que el atacante no logre

su objetivo, por lo que lo hay que presuponer qué ha motivado al atacante para realizar el ataque, y así determinar cuál es el objetivo que pretende alcanzar.

En la Figura 5.8 se muestra un esquema que pertenece al bloque de salida. En ella se pueden identificar las diferentes acciones nombradas anteriormente, en este caso la señal del bloque de medida, identifica si existen anomalías en el protocolo I2C. En primer lugar las señales del bus se conectan directamente al multiplexor para mantener las señales *sda* y *sda* sincronizadas.

En segundo caso, dependiendo de la localización del sensor, la respuesta al ataque puede ser diferente. Por un lado el sensor se encuentra en el mismo sustrato (es decir en la misma zona de seguridad), por lo que el sensor tendrá acceso directo al registro incluido en la transmisión I2C. Por lo tanto, el sensor será capaz de escribir directamente un cierto valor en el registro con el fin de incluir el comportamiento en la defensa. En el caso de que el sensor se encuentre dentro de la misma zona de seguridad, pero en un sustrato diferente, no tiene acceso directo al registro del protocolo. En este último caso, el sensor desconectará al esclavo del bus general y proporcionará un bus local independiente en el que el sensor hará las funciones de maestro. Luego enviará las señales de sincronización y de datos adecuadas para generar la trama de respuesta establecida.

En la Figura 5.8 se puede observar que la defensa implicará dos acciones diferentes, implementadas en bloques independientes. En primer lugar, el sensor debe regenerar la señal de reloj *scl* recibida por el maestro respetando los tiempos de transiciones. Esta regeneración de la onda *scl* es realizada por el bloque "*Generador scl*". Este bloque es similar al usado en [Jim13], pero este necesita la información del oscilador local que es el periodo de la señal del reloj local, del detector de transiciones para iniciar las operaciones y la del bloque de

medición que es el periodo de la señal *scl*. Una vez regenerada la señal *scl*, se genera una señal en la línea *sda*, con un orden de defensa por elemento de respuesta en la señal. Esta orden puede representar una o varias transacciones I2C (puede ser necesario escribir en varios registros del esclavo). En nuestro caso particular se va a presuponer que el objetivo del atacante será que el robot no atienda a determinadas órdenes, por lo que seguirá avanzando según la última orden recibida. Luego la respuesta que se va a programar en el sensor será la parada de los motores. Por lo tanto, el robot se detendrá en el punto en el que el atacante quiere alterar su comportamiento. Las señales que necesitará este bloque serán la señal *scl* regenerada (para establecer la temporización), y la señal del bloque de medida que indicará la presencia de un ataque (para activar la respuesta).

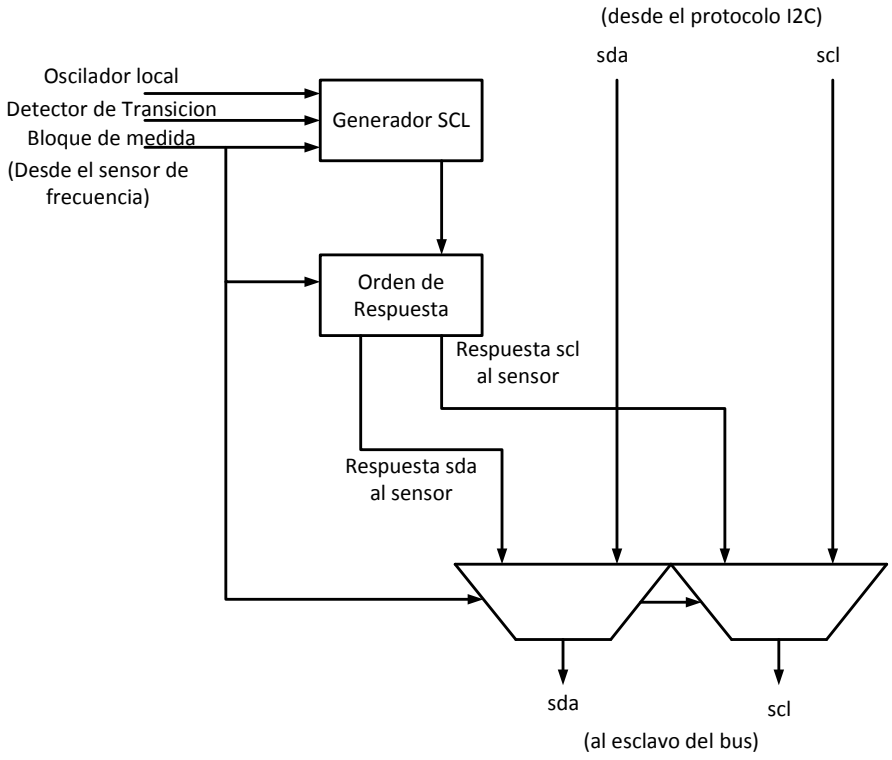


Figura 5.8 Nueva implementación del bloque de salida basado en el multiplexado del bus de datos.

5.3.4 Bloque comparador

Como ya se ha mencionado anteriormente, este bloque no precisa de ninguna modificación respecto al bloque del sensor original. Luego todas sus consideraciones pueden ser obtenidas de [Mag17].

5.3.5 Simulaciones del sensor

Los elementos que se han descrito en los subapartados 5.3.1, 5.3.2, 5.3.3 y 5.3.4 han sido implementados en una FPGA mediante el lenguaje VHDL. El uso de un oscilador en anillo implica que el periodo del reloj depende del retraso que pueda existir en el anillo, y por lo tanto, es necesaria una simulación encaminada a considerar estos retrasos. La implementación requiere el uso de un dispositivo determinado, y en este trabajo se ha usado una Spartan 3AN700. Sin embargo, el modelo VHDL permite implementar este diseño en cualquier plataforma o sistemas VLSI.

En la Figura 5.9 se observa la simulación de una transacción normal sin ataques. En dicha figura, el oscilador se encuentra en estado inactivo y el sensor comienza la inicialización cuando le llega una secuencia de inicio, esta secuencia comienza con el ciclo de la señal *scl*. No se identifica ningún ataque porque la señal de ataque no se encuentra activa. En este caso en particular, las señales *Salida_sda* y *Salida_scl* son copias de las señales *sda* y *scl* para evitar problemas de sincronización en la comunicación.

En Figura 5.10 se muestra un zoom de la forma de onda mostrada en la Figura 5.9, concretamente, muestra el funcionamiento del ciclo de la señal *scl*. Se puede apreciar que la señal de comparación está activa cuando el periodo entra en el intervalo permitido y se desactiva cuando llega un nuevo ciclo para

preparar una nueva verificación. Además, se muestra un detalle de la secuencia de inicialización en las señales de reset.

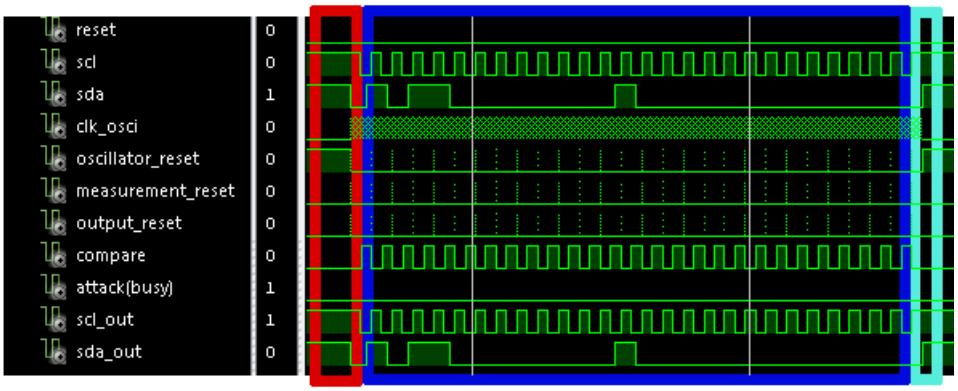


Figura 5.9 Simulación para una comunicación normal sin ataques en el protocolo I2C. Esta comunicación se ha realizado entre un maestro y un esclavo cuya dirección es X"B0". El maestro escribe el valor X"10" en el registro X"00".

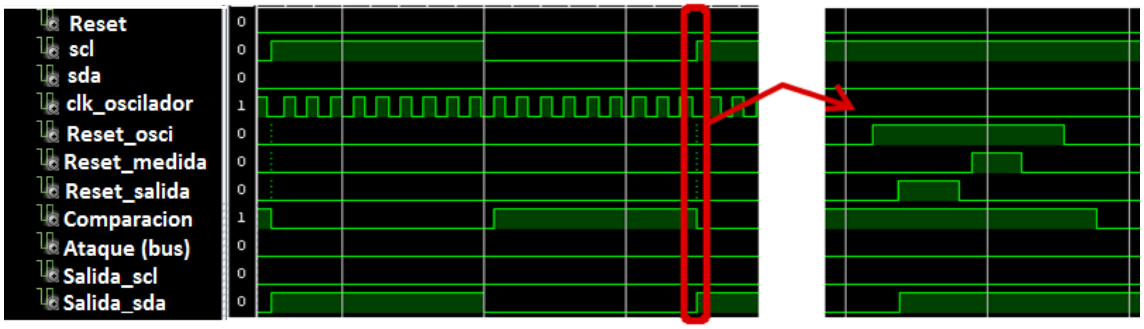


Figura 5.10 Zoom a la secuencia de inicialización de la Figura 5.9 y un detalle de la secuencia de inicialización.

En la Figura 5.11 se ilustra un ataque durante un proceso de comunicación, más concretamente en el octavo ciclo. En él, los primeros siete ciclos tienen un periodo correcto, y por lo tanto las señales, *scl* y *sda* pasan a las señales *Salida_scl* y *Salida_sda*. El octavo ciclo no llega porque el ataque quiere desactivar la comunicación, esta acción se identifica porque la señal de ataque se encuentra activada, y por lo tanto, el sensor envía una respuesta al esclavo. Esta respuesta comienza enviando una condición de parada, de modo que el esclavo cierra el proceso comunicación anterior, y comienza a escuchar la señal *sda*. Después se envía una condición de inicio y la dirección X "B0", que corresponde al esclavo. A continuación se envía la dirección de registro (X"00");

y finalmente el valor de escritura (X "10"). Una vez que se han terminado de mandar los datos, el sensor envía una condición de parada para terminar el proceso de comunicación.

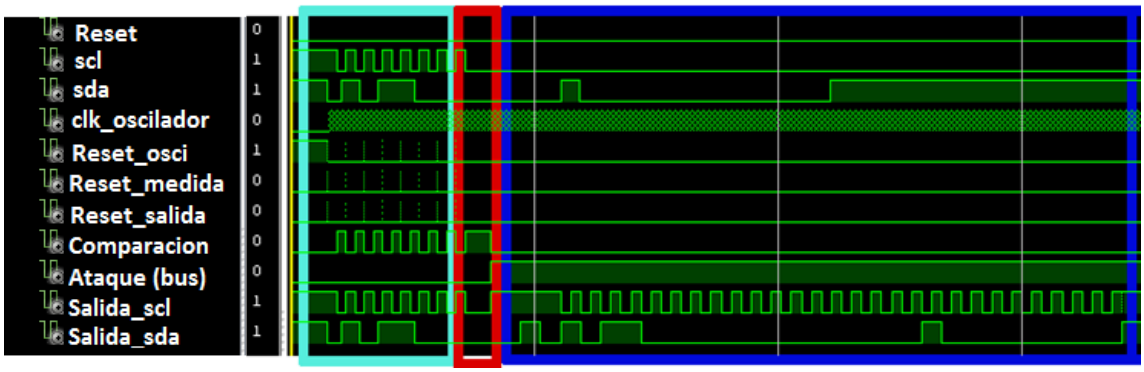


Figura 5.11 Simulación de un proceso de ataque cuando se establece una comunicación, con proceso de defensa activado.

5.4 Caso de estudio: Navegación de robots móviles

El interés de atacar a los robots se justifica por el uso de estos sistemas en una gran variedad de aplicaciones. El estudio presentado en este trabajo, se relaciona con la navegación de un robot móvil a lo largo de un camino definido previamente. Este caso representa una situación típica en muchas aplicaciones robóticas actuales (industria, agricultura, servicio, etc.). Realizadas ya sea en escenarios exteriores o interiores presentadas anteriormente en [Gom17], [Mor09], [Cue04], [Oll01] y [Nak15],[Kim11],[Cue04],[Oll01],[Nak15].

Se supone que las intenciones del usuario es visitar con el robot ciertas áreas de interés. Para este propósito se ha usado un algoritmo de planificación proporcionando un camino que circula junto a estas áreas. También se aplica un algoritmo de seguimiento de trayectorias para que el robot siga este camino con precisión.

Por lo tanto, aquellos que quieran modificar las intenciones del usuario deberán modificar las posiciones del robot, pero asegurando que el usuario no

se da cuenta de que está siendo atacado. Por consiguiente, la trayectoria planificada debe permanecer invariable y la interferencia externa puede ser aplicada al actuador de bajo nivel durante un corto periodo de tiempo, de modo que la perturbación tenga lugar temporalmente sin dejar rastros.

En la siguiente sección se describe con detalles la implementación citada anteriormente. Se han testeado situaciones de ataques con y sin defensa en la plataforma, emulando el movimiento de un robot y validando las hipótesis presentadas.

La plataforma experimental para la comprobación del sensor ha sido la misma que la descrita en el Capítulo 4.

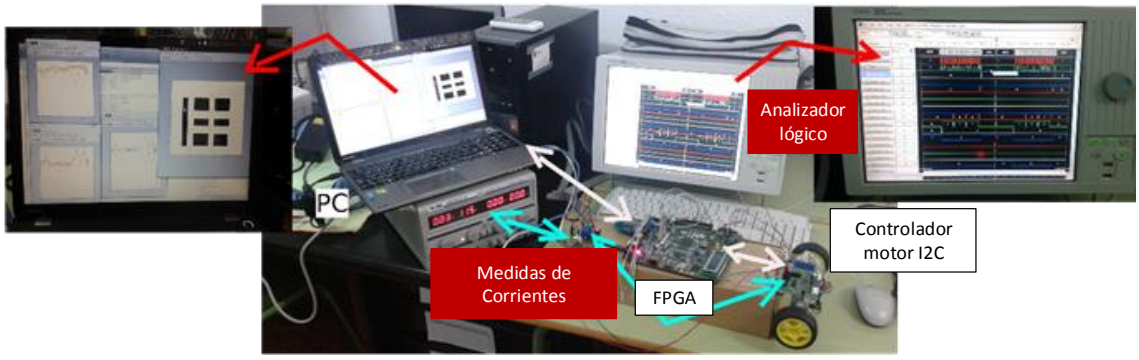


Figura 5.12 Plataforma robótica e instrumentación para las medidas.

5.4.1 Estrategias de defensa y ataques

5.4.1.1 Estrategias de ataques

La idea principal es atacar a la señal de reloj *scl* del bus I2C para interferir en las tareas programadas en el robot. En esta aplicación se ha generado la trayectoria que sigue el robot para asegurar que el vehículo pase por determinados lugares en un cierto orden. Como consecuencia, un primer objetivo para una estrategia de ataque sería modificar el curso del robot. En un momento seleccionado, cuando el maestro trata de comunicarse con los

motores, el módulo de ataque actúa sobre el reloj impidiendo dicha comunicación. Después se envía el acuse de recibo; debido a esto, se producen dos efectos:

- El módulo maestro considera que no ha habido ningún problema y el esclavo sigue las referencias.

- El módulo esclavo no recibe nada y mantiene el valor de velocidad anterior.

Las consecuencias pueden llegar a ser muy dramáticas: mientras el algoritmo de seguimiento de trayectoria ordena al robot que siga una trayectoria sin colisiones, el vehículo podría ir directamente hacia delante y estrellarse contra un obstáculo. Otra forma de llevar a cabo el ataque consiste en modificar la trayectoria del robot sin poner en peligro el propio robot, es decir, evitando colisiones e impidiendo que realice de forma correcta su tarea. Esta última acción es en la que nos centramos en los diversos experimentos. Según el criterio utilizado, un ataque con éxito debe cumplir un conjunto de restricciones:

(1) El robot debe seguir navegando de manera segura tanto durante el ataque como después del mismo.

(2) La consecuencia del ataque debe producir una de estas dos opciones:

   a. El robot no alcanzará una ubicación específica.

   b. El robot repetirá la ruta para visitar un área determinada.

Todos los requisitos pueden lograrse mediante un ataque apropiado que siga las prescripciones detalladas en la sección 5.2.1. Como se demostrará en la sección experimental, un ataque correctamente ejecutado tiene una posibilidad de éxito. El éxito sólo depende de seleccionar el momento óptimo para implementar el ataque de reloj. Las técnicas de ingeniería inversa pueden usarse para determinar el momento y la duración de la interferencia de la señal de reloj.

5.4.1.2 Estrategias de defensa

La estrategia para evitar este tipo de ataque consiste en desarrollar un módulo controlador similar al presentado en la sección 5.3 e ilustrado en la Figura 5.12. Cada vez que el sensor detecta un ataque en la línea *scl*, el módulo sensor está programado para generar una serie de órdenes que obligan a detener el funcionamiento de los motores. Una vez que el ataque a la señal de reloj ha cesado, el sensor detecta que la señal *scl* se encuentra libre de interferencias y vuelve a mandar los mensajes recibidos desde el maestro I2C. Seguidamente el robot puede iniciar la navegación con normalidad y llevar a cabo las tareas definidas.

La utilización de un esclavo estándar implica que el sensor no puede estar implementado en el mismo sustrato. Por lo tanto, la ubicación del sensor con respecto al esclavo y al bus de comunicaciones se muestra en la Figura 5.13. En ella se establece una zona de seguridad que incluye al sensor y al esclavo, de tal forma que la intervención del atacante se ejecutará fuera de la misma, y no entre ambos elementos.

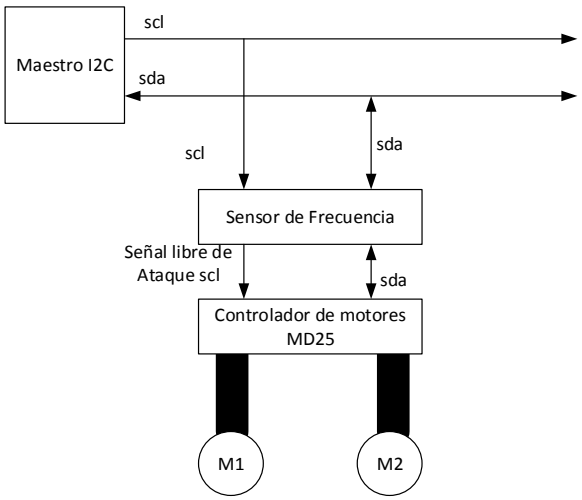
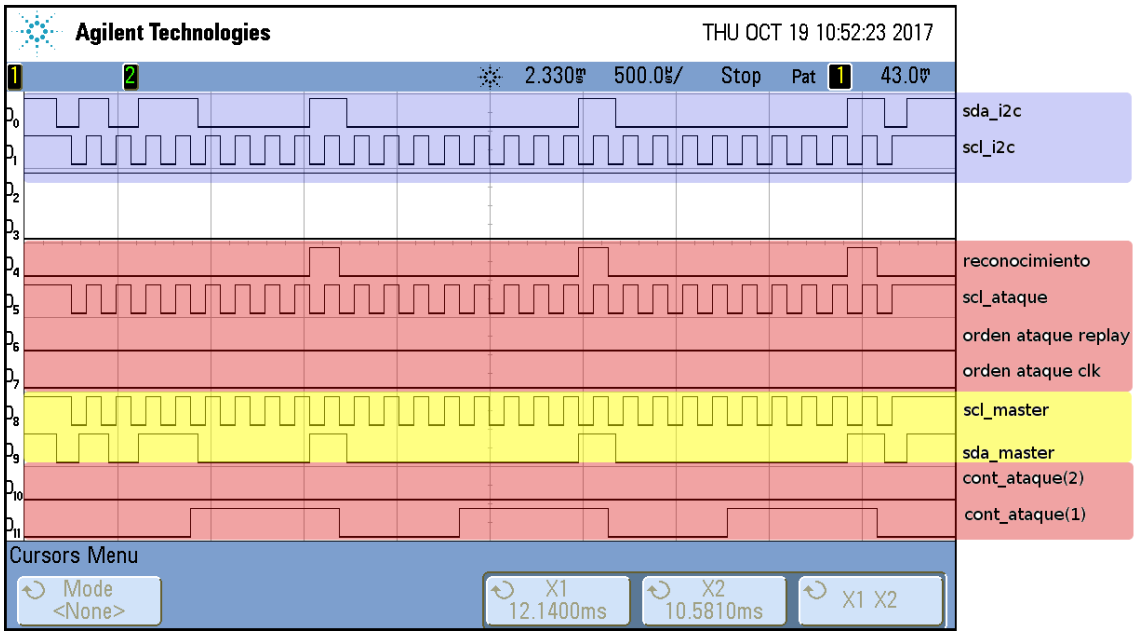


Figura 5.13 Esquema de la estrategia de defensa.

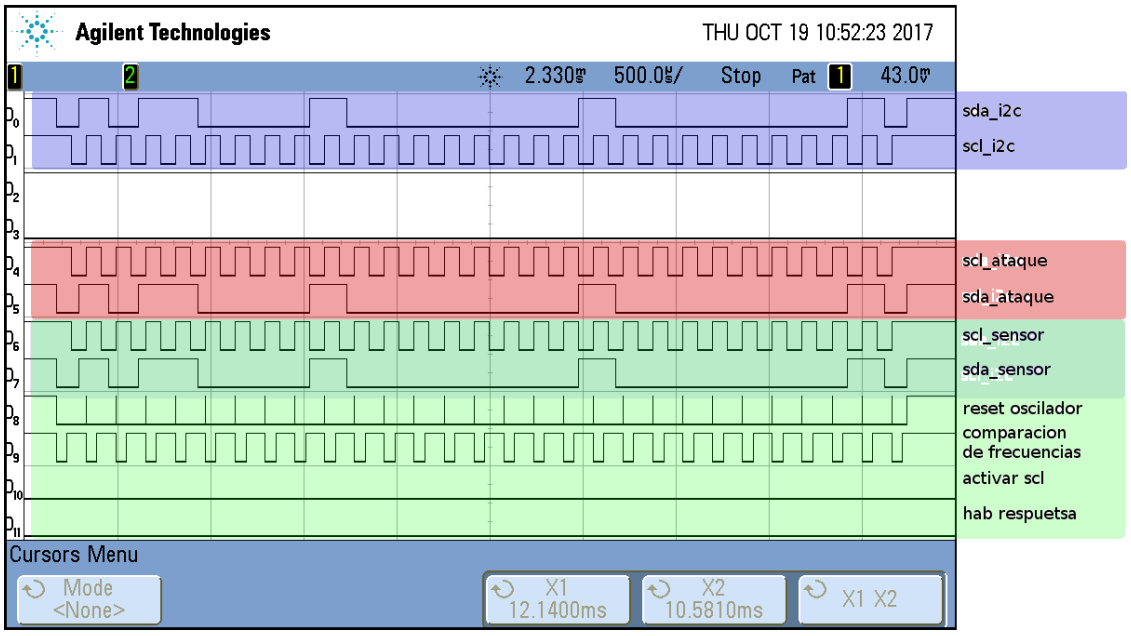
5.5 Resultados experimentales

En primer lugar, se va a mostrar el comportamiento real del sistema completo sin utilizar ninguna trayectoria real, sino simplemente órdenes I2C individuales. De hecho la orden utilizada es la escritura del valor 0 en el registro 0 del esclavo B0.

En la Figura 5.14 se muestra la respuesta del sistema a una situación en la que no se produce ningún ataque. Así, en la Figura 5.14(a) se destaca el sistema de ataque (cuyas ondas son destacadas en rojo), mientras que en la Figura 5.14(b) se destaca el comportamiento del sensor de frecuencia (cuyas ondas son destacadas en verde). En ambos casos, las ondas con fondo azul son las correspondientes a la entrada del dispositivo esclavo MD-32, después del sistema de defensa. Las ondas destacadas en amarillo son las correspondientes al maestro I2C que envía las órdenes.



(a)



(b)

Figura 5.14. Transmisión del paquete B0-00-00 sin ataque. (a) Comportamiento del ataque. (b) Comportamiento del sensor.

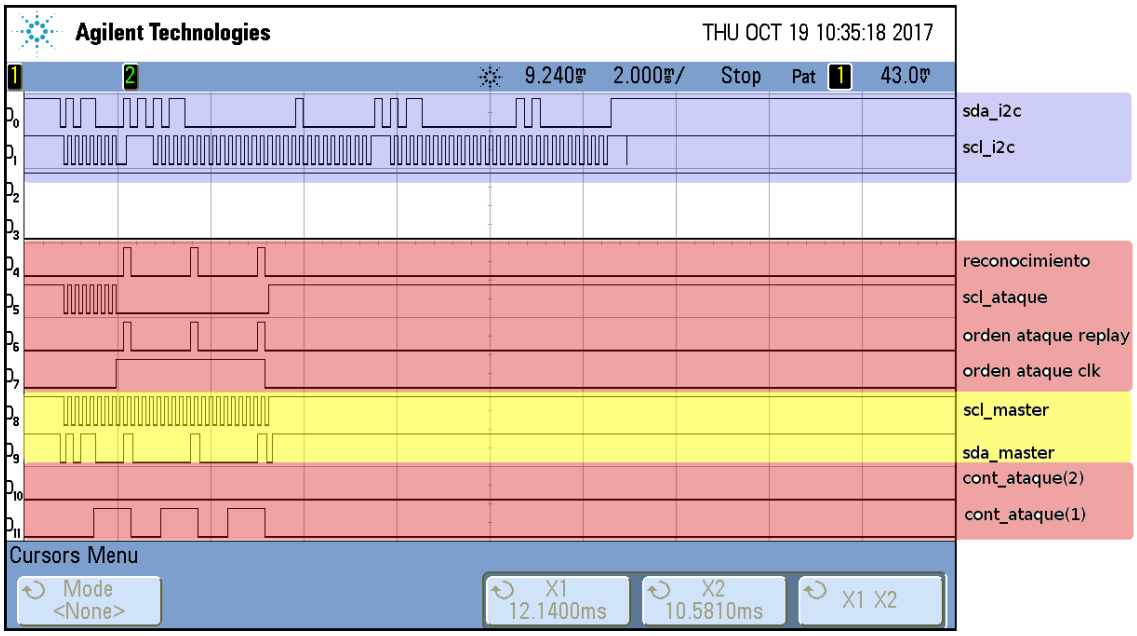
En la situación de no ataque se puede observar que el módulo de ataque simplemente no actúa sobre el bus (mostrado en la Figura 5.14a), indicado porque las señales de ataque para la señal de reloj (*orden_ataque_clk*) y para activar los reconocimientos (*orden_ataque_ack*). Luego, independientemente de

que el módulo de ataque reconozca todas las partes de la transmisión, las señales del bus I2C (*scl* y *sda*) son exactamente las mismas que envía el módulo maestro.

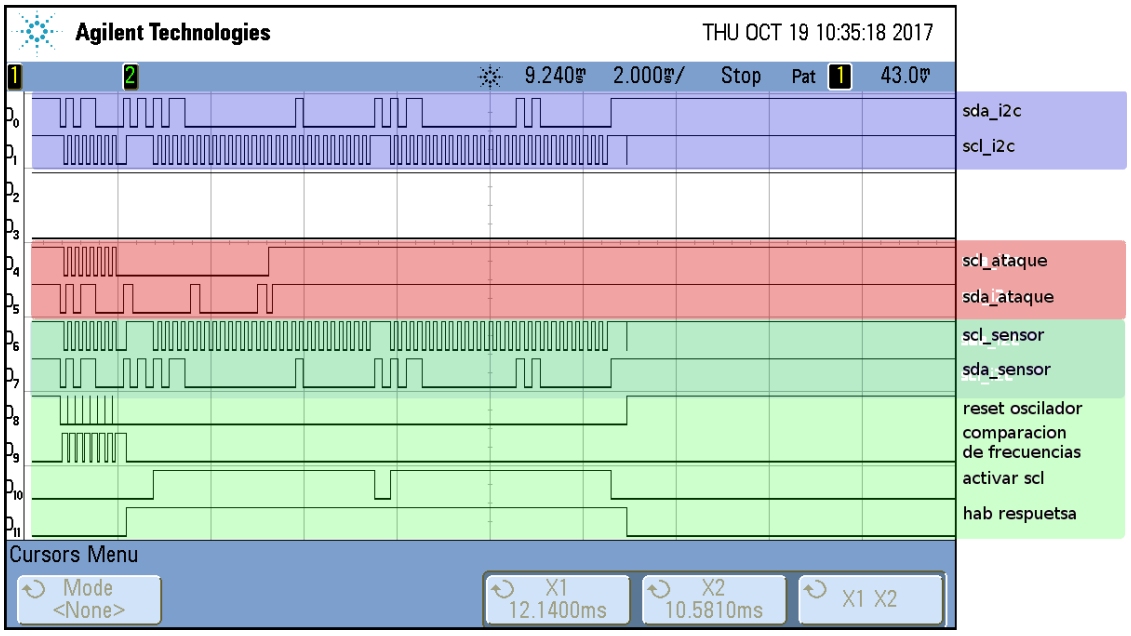
Con respecto a la Figura 5.14(b), el sensor evalúa que el periodo de la señal *scl* siempre esté en el rango permitido. Dicha situación se puede observar en el comportamiento de las señales *reset_oscilador* y *comparación_de_frecuencias*. La señal *reset_oscilador* se activa cuando llega un nuevo pulso de la señal *scl* o comienza una nueva transmisión, con el fin de comenzar una nueva evaluación. La señal *comparación_de_frecuencias* comenzará a nivel bajo indicando que el periodo (hasta ese momento) está fuera del rango permitido. Después se activará indicando que el periodo ha entrado en el rango permitido. Como llega el siguiente pulso (la siguiente activación de la señal *reset_oscilador)* mientras está activa, el periodo de la señal *scl* está en el rango permitido, y por lo tanto no se activa la defensa (las señales *activar_scl* y *hab_respuesta* no se activan). Este comportamiento se mantiene durante toda la transmisión.

Por el contrario, en la Figura 5.15 se muestra el comportamiento del sistema a una situación de ataque. Dicho ataque se va a producir sobre cualquier transmisión que tenga como destino el esclavo de dirección B0. Luego, como se puede apreciar en la Figura 5.15(a) el módulo de ataque se activa (a través de la señal *orden_ataque_clk*) cuando se ha transmitido la dirección, y antes del bit de lectura-escritura.

En ese momento, la señal *scl* del módulo de ataque permanece a nivel bajo durante el resto de la transmisión, para lo cual reconoce el momento de llegada de las señales de reconocimiento. Adicionalmente, las señales *orden_ataque_ack* activará las señales de reconocimiento con el fin de que el maestro no detecte ninguna anomalía en la transmisión.



(a)



(b)

Figura 5.15 Transmisión del paquete B0-00-00 con ataque. (a) Comportamiento del ataque. (b) Comportamiento del sensor.

En la Figura 5.15(b) se observa el comportamiento del sistema de defensa. Mientras se está enviando la dirección del esclavo, el módulo de ataque no actúa por lo que los periodos de la señal *scl* están en el rango permitido. No obstante, cuando el módulo de ataque detecta que la comunicación se quiere hacer con el esclavo a atacar, el módulo de ataque para la señal *scl*. En ese momento, el

sensor de frecuencia detecta que el periodo de la señal *scl* está fuera del rango permitido (ya que la señal *comparación_de_frecuencia* se desactiva antes de activarse la señal *reset_oscilador*), y se activa la respuesta del sensor (a través de la activación de la señal *hab_respuesta*). La respuesta consiste en la parada de los motores asociados al esclavo, por lo que se transmiten los paquetes B0-00-80 (para parar el motor 0) y B0-01-80 (para parar el motor 1) más las condiciones de inicio y de fin para completar las transmisiones. Ello implica generar una señal *scl* con el periodo de una transmisión válida, que no deberán pasar en las *activar_scl* condiciones de inicio y de fin. Por lo tanto, la señal indicará cuándo la señal *scl* debe llegar a la salida del sensor y cuándo debe permanecer en alta impedancia.

Utilizando la plataforma experimental descrita en el capítulo 4, se han realizados muchos experimentos en diferentes entornos; la idea es encontrar esos caminos y escenarios de los ataques explicados en la sección 5.4.1.2. En los diversos experimentos se muestra la modificación de la trayectoria del robot cuando la misma ha sido predefinida en el controlador de alto nivel.

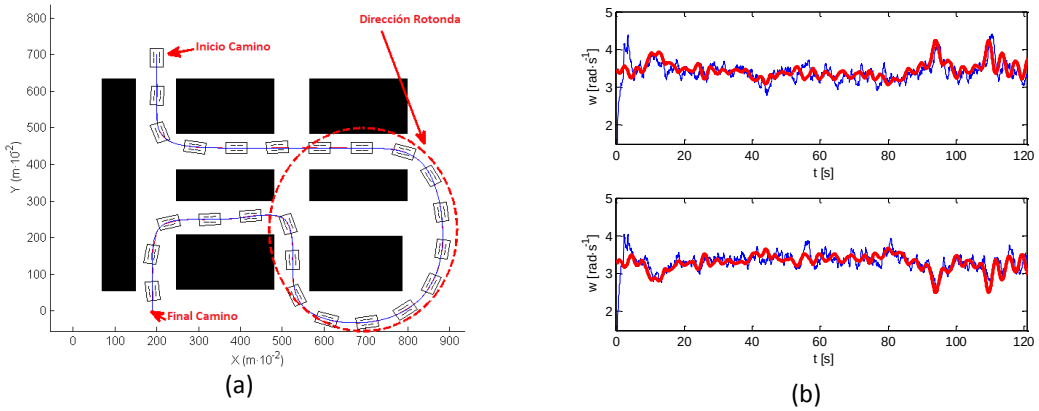


Figura 5.16 Experimentos sin ataques. (a) Trayectoria planificada y robot. (b) Velocidad de las ruedas y señal de los codificadores.

Uno de los resultados más interesantes se ha encontrado en el escenario que se observa en la Figura 5.16(a). En esta figura se muestra con flechas el inicio y final de la trayectoria donde el robot se debe de mover rodeando una

zona específica. En este primer experimento mostrado, la navegación se ha realizado sin ningún tipo de ataque, por lo que el algoritmo "*path-tracking*" ha controlado en todo momento la trayectoria planificada en el robot. En la Figura 5.16(b) se ilustra la referencia de las velocidades angulares en rojo provenientes del controlador de alto nivel, y en azul la velocidad real que procede de los datos monitorizados en los encoders (motor derecho en la gráfica superior y motor izquierdo en la gráfica inferior).

En el siguiente experimento, ilustrado en la Figura 5.17, se ha implementado un ataque al funcionamiento de los dos motores. En Figura 5.17(a) se puede observar en rojo el camino del robot y en azul la trayectoria que debía de haber seguido. En la Figura 5.17(b) se ilustra la velocidad impuesta por el controlador de alto nivel en rojo, en azul la velocidad real monitorizada de los encoders de los motores, y en negro el periodo de tiempo que ha sido atacado el protocolo de comunicaciones.

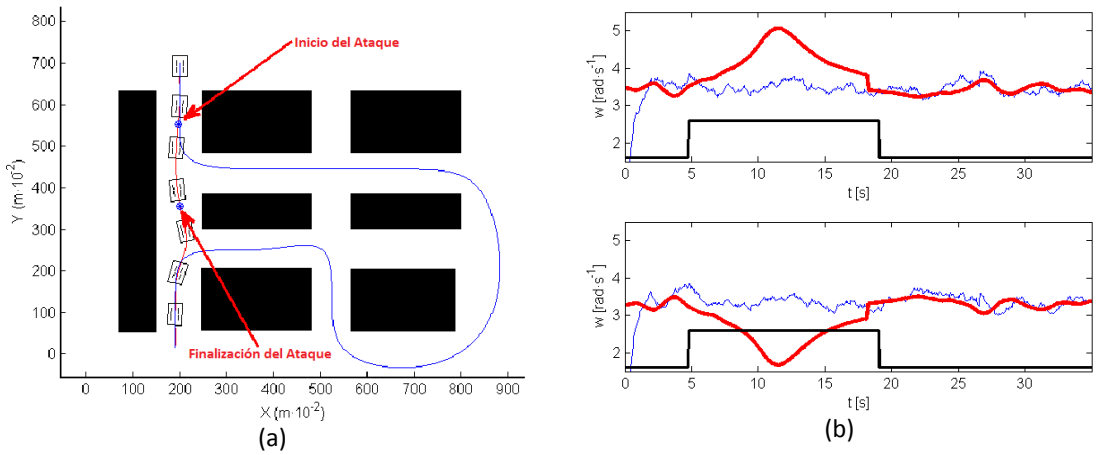


Figura 5.17 Evitando la rotonda. (a) Trayectoria planificada y robot. (b) Señal de los codificadores.

Se puede observar en la Figura 5.17(b) cómo, durante el ataque, el controlador de alto nivel envió referencias a los motores para hacer que el robot se moviera a la sección más cercana de la trayectoria, sin embargo, y debido al ataque, ninguno de los motores escucha estos comandos. Una vez ha finalizado

el ataque el robot detecta como siguiente punto cercano la finalización de la trayectoria, por lo que realiza el camino hasta allí. Una consecuencia de este ataque ha sido que el vehículo no ha visitado zonas del camino que entraban dentro de su trayectoria definida inicialmente.

En el siguiente experimento mostrado en la Figura 5.18, se ha realizado el ataque con la idea de que el robot visite dos veces la misma zona. En la Figura 5.18(a) se puede observar cómo el vehículo rodea dos veces la misma rotonda, repitiendo un camino que no se encuentra marcado en su trayectoria inicial. Junto a esta figura, se ilustran nuevamente las velocidades angulares definidas por el controlador de alto nivel en rojo, en azul las velocidades reales leídas desde los encoders, y negro el tiempo que dura el ataque.

El ataque a la señal de reloj ocurre cuando la rotonda estaba terminando. A partir de este momento, el robot navega en línea recta. El ataque finaliza cuando el robot estaba cerca del comienzo de la rotonda; entonces, el algoritmo inicia el seguimiento de la trayectoria, según supone que se encuentra el robot, hasta finalizar, por lo el robot sigue la rotonda nuevamente, visitando esta área dos veces, ya que el algoritmo no sabe que ha sido hackeado.

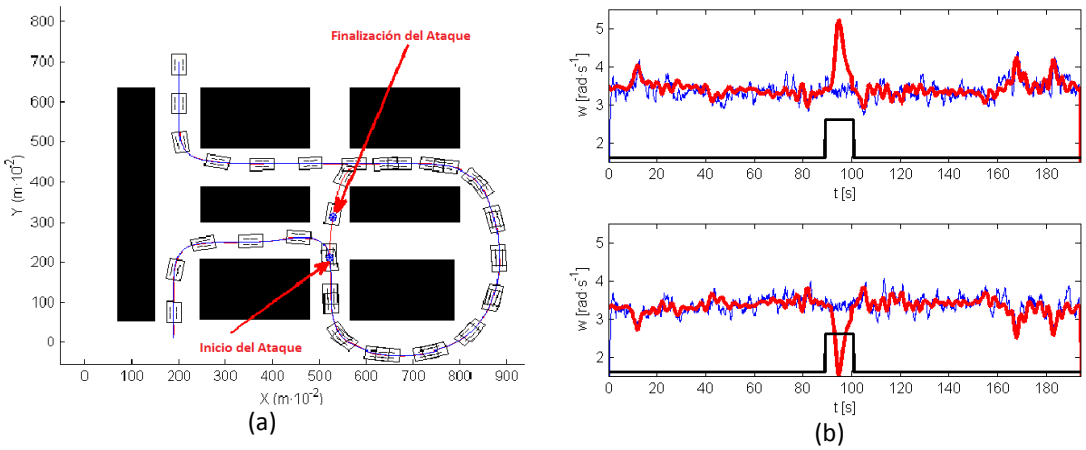


Figura 5.18 Visitando la rotonda dos veces. (a) Trayectoria planificada y robot. (b) Señales de los codificadores.

Estos resultados confirman que una conveniente elección para el momento del ataque puede hacer que el robot cambie el comportamiento esperado. Sin embargo, los movimientos resultantes parecen ser "naturales"; de hecho, el robot navega de forma segura y regresa a la trayectoria deseada. Sólo aparece un comportamiento extraño de los motores, que probablemente nadie podría entender, sin saber que el sistema está siendo objeto de un ataque.

El siguiente experimento ilustra la eficiencia del sensor propuesto en la defensa de los ataques anteriores. El sensor de frecuencia se conectó entre el MD23 y el bus I2C protegiendo al driver de los motores de los ataques. En el experimento mostrado en la Figura 5.19(a) se puede observar con flechas rojas dónde se han producido dos ataques en los mismos instantes que en los experimentos anteriores. En este caso, el vehículo se ha defendido de los ataques como se puede observar y ha podido terminar la trayectoria sin ningún tipo de percance. En la Figura 5.19(b) se ilustran las señales correspondientes a las velocidades angulares ordenadas por el controlador central (en rojo), las velocidades angulares monitorizadas en el esclavo (en azul), y la orden de ataque (en negro). Se puede observar que mientras no se produce ningún ataque, el seguimiento de las velocidades angulares del controlador por parte del esclavo es adecuado. En cambio, cuando se produce un ataque, destacan dos situaciones. En primer lugar, el controlador sigue enviando órdenes a través de velocidades angulares tratando de que el robot siga en movimiento; por lo tanto, no tiene información de que está siendo atacado. En segundo lugar, el sensor sí ha detectado el ataque y ha ejecutado el mecanismo de defensa. Dicho mecanismo ha independizado el esclavo del bus, y ha programado que ambos motores se detengan (la línea azul cae a cero mientras dura el ataque). Una vez que el atacante ha liberado el bus, el sensor detecta la falta de ataque, y vuelve a conectar el esclavo en el bus de comunicaciones.

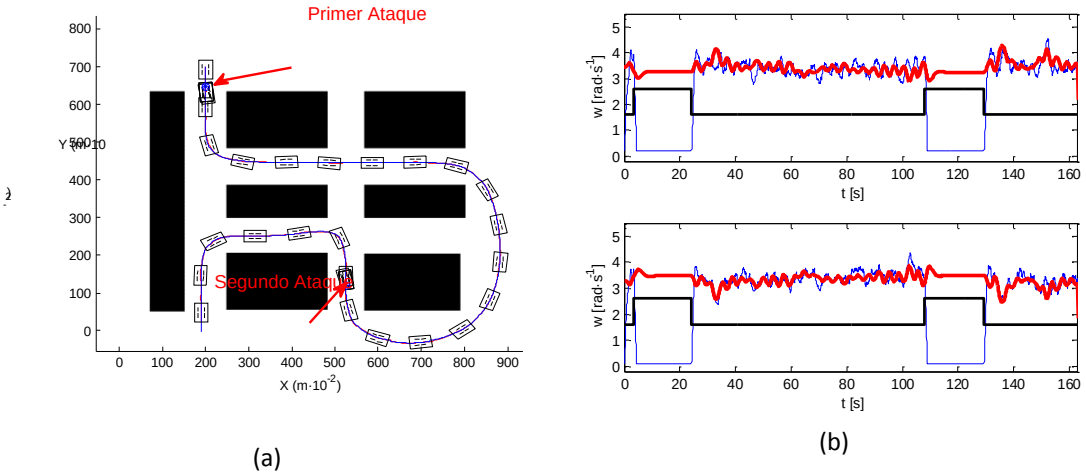


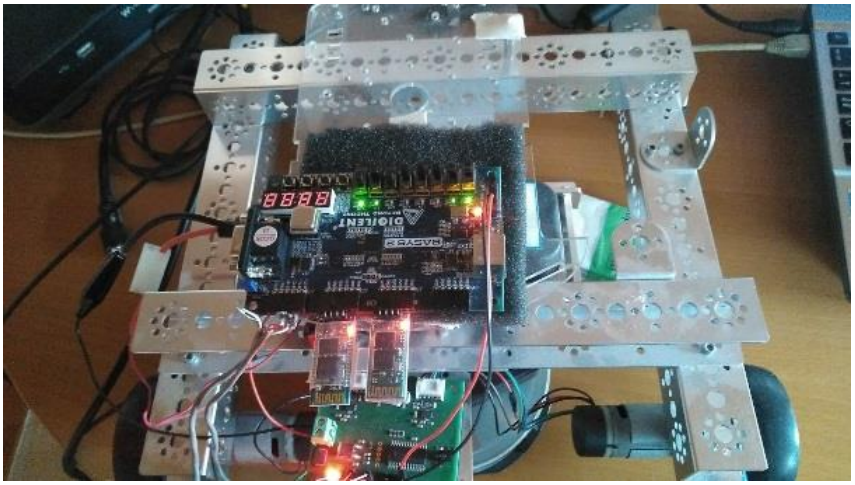
Figura 5.19 Evitando los ataques. (a) Trayectoria planificada y robot. (b) Señal de los codificadores.

El sistema también ha sido desconectado de la plataforma experimental para dar la posibilidad de movimiento real al robot. Para ello se han realizado pruebas con las siguientes modificaciones:

- Se han realizado pruebas sobre dos placas de desarrollo basadas en FPGA con diferente tamaño y peso. La primera de ellas ha sido una placa reducida del modelo Basys2 [Bas17] basada en el dispositivo Spartan 3E-100. La segunda de ellas ha sido la misma placa utilizada en la plataforma experimental, basada en el dispositivo Spartan 3AN-700.

- Para dotar de más libertad al robot móvil, se ha sustituido la conexión RS232 por cable, por una conexión Bluetooth. Para ello, se ha utilizado el módulo HC-06 [HC-06]. Luego, las órdenes del controlador central llegarán a través de una comunicación inalámbrica.

- Se ha añadido un nuevo canal serie (a través de otra conexión Bluetooth) para controlar el módulo de ataque. De esta forma, la orden de activación y desactivación de ataque puede ser enviada mediante un teléfono móvil.



(a)

(b)

Figura 5.20 Fotografía de la plataforma móvil utilizando (a) la placa de desarrollo basada en Spartan 3E-100 y (b) la placa de desarrollo basada en Spartan 3AN-700.

- Se han utilizado dos formas de generar el movimiento del robot. En primer lugar, el movimiento ha sido generado a través de una trayectoria planificada, como la comentada en el capítulo 4. En segundo lugar, el movimiento ha sido generado directamente por el usuario a través de la conexión de un *joystick* al controlador central.

En la Figura 5.20 se muestran las fotografías de las plataformas móviles basadas en las dos placas de desarrollo. En ambos casos, el peso no es un problema porque la potencia de los motores utilizados es suficiente para ejercer el par adecuado; y los resultados obtenidos verifican los mismos resultados anteriores cuando el robot está en vacío, es decir, las ruedas no están sobre el suelo.

5.6 Conclusiones

En este capítulo se ha ilustrado un ejemplo de ataques a las señales de reloj del protocolo de comunicaciones I2C, y se ha propuesto una estrategia de defensa a dichos ataques. Particularmente se ha estudiado el ataque sobre la señal *scl* en los procesos de comunicaciones entre el maestro y esclavo. Se propone el diseño de un nuevo sensor que representa una efectiva defensa frente a las diversas perturbaciones. Las estrategias de defensa y ataques han sido validadas en una plataforma experimental, que emula el funcionamiento de un robot diferencial. Se han realizados varios experimentos relacionados con la navegación de robots móviles, en circunstancias particulares se ha caracterizado que los ataques a reloj I2C aumentan de forma considerable la vulnerabilidad del sistema. Los experimentos muestran la eficiencia del sensor.

Capítulo 6. Conclusiones y líneas futuras de investigación

6.1 Conclusiones

El presente trabajo pretende ser un punto de partida en la seguridad y fiabilidad en las comunicaciones inalámbricas y cableadas. Por un lado, se ha implementado un sensor capaz de detectar la influencia de señales externas en un sistema de comunicaciones cableadas. En otro sentido, se han realizado diversas aplicaciones inalámbricas obteniendo la fiabilidad en zonas hostiles.

El estudio de un estándar de comunicaciones inalámbricas es usado como punto de partida para realizar una serie de estrategias firmware. Estas estrategias dotan al estándar IEEE 802.15.4 de la capacidad de asociar nodos móviles. Además, centrado en la misma línea de fiabilidad en las comunicaciones inalámbricas, se ha efectuado la siguiente aplicación: se trata de un sistema para monitorizar en tiempo real motores de corriente alterna, consiguiendo un alto rendimiento energético gracias al diseño hardware realizado y consiguiendo alta fiabilidad en la transmisión de información en zonas hostiles.

En busca de un mayor estudio en la seguridad de las comunicaciones cableadas, se ha optado por estudiar el protocolo de comunicaciones I2C. Para

ello, se ha diseñado una plataforma móvil que cumpla los suficientes requisitos como para poder monitorizar cada una de las señales que se encuentran dentro de la plataforma. Una vez realizados los experimentos en la plataforma, se ha podido demostrar de forma fehaciente la vulnerabilidad del bus de comunicaciones I2C cuando se producen ataques hardware a la señal de reloj. Para solucionar este problema se ha implementado un sensor que detecta cuándo un dispositivo está siendo atacado, y seguidamente manda la orden de parar los motores, evitando daños mayores.

En la tesis se han alcanzado varias contribuciones de alto nivel, tanto a nivel experimental como a nivel de desarrollo e implementación. Las principales contribuciones, en las vertientes de seguridad y de fiabilidad en las comunicaciones cableadas e inalámbricas, son las siguientes:

- Sistema autónomo y de alta fiabilidad dotando al estándar de comunicaciones IEEE 802.15.4 de la posibilidad de desarrollar estrategias de comunicaciones hacia nodos móviles. Para ello, se ha estudiado la fiabilidad de las comunicaciones en una serie de aplicaciones donde el estándar no era capaz de abarcar.

- Se ha diseñado e implementado un sistema de alta fiabilidad y alto rendimiento energético, capaz de gestionar de forma inalámbrica y autónoma las posibles averías que se puedan producir en un sistema de motores, asegurando su mantenimiento preventivo. Se han realizado experimentos tanto en laboratorio como en campo.

- Se ha diseñado una plataforma para el estudio de comunicaciones cableadas. Se han estudiado los efectos que produce un ataque a la señal de reloj del protocolo de comunicaciones I2C, probando y

validando la vulnerabilidad de este protocolo de comunicaciones a tales ataques.

- Considerando un tipo particular de ataque a la señal de reloj en el protocolo I2C, se propone la implementación de un nuevo sensor para detectar y defender contra este tipo de perturbaciones. El análisis del ataque y la defensa se valida mediante una plataforma experimental configurable que emula un robot de accionamiento diferencial. Los resultados experimentales confirman el interés de las vulnerabilidades estudiadas y la eficiencia del sensor propuesto en la defensa contra este tipo de situaciones.

6.2 Líneas futuras de investigación

Como líneas futuras de interés para este trabajo, que se abren a partir de las propuestas iniciales, se proponen:

- Buscar estrategias para el aumento de nodos móviles en las redes de sensores inalámbricas basadas en el estándar de comunicaciones IEEE 802.15.4.

- Realizar estrategias para aumentar la fiabilidad en redes de sensores inalámbricos con nodos móviles dentro del estándar de comunicaciones IEEE 802.15.4.

- Proponer estrategias para aumentar la autonomía y la fiabilidad de redes inalámbricas en zonas hostiles de trabajo.

- Implementación de un sistema de alto rendimiento energético con estrategias más seguras dotando al estándar de comunicaciones IEEE 802.15.4 de una mayor fiabilidad.

- Estudiar el ataque a la señal de reloj y su respuesta en otros tipos de protocolos de comunicaciones.

- Implementar un sensor más genérico y que no sea tan dependiente de la aplicación.

Anexo 1. Estudio de Tecnologías Inalámbricas

Anexo 1.1. Introducción

En la tesis se ha realizado el estudio de varios protocolos y estándares de comunicaciones, tanto inalámbricas como cableadas. Se ha buscado la comunicación tanto cableada como inalámbrica más idónea para el estudio, y caracterizar los posibles problemas por medio de diversos experimentos.

En este Anexo se detalla el breve estudio realizado para la selección de comunicaciones tanto cableadas como inalámbricas.

Anexo 1.2. Introducción a las comunicaciones inalámbricas

Las redes de sensores inalámbricos WSN (*Wireless Sensor Network*) se encuentran en continuo desarrollo debido a su creciente auge en un gran número de aplicaciones tan variadas como: monitorización ambiental, aplicaciones militares, monitorización y control industrial, aplicaciones médicas…

El número de aplicaciones para este tipo de redes es tan elevado, que a pesar de los grandes avances de investigación realizados, siguen siendo muchos los problemas a resolver, girando la mayoría de ellos en la búsqueda de

soluciones de bajo consumo de energía. El abanico de oportunidades a la investigación en WSN es amplio, tales como nuevos protocolos de comunicación, servicios de baja latencia, sincronización de la red, adaptabilidad de la red, etc.

En los siguientes apartados se pretende realizar un pequeño análisis de los estándares y especificaciones de comunicaciones inalámbricas existentes en el mercado. Para cada comunicación se analizarán diversos aspectos relacionados con WSN para determinar la aplicación en la que serían adecuados. En los siguientes apartados se elegirá un estándar determinado y se buscará una posible solución a una red con nodos móviles.

Anexo 1.3. Redes de sensores inalámbricos

La tecnología MEMS (*Micro-Electro-Mechanichal Systems*) ha contribuido de forma significativa al mencionado auge de las redes de sensores inalámbricos. Estos avances han facilitado el desarrollo de los sensores, los cuales presentan las siguientes características: tienen reducido tamaño, una limitada capacidad de procesamiento, recursos limitados y son más económicos que los sensores tradicionales. Estos nodos sensores pueden tomar medidas y recoger información del entorno, para posteriormente, basado en un proceso de decisión local, transmitir los datos recogidos a una estación base donde son analizados y visualizados por el usuario.

Los nodos sensores son dispositivos de bajo consumo equipados con uno o más sensores, un procesador, un transceptor (*transceiver*) de comunicación y posiblemente un actuador. Dado que los nodos sensores tienen una memoria limitada, y normalmente son dispuestos en localizaciones de difícil acceso, la

inclusión del *transceiver* de radiofrecuencia implementa una comunicación inalámbrica para transferir los datos a un nodo sumidero (*sink*) de información.

En los nodos que conforman una red WSN la batería suele ser la fuente principal de energía; y en ocasiones, va acompañada de un segundo sistema de generación de energía (*harvesting*) que es capaz de extraer energía del entorno (por ejemplo células solares) para aumentar el tiempo de vida útil del nodo sensor.

Las infraestructuras que poseen las WSN no son prefijadas, ya que consisten en una serie de nodos sensores que trabajan juntos para obtener datos del ambiente donde se encuentren, o controlar una zona. Hay dos tipos de redes inalámbricas de sensores: estructuradas y no estructuradas. Una WSN desestructurada es aquella red que contiene un denso número de nodos, donde los sensores pueden ubicarse de forma *ad-hoc*. Una vez desplegada la red, se deja desatendida para realizar el seguimiento y presentación de informes. Por otro lado, en una red estructurada, la gestión de la conectividad y detección de fallos es difícil debido a la cantidad de nodos existentes. Su principal ventaja es el número de nodos que llevan el mantenimiento de la red, con lo que el coste energético es menor [Jen05].

Las aplicaciones de las WSN han ido creciendo y variando con el paso del tiempo: seguimiento de objetivos militares[Chi09],[Sim04]; monitorización (e incluso prevención) de desastres naturales [Cas04]; biomedicina, con la finalidad de monitorizar pacientes [Lor04]; monitorización de condiciones meteorológicas [Gao05]; agricultura [Kew06]; rescates en fuegos a campo abierto [Bul01]; o incluso exploraciones ambientales y previsiones sísmicas [Wen06].

A diferencia de las redes inalámbricas tradicionales, una WSN posee su diseño propio y limitaciones en los recursos, que incluyen limitación de energía, poca cobertura de los dispositivos, reducido ancho de banda, baja capacidad de procesamiento y escaso almacenamiento en los nodos. Las limitaciones de la red son muy dependientes de la finalidad de la aplicación y del ambiente a controlar; y las condiciones físicas del lugar donde se ubique la WSN forma un papel clave para determinar el tamaño, el despliegue y la topología de la misma [Jen05]. Existe una gran diferencia en aplicaciones *indoor* donde se necesitarán menos nodos, respecto de aplicaciones en campo abierto donde el número de nodos para recopilar información será mucho mayor, para cubrir el mayor espacio posible. Una implementación *ad-hoc* es preferible utilizarla cuando son ambientes inaccesibles para el ser humano, o cuando la red se compone de miles de nodos. Las obstrucciones del medio limitan las comunicaciones entre los nodos, que a su vez afectan a la conectividad de la red o a la topología.

Los nodos pueden tener una ubicación fija o no. Las redes de estos dispositivos pueden ser de tipo tradicional utilizando un solo nodo sumidero o *sink*, o utilizar varios de ellos como se ha propuesto en algunos artículos [Chi09]. Las redes *single-sink* son poco escalables, ya que al aumentar el número de nodos, la información que recogen aumenta de tal forma, que llega un momento en que estos dispositivos no pueden recoger más datos por dificultades de enrutamientos y otros motivos relacionados con la MAC (*Media Access Control*) [Chi09]. Por otro lado, en las redes *multiple-sink* la densidad de los nodos puede ser mucho mayor, ya que al aumentar el número de dispositivos sumidero en la red se disminuye el número de nodos por *sink.* De esta forma, disminuye la probabilidad de que existan grupos aislados de nodos, que no hagan llegar la información a la puerta de enlace.

En principio, las redes WSN utilizan la idea de varios sumideros de información hacia las puertas de enlaces, con la finalidad de que la información llegue a su destino en el menor tiempo posible y con el menor número de saltos. La Figura 1 ilustra lo comentado anteriormente.

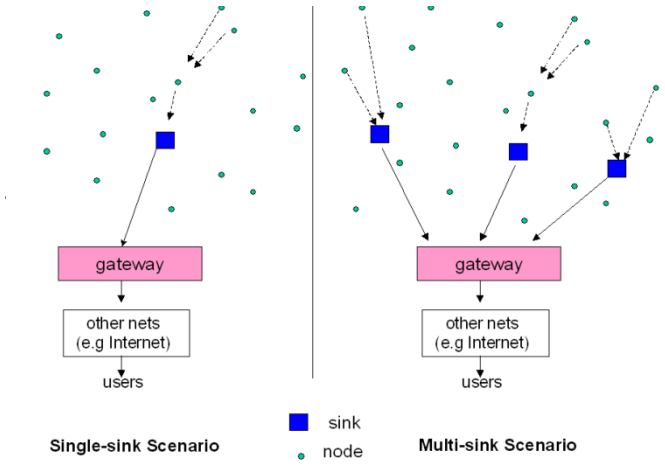


Figura 1. Red *single-sink* (a la izquierda); red *multi-sink* (a la derecha)

En la realización del diseño de una red de sensores inalámbricos, se deben tener en cuenta los siguientes aspectos [Ian02]:

- *Eficiencia energética, o limitaciones de energía*. Debido a las aplicaciones de estos dispositivos, en ocasiones el acceso de los seres humanos al medio es casi inviable (aplicaciones tales como estudios de hormigón, bajo agua…), con lo cual, cuanto menor consumo tengan estos dispositivos más información se tendrán en los estudios realizados.

- *Medio de transmisión*. Las redes de sensores cableadas son seguras, pero el mantenimiento y coste de la instalación supera con creces los costes de los dispositivos sensores. En las comunicaciones inalámbricas, en el 90% de los casos, la comunicación que se usa es radiofrecuencia (RF), aunque también existen comunicaciones como

infrarrojo (IR), laser y ultrasonido. Estos últimos medios de comunicación son los menos usados, debido a que los sensores deben "verse" para poder realizar la transmisión de datos [Ian02].

- *Tolerancia de fallos*: Se debe prevenir el fallo de nodos que se encuentran en el camino de la señal (entenderemos a partir de ahora como nodos *routers*), con lo que se buscará el menor tiempo en la transmisión de la información (menor latencia), y que genere menor consumo.

Restricciones del hardware: estos dispositivos poseen ciertas limitaciones debido a su pequeño coste energético. Dichas limitaciones deben ser reconocidas para obtener mayor calidad en la red.

- *Escalabilidad*. El número de nodos sensores desplegados para el estudio de una zona o fenómeno puede elevarse a miles, dependiendo de la aplicación. Los nuevos esquemas deben ser capaces de trabajar con un número de nodos muy elevado y utilizar alta densidad de redes. La densidad puede variar desde unos pocos nodos a cientos de ellos en una región [Ian02].

- *Coste*. No sólo hace referencia al coste en la instalación y al tipo de comunicación inalámbrica usada, sino también al coste de los nodos, que debe ser muy reducido, ya que se prevén aplicaciones donde las redes estén compuesta por cientos de sensores, así como el coste del mantenimiento. Al encontrarse los nodos sensores en lugares recónditos, no conviene tener una instalación de cables, ni llevar un mantenimiento preventivo, sino llegar y cambiar los nodos averiados por completo.

- *Entorno*. Los nodos están densamente desplegados y ubicados muy cerca de un fenómeno meteorológico, bajo agua en un océano o áreas geográficas prácticamente inaccesibles, entre otros muchos ejemplos

- *Topología*. La red debe permitir una reestructuración ya que debe ser capaz de obtener información de nodos móviles. Otra característica importante es que permita la comunicación *Peer-to-Peer*.

En redes WSN se montan tres topologías, estrella, árbol y malla:

- *Estrella*: Una red en estrella es aquella en la cual los nodos están conectados directamente a un punto central (*sink*), y todas las comunicaciones se han de hacer necesariamente a través de este nodo. Los dispositivos no están directamente conectados entre sí, y no se permite demasiado tráfico de información.

- *Árbol*: puede verse como una combinación de varias topologías en estrella. Tanto la de árbol como la de estrella son similares: cuando el nodo de interconexión trabaja en modo difusión, la información se propaga hacia todos los nodos; solo que en esta topología las ramificaciones se extienden a partir de un punto raíz (estrella) a tantas ramificaciones como sean posibles, según las características del árbol.

- *Malla*: En esta topología cada nodo está conectado a todos los demás nodos. De esta manera, es posible llevar los mensajes de un nodo a otro por diferentes caminos. Si la red de tipo malla está

completamente conectada, no puede existir absolutamente ninguna interrupción en las comunicaciones.

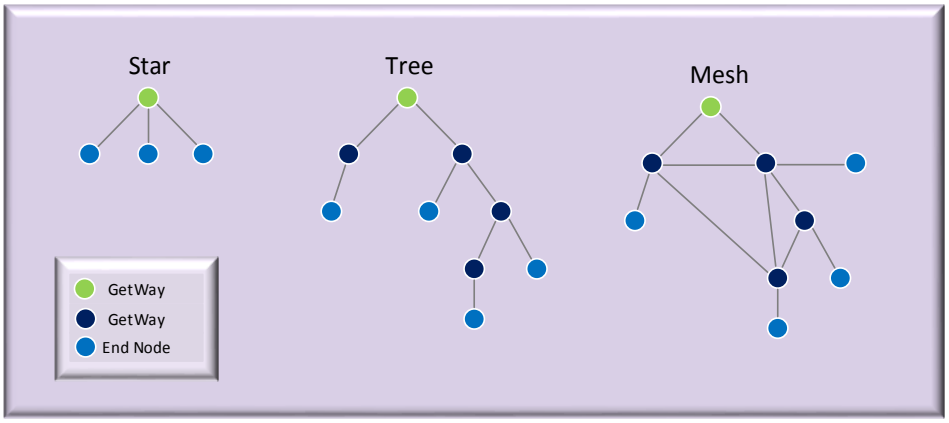


Figura 2. Topologías existentes de redes WSN: estrella, árbol y malla.

- *Cobertura*. La cobertura es un aspecto muy relacionado con la topología y el consumo de energía. En primer lugar, si la topología permite asociar un nodo móvil a cualquier otro nodo cercano, la cobertura del nodo móvil debe elevarse a medida que se aleja del nodo asociado. Si por el contrario la topología utilizada no permite nodos móviles, debido a la aplicación o por sí mismas, los dispositivos pueden estar más dispersos para coberturas más grandes, o existirá mayor densidad si la cobertura es menor.

Estos son algunos de los requerimientos generales a tener en cuenta en diseño de una red WSN. Teniendo en cuenta que la red es muy dependiente de la aplicación, cada aplicación necesita priorizar algunos de estos parámetros, lo que provocará la degradación de otros ya que tienen efectos contrapuestos. Por ejemplo, cobertura frente a consumo de potencia: un aumento de cobertura conlleva un aumento en el consumo de potencia a la hora de realizar la transmisión. Otro ejemplo es la latencia: si decidimos hacer una red con baja latencia, se deben mantener los nodos despiertos, lo que hace que también aumente la potencia de la red. Si consideramos la tolerancia de fallos en la red,

ésta es inversamente proporcional a la latencia: si se pretende transmitir un paquete de datos sin fallos, se pueden colocar bits de ACK (*acknowledge*), lo que conlleva un aumento de tiempo en pasar el dato a otro nodo, ya que el nodo transmisor debe esperar la respuesta del nodo receptor aumentando de forma considerable la latencia.

En el diseño de una WSN, no sólo hay que tener en cuenta los parámetros anteriores, también existen otros elementos menos significativos, pero muy influyentes para el programador:

- *Tasa de datos:* depende del consumo de energía, de la encriptación de la red y la latencia. Si se necesita incluir una cabecera para encriptar la red, se pierden datos del *payload*. Si por el contrario se necesitan enviar más datos, el tiempo de transmisión puede variar dependiendo de la cantidad de datos, con lo cual el nodo estará más o menos tiempo despierto.

- *Sincronización*. La sincronización de las redes de sensores es muy importante ya que de ella depende la latencia. Si un nodo tarda mucho tiempo en despertar, aumenta la latencia de la red de forma muy considerable, de forma que los fines de la monitorización no se consiguen. Por otro lado, si la red se despierta a la vez, y se consigue encaminar los datos hacia la pasarela en el menor tiempo posible, se conseguirá bajar la latencia de la red, monitorizar en el menor tiempo posible y reducir el consumo.

- *Robustez*. Los dispositivos que funcionan en bandas ISM *(Industrial, Scientific and Medical)* pueden recibir interferencias cuando se encuentre cerca otro dispositivo, aunque sea del mismo fabricante.

Para conseguir una robustez frente a estas interferencias, existen estándares que proponen soluciones como: saltos en frecuencias, saltos en canales o funcionar en un solo canal.

- *Seguridad*. La gran ventaja de la comunicación cableada frente a la inalámbrica es que nadie puede obtener los paquetes que se envían entre dos nodos. Las tecnologías inalámbricas han realizado grandes avances en este sentido, incluyendo cabeceras para las encriptaciones, perdiendo datos en *payload* para obtener más seguridad en las redes, y encriptado señales.

- *Calidad en el servicio (QoS, Quality of Service)*. Este parámetro es uno de los más importantes que define una red WSN: se busca la emisión y recepción de todos los datos con la menor latencia posible. Como se ha visto anteriormente, la latencia es dependiente de la sincronización de la red, y la emisión de datos es muy dependiente de la seguridad.

Anexo 1.4. Tecnologías inalámbricas

El desarrollo de una pila de protocolo fiable y eficiente es muy importante para las aplicaciones en WSN. El número de nodos es dependiente de la aplicación, y cada nodo utiliza una pila de protocolos para comunicarse entre ellos y con el nodo sumidero. Por lo tanto, la pila de protocolos debe ser eficiente en términos de comunicación, y debe ser capaz de trabajar de manera eficiente en términos de energía en los diferentes protocolos.

Algunas de las tecnologías inalámbricas que se encuentran en el mercado se exponen en los siguientes sub-apartados, detallando los estándares y

especificaciones inalámbricas más representativas para su aplicación en una WSN.

a. WiFi (Wireless Fidelity)

La designación de las normas de red 802 son realizadas por IEEE (*Institute of Electrical and Electronics Engineers*). Los estándares 802.11 LAN inalámbrico denota un conjunto de normas que ha desarrollado el grupo de trabajo 11, de ahí la denominación definitiva del estándar IEEE 802.11.

En la actualidad se incluyen seis técnicas de modulación [IEE17]. Para la transmisión de datos que cumplan dicho estándar. Además, también varía el número de protocolos de interconexión entre nodos. De dichos protocolos los más conocidos son 802.11a, 802.11b y 802.11g, que recogen las modificaciones en la seguridad del estándar original y que es reforzada a través de la enmienda 802.11i. Otras normas de la familia (c-f, h, i, j y n) implementan mejoras en los servicios y ampliaciones o correcciones de las especificaciones.

Los estándares IEEE 802.11b y 802.11g utilizan una banda de frecuencia a 2.4 GHz, mientras que el estándar IEEE 802.11a utiliza una banda de 5 GHz. Sin embargo los problemas aparecen en los dispositivos que operan en una banda de frecuencia de 2.4 GHz y no se encuentran reguladas por el estándar ya que sufren interferencias con dispositivos tales como teléfonos móviles, microondas y otros dispositivos que cumplan los requisitos ISM.

En la siguiente tabla se puede observar el gran número de variaciones que ha recibido la norma [Gon04].

Norma	Modificación
IEEE 802.11	Inicio de la norma con velocidad de 1-2 Mbps, en un estándar IR 1.999
IEEE 802.11a	54Mbps, a 5GHz la norma se termina en 1.999, los dispositivos de comercializan en 2.001
IEEE 802.11b	Mejora de la norma inicial, soporta tasa de datos de 5.5 a 11Mbps. 1.999
IEEE 802.11c	Operaciones puentes. Incluido en el estándar IEEE 802.1D (2.001)
IEEE 802.11d	Extensiones internacionales entre países (2.001)
IEEE 802.11e	Mejora en QoS, incluido el envío de paquetes (2.005)
IEEE 802.11F	Protocolo de acceso 2.003
IEEE 802.11g	Tasas de 54Mbps a 2.4GHz (compatible con el estándar b) 2.003
IEEE 802.11h	Administrador del espectro de la señal del estándar 802.11ª para la compatibilidad en Europa 2.004
IEEE 802.11i	Mejora de la seguridad 2.004
IEEE 802.11j	Extensión para Japón 2.004
IEEE 802.11k	Mejoras en los recursos de medición de la señal
IEEE 802.11l	Reservado para tipologías
IEEE 802.11m	Mantenimiento de la norma.
IEEE 802.11n	Mejoras en el rendimiento
IEEE 802.11o	Reservado para tipologías
IEEE 802.11p	WAVE, incorporación a vehículos móviles.
IEEE 802.11q	Reservado para tipologías
IEEE 802.11r	Mantenimiento de la norma
IEEE 802.11s	Redes Mesh
IEEE 802.11T	Wireless Performance Prediction (WPP), test para métodos y mediciones
IEEE 802.11u	Conexión de redes 802 con telefonía
IEEE 802.11v	Gestión de redes inalámbricas
IEEE 802.11w	Gestión de seguridad en las redes

Tabla 1 Evolución de la tecnología WiFi.

A pesar de la variedad de tecnologías IEEE 802.11x, posee una serie de características idóneas para la implementación de WSN como son:

- Alta escalabilidad.
- Posibilidad de topologías *peer-to-peer*.
- La configuración de un nuevo nodo a la red es elevada.

Sin embargo, el consumo de energía se postula como inconveniente para su utilización en redes de sensores inalámbricas, ya que el dispositivo no se puede dormir, pues pierde la configuración de la red. Otro inconveniente importante es que la pila de protocolos para este estándar necesita de amplios

recursos hardware para su ejecución, mientras que los nodos que integran una WSN disponen, en general, de muy pocos recursos de almacenamiento y procesamiento. En definitiva, aunque esta comunicación tiene ventajas muy importantes para comunicaciones inalámbricas, sus características suponen un duro obstáculo para el diseño de redes de sensores inalámbricas.

b. WiMax

El estándar de comunicaciones IEEE 802.16 es un estándar de comunicaciones inalámbricas de banda ancha, capaz de proporcionar transmisiones en áreas de hasta 48 Km de radio con velocidades de hasta 100Mbps [IEE16]. *Wireless Interoperability for Microwave Access* (WIMAX) es una certificación que pasó un test de seguimiento del estándar IEEE 802.16. En la siguiente tabla se muestran los estándares que forman WIMAX.

Estándar	Descripción
802.16	WiMAX, rango de frecuencia de 10 a 66GHz
802.16a	WiMAX para estaciones fijas, frecuencias inferiores a 11GHz
802.16b	Frecuencias exentas de licencia, rango de frecuencias de 5 a 6 GHz
802.16c	Detalles del sistema para la banda de 10 a 66GHz
802.16d	Estándar.
802.16-2004	Reemplaza a los estándares 802.16ª y 802.16d (incluye OFDMA)
802.16e	WiMAX para estaciones en movimiento (velocidad limite de 120km/h; tamaño FFT 128, 512, 1024 y 2048)
802.16f	Gestión MIB (Base de Información de Gestión)

Tabla 2 Estándar WiMax.

WiMax integra la familia de estándares IEEE 802.16 y el estándar HyperMAN del organismo de estandarización europeo ETSI (*European Telecommuniations Standards Institute*). El estándar inicial 802.16 se encontraba en la banda de frecuencias de 10-66 GHz y requería torres con línea de visión

directa. La posterior versión, 802.16a, ratificada en marzo de 2003, utiliza una banda del espectro más estrecha y de menor frecuencia, de 2-11 GHz, facilitando su regulación. Además, como ventaja añadida, no requiere de torres donde existan enlaces con línea de visión directa, sino únicamente el despliegue de estaciones base (BS) formadas por antenas emisoras/receptoras con capacidad de dar servicio a unas 200 estaciones suscriptoras (SS) que pueden dar cobertura y servicio a edificios completos. Su instalación es muy sencilla y rápida (culminando el proceso en dos horas) y su precio es competitivo en comparación con otras tecnologías de acceso inalámbrico como WiFi.

En el año 2004 se publicó el estándar de comunicaciones IEEE 802.16-2004, que especifica la interfaz de la capa MAC y múltiples capas PHY, además del acceso desde un punto fijo a una banda ancha inalámbrica FBWA (*Fixed Broadband Wireless Access*). Con esta versión se conseguían mayores velocidades en grandes áreas geográficas aumentando el número de los puntos de acceso, y también se mejoró la calidad de la señal. La última versión ha contado con la colaboración del grupo de trabajo NWG (*WiMax Forum Group*) que desarrolla especificaciones para sistemas móviles de WiMax, por lo que la cooperación entre IEEE 802.16 y NWG ayuda a definir una red WiMax.

Los sistemas móviles de la norma ofrecen una alta escalabilidad, tanto en tecnología de acceso como en arquitectura de red, proporcionando así una gran flexibilidad en las opciones de despliegue de la red. Las características más importantes son:

- *Alta tasa de datos*. Esto se logra al incluir técnicas MIMO en antenas, junto con esquemas flexibles para la utilización de canales, avances de codificación y modulación. Con todas estas características permiten a la

tecnología aportar velocidades de pico de 63Mbps para datos DL (*Down Link*), y de 28Mbps para datos UL (*Up Link*) en un canal de 10MHz.

- *Elevada QoS*. La premisa fundamental de la tecnología es la calidad proporcionada en el servicio. Para mantener los niveles de QoS, la especificación posee una serie de negociaciones que obliga a garantizar unos niveles mínimos de conexión para establecer el enlace [Mac93].

- Escalabilidad unas 200 SS (*Subscribers Station*) por cada BS.

- La utilización del espectro de frecuencias inalámbricas es distinto en cada país, de forma que cada país posee su ancho de banda y permite el funcionamiento en sólo algunas frecuencias. La tecnología WiMax se diseñó para ser capaz de trabajar con diferentes canales y con distintos anchos de bandas, 1,20-25 MHz, con el esfuerzo de intentar alcanzar la homogenización. Esto permite diversos beneficios a la tecnología dependiendo de las necesidades geográficas, tales como acceso asequible a internet en zonas rurales, acceso a redes metropolitanas y áreas sub-urbanas.

- *Seguridad*. Las características que se proporcionan para una red inalámbrica móvil son muy buenas, ya que utilizan autentificación basado en EAP, cifrado basado en AES-CCM, envío de mensajes basado en CMAC y HMAC para el control de protección. Existe apoyo para la seguridad con dispositivos SIM con certificados digitales.

- *Movilidad*. WiMax es compatible con sistemas de entrega optimizada con latencia inferior a 50 ms para garantizar aplicaciones en tiempo real, como puede ser VoIP, sin degradación de los servicios.

Esta tecnología de acceso transforma las señales de voz y datos en ondas de radio dentro de la citada banda de frecuencias. Está basada en la modulación OFDM (*Orthogonal Frequency Division Multiplex*), con 256 sub-portadoras que pueden llegar a cubrir un área de 50 kilómetros permitiendo la conexión sin línea de visión directa, es decir, con obstáculos interpuestos. Con capacidad para transmitir datos a una tasa de hasta 75 Mbps y con una eficiencia espectral de 5.0 bps/Hz puede dar soporte a miles de usuarios con una variedad de canales 1.5 MHz a 20 MHz.

WiMax se sitúa en un rango intermedio de cobertura entre las tecnologías de acceso de corto alcance y ofrece velocidades de banda ancha para un área metropolitana; a su vez, WiMax Forum define a la tecnología como referencia de una red IP. Los servicios *end-to-end* son encargados sobre una arquitectura IP que cuenta con protocolos basados en IP.

En la Figura 3 se puede observar cómo las estaciones subscriptoras y las estaciones móviles se conectan a las estaciones base (BS), y cómo se puede establecer la conexión entre ambas bases mediante estaciones bases. A su vez se puede observar cómo la conexión entre BS no las especifica el estándar.

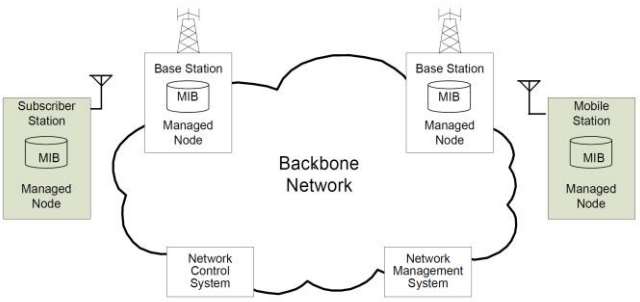


Figura 3. Modelo de referencia de las redes WiMax.

La escalabilidad de dispositivos y las velocidades de transmisión que soporta esta tecnología son elevadas, permitiendo aplicaciones WSN con bajo tiempo de respuesta y con gran número de dispositivos. Sin embargo, los grandes inconvenientes de este tipo de tecnología inalámbrica radican en el consumo de energía, dado que son comunicaciones muy pesadas para cubrir grandes distancias.

Además, al igual que WiFi, los dispositivos no pueden realizar ahorro energético permaneciendo en estados de bajo consumo ya que pierden su configuración, lo que complica la consecución de eficiencia energética en la red. Se han realizado estudios para WiMax donde el consumo es estimado entre 500 mW y 2 W cuando usan varios canales para la transmisión de datos.

c. Bluetooth

En 2002 se definió la primera versión del estándar IEEE 802.15.1 para WPAN (*Wireless Personal Area Network*), en la cual se basó la especificación Bluetooth v1.1. Este estándar incluye el acceso al medio de comunicación y las especificaciones de la capa física. La última actualización de la norma se basa en la especificación Bluetooth v5.0 que se publicó a finales del 2016.

El grupo de investigación encargado del estándar votó por unanimidad no realizar más versiones e interrumpir las relaciones con Bluetooth SIG, quien hoy en día se ha encargado realizar la nueva versión. Afirman que consiguen duplicar la velocidad, mejor fiabilidad y rango de cobertura, y aseguran que se multiplicará por 8 la tasa de datos.

Bluetooth es un sistema de enlace mediante radiofrecuencia con el fin de evitar los tediosos cables en las redes PAN (*Personal Area Network*). En la

actualidad existen numerosas aplicaciones que utilizan dicha tecnología, teléfonos móviles, ordenadores portátiles, mandos a distancias, etc…, con el fin de conectar dispositivos entre sí transfiriendo datos. La comunicación opera en la banda ISM a 2.4 GHz y para evitar las interferencias se utilizan técnicas de saltos en frecuencias (*frequency hop*).

Según el estándar, se aplican en la comunicación *slots* de tiempo de 625 μs cada uno. Con la intención de emular una transmisión *full-duplex* se utiliza una trama TDD (*Time Division Duplex*) y dentro del canal la información se intercambia por medio de paquetes.

El estándar IEEE 802.15.1 llega a soportar un canal de datos asíncronos capaz de tener tasas de velocidades hasta 2 Mbps para datos. Sigue utilizando tres canales sincronizados simultáneamente de voz donde estos datos llegan a tener una velocidad de 128 kbps. La especificación establece una unidad de control que proporcione la conexión *peer-to-peer*, o una conexión punto a multipunto. En este último caso el canal se comparte entre las unidades que se conecten entre sí.

Las redes que se montan son conocidas como *PICONET*, la cual se da cuando dos o más dispositivos comparten un mismo canal. Un dispositivo debe trabajar como administrador (*master*) de la *piconet*, mientras el resto trabaja como esclavos (*slaves*). El estándar limita las *piconet* a siete esclavos.

Las topologías que se pueden montar son gracias a una tecnología conocida como *Scatternet*. En este caso la topología restringida a un administrador con siete esclavos puede verse ampliada, ya que un esclavo puede ser maestro de otra red.

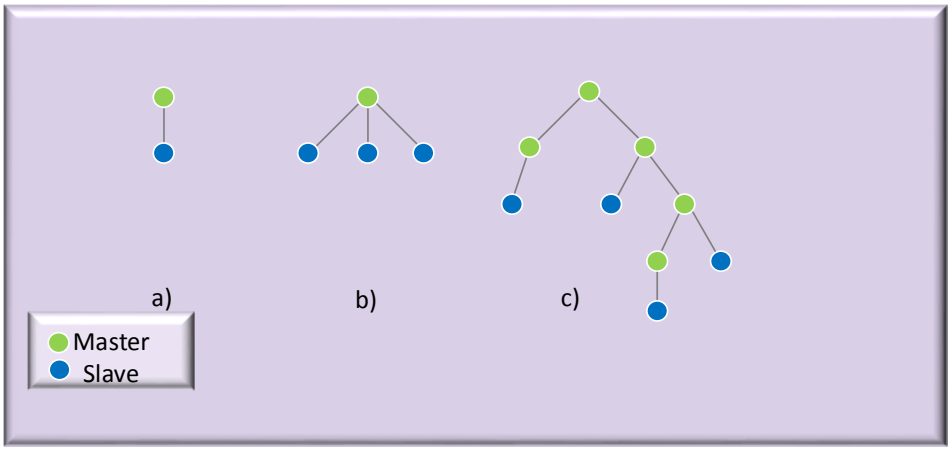


Figura 4. Tipos de redes Piconet a) Operación con un solo esclavo, b) Multi-Esclavo, c) Operación tipo Scatternet

Bluetooth se puede considerar una buena tecnología para fines WSN; sobre todo la especificación BLE (*Bluetooth Low Energy*) es idónea para este tipo de redes. Posee grandes ventajas, sobre todo en eficiencia para la transmisión de datos gracias a las características de saltos en frecuencia que hace que los enlaces sean muy robustos.

Una deficiencia de BR/EDR (*Basic Rate/Enhanced Data Rate*) es que los dispositivos no pueden permanecer en estados de bajo consumo, o en caso contrario, aumentaría de forma considerable la latencia, ya que cuando un nodo se asocia a otro dentro de una red tarda un tiempo aproximado de unos 20 segundos si es la primera vez, o 3 segundos si ya ha realizado el *pairing* con su dispositivo maestro. Otra característica negativa es el número de nodos ilimitado que puede formar la red realizando *scatternet*, aunque esta técnica degrada las características de sincronización y QoS de la red.

Por otro lado, las modificaciones realizadas en la capa física y MAC de BLE han logrado superar la limitación de tiempo de *wake up* que tenía la versión BR/EDR. Esta comunicación de ultra-bajo consumo permite una topología en estrella con un número limitado de nodos.

d. IEEE 802.15.4

El estándar 802.15.4 nació con la idea de desarrollar una tecnología de baja tasa de transmisión de datos, bajo coste y muy bajo consumo. Inicialmente los fabricantes de comunicaciones inalámbricas se dedicaron a utilizar una tecnología propia, lo que trajo diversos problemas de interoperabilidad entre varios fabricantes. Debido a este problema surgió la necesidad de una comunicación estandarizada que pudiera proporcionar compatibilidad de productos entre distintos fabricantes.

El estándar IEEE 802.15.4 define las capas físicas y MAC para redes de baja tasa de datos y bajo consumo, por lo que el diseño de este estándar se encuentra definido principalmente para aplicaciones WSN.

La norma ofrece una serie de características:

- Flexibilidad, o facilidad de integración en la red de un gran número de nodos (635536).

- Bajo coste, debido a que posee unos esquemas sencillos de modulación.

- Bajo consumo, tanto en las comunicaciones, como en los diversos modos de funcionamiento en bajo consumo.

En la Figura 5, se muestra la arquitectura que ofrece el estándar en [IEE06]:

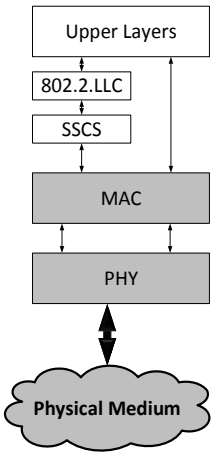


Figura 5. Arquitectura OSI de IEEE 802.15.4.

En una red basada en el estándar 802.15.4 existen una serie de dispositivos que la norma denomina FFD (*Full Function Devices*) y RFD (*Reduced Function Devices*). Una red incluye dispositivos FFD operando como coordinadores, y RFD cuya función es despertarse tras periodos de tiempo o eventos y enviar datos, sin necesidad de realizar ninguna otra gestión sobre los paquetes de información de otro nodo o sobre asociaciones de otros dispositivos a él.

El desarrollo de la norma se enfoca para las redes PAN de cortas distancias. Al igual que Bluetooth, IEEE 802.15.4 permite que dispositivos inalámbricos como PC's, PDA's, teléfonos, sensores y actuadores puedan comunicarse e inter-operar unos con otros. Para que esto pueda suceder, en la norma se definen los *189therne*, que se encargan de transferir los datos que reciben del nodo coordinador de la red, a otra tecnología de comunicación.

Se trata de un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con bajas tasas de datos (*LR-WPAN, Low-Rate Wireless Personal Area Network*). El propósito que persigue la norma es definir los niveles básicos para dar servicio de comunicación entre dispositivos con bajo coste y baja velocidad. En la Tabla 2.3 se presentan los valores más significativos.

Banda de Frecuencias-Rango de transmisión de datos	868 MHz-20Kbps 915MHz-40Kbps 2.4GHz-250Kbps
Alcance	10-20m
Asociación de un nodo a la red.	<15ms
Canales	868/915 MHz: 1/10canales 2.4GHz: 16 canales
Modos de direccionamiento	64 bits IEEE extended address 16 bits short address
Canal de Acceso	CSMA/CA
Seguridad	128 AES
Escalabilidad	Más de 65000 dispositivos

Tabla 3 Características de la tecnología IEEE 802.15.4.

La norma presenta las siguientes topologías básicas:

Formación de una red en estrella: en esta red el coordinador es quien gobierna la comunicación de los dispositivos. El coordinador de la red PAN debe ser un dispositivo FFD que puede comunicarse con dispositivos RFD; sin embargo los dispositivos RFD no se pueden comunicar entre ellos. En la Figura 6 se muestra la topología y como se establece el flujo de datos entre el coordinador de la red y los dispositivos ED (*End Devices*).

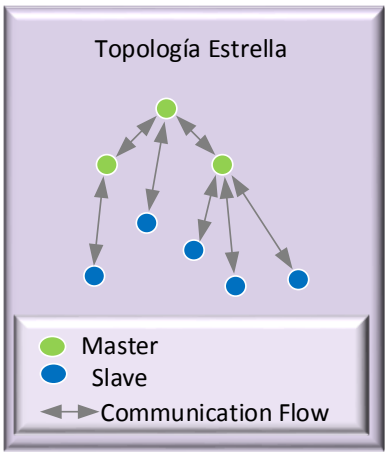


Figura 6. Topología estrella.

Formación de una red *peer-to-peer*: en una topología punto a punto, cada dispositivo puede ser capaz de comunicarse con cualquier otro dispositivo dentro de su radio de cobertura. En este tipo de redes, un nodo es designado como nodo coordinador de la red siendo el primer dispositivo que se comunique por el canal y debe imponer las restricciones para formar la red. Es evidente que todos estos dispositivos son FFD, y la red que se monta es conocida como *mesh* (malla), como se puede observar en la Figura 7.

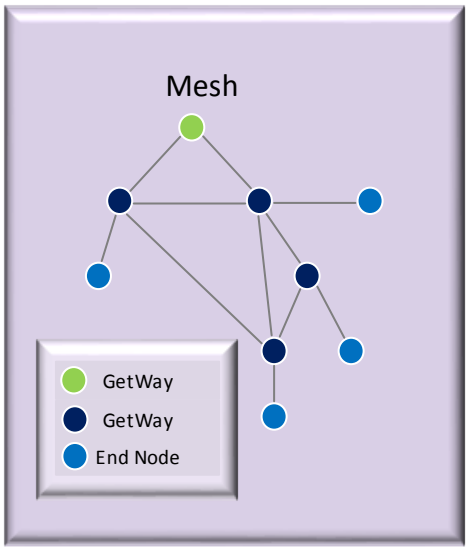


Figura 7. Topología Mesh.

El estándar de comunicaciones establece dos métodos para sincronizar los datos que navegan por la red:

Modo *Beacon-enabled*: Cuando se habilita el modo de balizamiento o con balizas, el coordinador de la red utiliza supertramas para establecer la comunicación dentro de la red. El formato de la supertrama lo define el *PAN Coordinator*, y dicho formato se envía periódicamente dentro de una trama llamada *beacon*, al resto de dispositivos asociados a él. El acceso de los nodos en el interior de la supertrama se realiza mediante CSMA/CA Slotted (*Carrier Sense Multiple Access with Collision Avoidance*).

Modo *Non-Beacon-enabled*: En este modo de funcionamiento los dispositivos simplemente envían sus datos mediante los mecanismos de acceso al medio CSMA/CA. En este tipo de modo no se utilizan supertramas.

Anexo 2. Estudio de Tecnologías Cableadas

Anexo 2. 1. Introducción

Se pretende realizar un estudio de dos comunicaciones conocidas en robótica, y seleccionar la más idónea para los estudios que hay que realizar. El requisito imprescindible para la selección del protocolo más adecuado es la capacidad de trabajar el bus de comunicaciones en alta impedancia. Este requisito evita fallos a la hora de realizar ataques hardware.

Anexo 2. 2. El bus SPI

El protocolo de comunicaciones SPI es un estándar de comunicaciones que se basa en 4 señales. Estas señales se conectan físicamente a los pines de salida y control del microcontrolador que realice las labores de maestro del bus.

Las señales se identifican como:

- SCLK: Es el pin encargado de marcar el pulso de sincronización.
- MOSI (*Master Output Slave Input*): Salida de información desde el maestro hasta el esclavo seleccionado

- MISO (*Master Input Slave Output*): Salida de información desde el esclavo hasta el maestro.

- SS/Select: Pin que activa al esclavo para la escucha de datos desde el maestro.

La sincronización de la información tiene que seguir una serie de pasos: inicialmente el maestro del bus selecciona un esclavo por el pin *ss*; seguidamente el maestro pone en marcha el reloj y envía un dato sincronizado con el reloj; cada bit elegido está sincronizado en la subida de reloj como se muestra en la siguiente figura.

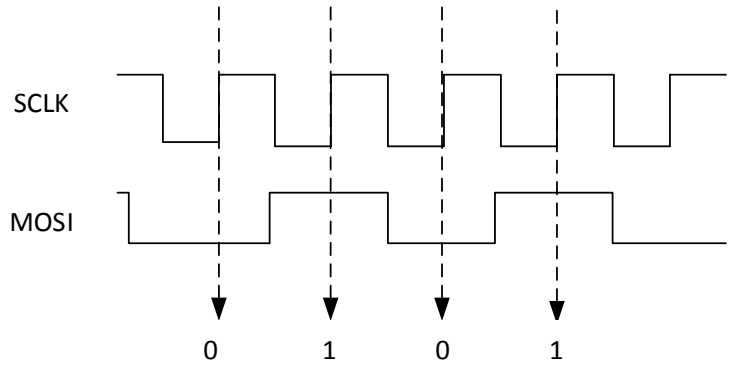


Figura 8. Sincronización del protocolo SPI.

Se trata de un estándar de comunicaciones *full-duplex,* permitiendo la comunicación en ambos sentidos a la vez. El maestro utiliza la línea MOSI para transmitir la información al esclavo y éste último realiza la misma acción por la línea de datos MISO.

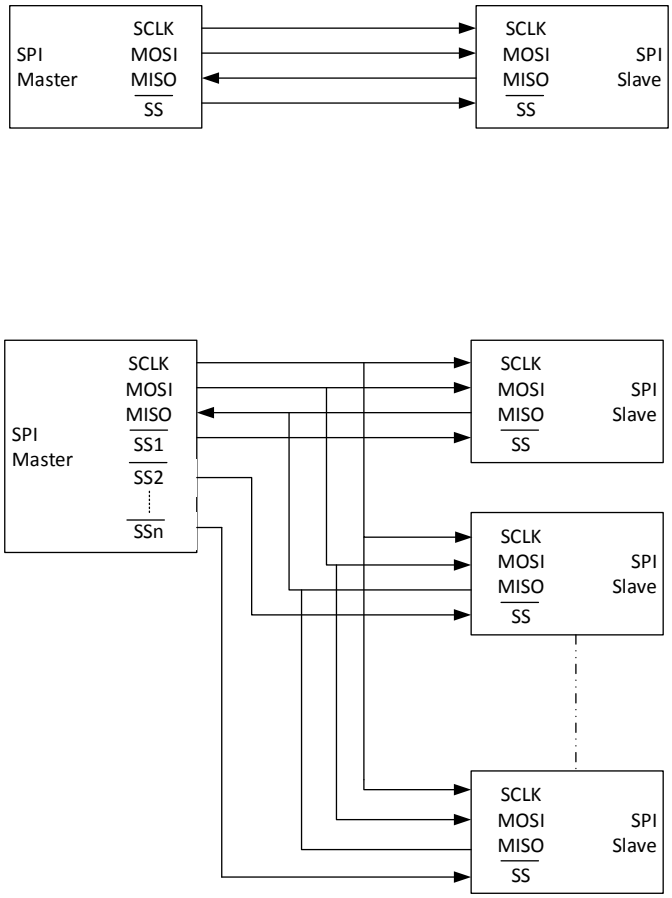


Figura 9. Sistema de conexionado SPI. Arriba punto a punto; abajo punto a multipunto.

Anexo 2. 3. El bus I2C

El bus I2C se basa en dos señales generalmente llamadas SDA (línea de datos) y SCL (línea de reloj) y eventualmente una señal de tierra (todos los módulos deben tener la misma referencia de voltaje). Todos los módulos están conectados a las mismas líneas del bus, como se muestra en la Figura 10, y en general, cualquier módulo permite la conexión de un nuevo módulo al bus. Existen dos tipos de módulos: un módulo principal que genera la señal SCL y controla las transmisiones; y uno o varios módulos esclavos que serán la fuente (operaciones de lectura) o destino (operaciones de escritura) de la información. La señal SDA es generada tanto por el maestro como por los módulos esclavos cuando se produce la comunicación. Los valores lógicos en el protocolo I2C son:

tierra para el nivel bajo nivel y alta impedancia para el nivel alto. La utilización de alta impedancia implica que los diferentes módulos no tienen que utilizar la misma fuente de polarización y, por lo tanto, los módulos que necesitan alta tensión se pueden conectar al mismo bus que los módulos necesarios de baja tensión.

Las características iniciales de protocolo I2C (un protocolo abierto y el uso de alta impedancia para el nivel alto) representan una alta vulnerabilidad en el sistema. En primer lugar, un protocolo abierto permite la conexión de cualquier módulo (autorizado o no autorizado) al bus. En segundo lugar, la utilización de alta impedancia como nivel lógico permite sobrescribir la información usando niveles bajos sin ninguna consecuencia en la señalización del protocolo.

El comportamiento de una operación de escritura usando un protocolo I2C se ilustra en la Figura 10(b) y está ampliamente descrito en la literatura científica y técnica ([Ham13]).

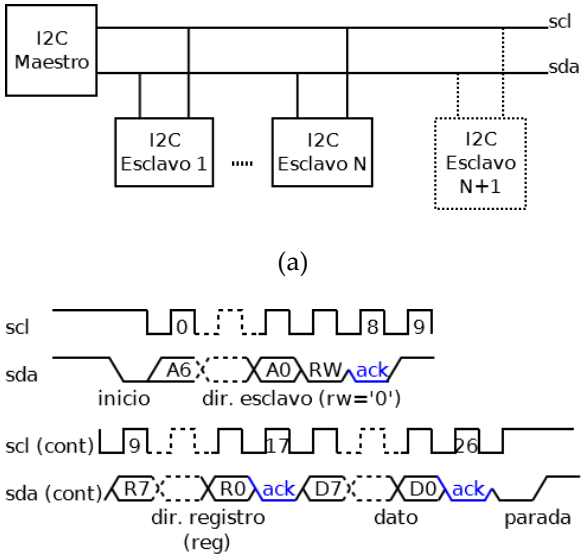


Figura 10. (a) Arquitectura de un sistema basado en el protocolo I2C; (b) señales del procedimiento de escritura en el protocolo I2C.

Referencias

[Alk06] Alkalai, L., CHAU, S. N., Tai, A. T. "*Fault-tolerant communication channel structures*". U.S. Patent No 7,020,076, 2006.

[And96] Anderson, R., Kuhn, M. "Tamper Resistence a Cautionary Note", *2nd USENIX Workshop on Electronic Commerce Proceeding*, 1-11, 1996.

[Aro08] Arora, A., Telang, R., & Xu, H. "Optimal policy for software vulnerability disclosure". *Management Science*, 54(4), 642-656, 2008.

[Are06] Arena, P., Buscarino, A., Fortuna, L., Frasca, M. "Separation and synchronization of piecewise linear chaotic systems". *Phys. Rev. E* 2006, 74, 026212.

[Ash90] Ashenden, P. J., 1990. *The VHDL cookbook*. Department of Computer Science, University of Adelaide.

[Bar02] Bartsch, E. R., Fisher, C. W., France, P. A., Kirkpatrick, J. F., Heaton, G. G., Hortel, T. C., Stigall, J. R., 2002. U.S. Patent No. 6,459,955. Washington, DC: U.S. Patent and Trademark Office.

[Bas04] Basu, P., & Redi, J. "Movement control algorithms for realization of fault-tolerant ad hoc robot networks". *Network, IEEE*, 18(4), 36-44, 2004.

[Bas17] Basys 2 Reference Manual, 2017

[Bog13] Bogue, R. "Sensors for condition monitoring: A review of technologies and applications". *Sens. Rev*. 2013, 33, 295–299.

[Bro81] Brown, D. W. "A state-machine synthesizer—SMS". *In Proceedings of the 18th Design Automation Conference*, 301-305, 1981.

[Bru05] Bruschi, D., Cavallaro, L., Lanzi, A. "Replay Attack in TCG Specification and Solution". *In Proceedings of the 21st Annual Computer Security Applications Conference, IEEE Computer Sociality*, 127–137, 2005.

[Bul01] Bulusu et al., "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems". *ISCTA* 2001, Ambleside, U.K., July 2001.

[Cañ14] Cañas, N., Hernández, W., González, G., & Sergiyenko, O. "Controladores multivariables para un vehículo autónomo terrestre: Comparación basada en la fiabilidad del software". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 11(2), 179-190, 2014.

[Car10] Carlos García Arano. *Impacto de la Seguridad en Redes Inalámbricas de Sensores IEEE 802.15.4*. Universidad Complutense de Madrid. 2010.

[Cas04] M. Castillo-Effen, D.H. Quintela, R. Jordan, W. Westhoff, W. Moreno. "Wireless sensor networks for flash-flood alerting". *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits, and Systems, Dominican Republic*, 2004.

[Chi09] Chiara Buratti, Andrea Conti, Davide Dardari and Roberto Verdone. "An Overview on Wireless Sensor Networks Technology and Evolution". *Sensors*, 9(9), 6869-6896, 2009.

[Cha06] Chapman, K., 2006. Initial Design for Spartan-3E Starter Kit (LCD Display Control). Xilinx Ltd 16th February.

[Cue04] Cuesta, F., Gómez-Bravo, F., & Ollero, A. Parking maneuvers of industrial-like electrical vehicles with and without trailer. *Industrial Electronics, IEEE Transactions on*, 51(2), 257-269, 2004.

[Duj13] Du, J., Wu, Y.C. "Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis". *IEEE Trans. Wirel. Commun.* 2013, 12, 5908–5917.

[Dut11] Dutertre, J.M.; Fournier, J.J.; Mirbaha, A.P.; Naccache, D.; Rigaud, J.B.; Robisson, B.; Tria, A. "Review of fault injection mechanisms and consequences on countermeasures design". *In Proceedings of the 6th IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Athens, Greece, 6–8 April 2011; pp. 1–6.

[Fer11] Ferruz, J., Vega, V. M., Ollero, A., & Blanco, V. "Reconfigurable control architecture for distributed systems in the HERO autonomous helicopter". *Industrial Electronics, IEEE Transactions on*, 58(12), 5311-5318, 2011.

[Fuk04] Fukuhara, R.; Day, L.; Luong, H.H.; Rasmussen, R.; Chau, S.N. *I2C Bus Protocol Controller with Fault Tolerance*. U.S. Patent 6,728,908, date month 2004.

[Gao05] T. Gao, D. Greenspan, M. Welsh, R.R. Juang, A. Alm. "Vital signs monitoring and patient tracking over a wireless network". *Proceedings of the 27th IEEE EMBS Annual International Conference*, 2005.

[Gar07] Garcia-Cerezo, A., Mandow, A., Martinez, J. L., Gómez-de-Gabriel, J., Morales, J., Cruz, A., Seron, J. "Development of ALACRANE: A mobile robotic assistance for exploration and rescue missions". *In Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pp 1-6, 2007.

[Gom01] Gómez-Bravo, F., Cuesta, F., & Ollero, A. "Parallel and diagonal parking in non-holonomic autonomous vehicles". *Engineering applications of artificial intelligence*, 14(4), 419-434, 2001.

[Gom15] Gomez-Bravo, F., Naharro, R. J., García, J. M., Galán, J. G., & Raya, M. S., 2015. "Sobre la vulnerabilidad de los robots móviles frente a los ataques hardware". *XXXVI Jornadas de Automática*, pp. 358-365.

[Gom16] Gomez-Bravo, F., Naharro, R. J., García, J. M., Galán, J. G., & Raya, M. S. Hardware Attacks on Mobile Robots: I2C Clock Attacking. *In Robot 2015: Second Iberian Robotics Conference*, pp. 147-159, 2015.

[Gom17] Gomez-Bravo, F.; Medina García, J.; Jiménez Naharro, R.; Gómez Galán, J.A.; Sánchez Raya, M. "Experimental Platform for Studying Hardware Vulnerabilities on Mobile Robots: I2C Bus, a Case of Study". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 14, no 2, pp. 205–216, Abril 2017.

[Gon04] Gonzalo Álvarez Marañon, Pedro Pablo Pérez García. *Seguridad en Redes Inalámbricas Wifi*. Departamento de Tratamiento de la Información y Codificación. Instituto de Física Aplicada. Consejo Superior de Investigaciones Científicas de España. 2004.

[Gou12] Guo, P.; Infield, D.; Yang, X. "Wind turbine generator condition monitoring using temperature trend analysis". *IEEE Trans. Sustain. Energy*, 3, 124–133, 2012.

[GuF15] Gu, F.; Wang, T.; Alwodai, A.; Tian, X.; Shao, Y.; Ball, A.D. "A new method of accurate broken rotor bar diagnosis based on modulation signal bispectrum analysis of motor current signals". *Mech. Syst. Signal Process*., 50, 400–413, 2015.

[Hagai06] Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall and C. Whelan. "The Sorcerer's Apprentice Guide to Fault Attacks", *Proceedings of the IEEE*, vol. 94, no. 2, 2006.

[Ham13] Hamblen, J.O., van Bekkum, G.M.E. "An Embedded Systems Laboratory to Support Rapid Prototyping of Robotics and the Internet of Things", *Education, IEEE Transactions on*, 56 (1), 121-128, 2013.

[HC-06] Tutorial Módulo Bluetooth HC-06

[Hee11] Heelan, S. "Vulnerability detection systems: Think cyborg, not robot". *IEEE Security & Privacy*, (3), 74-77, 2011.

[Hei99] Heinzelman, W. R., Kulik, J., and Balakrishnan, H. "Adaptative protocols for information dissemination in wireless sensor networks", *Proceedings of the ACM MobiCom'99*, 174-185, 1999.

[Hou12] Hou, L.; Bergmann, N.W. "Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis". *IEEE Trans. Instrum. Meas.*, 61, 2787–2798, 2012.

[Hua03] Huang, A. "*Hacking the Xbox: An Introduction to Reverse Engineering*," No Starch Press, 2003, ISBN: 978-1593270292.

[Hua15] Huang, Q.; Tang, B.; Deng, L. "Development of high synchronous acquisition accuracy wireless sensor network for machine vibration monitoring", *Measurement*, 66, 35–44, 2015.

[Ian02] Ian F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and Erdal Cayirci. "A Survey on Sensor Networks". *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[I10] Tehranipoor, M., Koushanfaar, F. "A Survey of Hardware Trojan Taxonomy and Detection". *IEEE Design and Test of Computers*, 27(1), 10-25, 2010.

[IEE06] IEEE 802.15.4 MAC 2006

[IEE16] The IEEE 802.16 Working Group on Broadband Wireless Access Standards. http://ieee802.org/16/.

[IEE17] www.IEEE.org

[ISO10] ISO 10816–1. *Evaluation Standard for Vibration Monitoring. International Organization for Standardization*, Geneva, Switzerland, 1995.

[Jar08] Jardón, A., Giménez, A., Correal, R., Martinez, S., & Balaguers, C. "Asibot: Robot portátil de asistencia a discapacitados. Concepto, arquitectura de control y evaluación clínica". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 5(2), 48-59, 2008.

[Jim12] R. Jimenez, G. Feria, M. Sanchez-Raya, J. Galán and F. Gómez. "FPGA Implementation of Hardware Countermeasures". *Programmable Logic (SPL), 2012 VIII Southern Conference On.* 2012.

[Jim13] Jiménez-Naharro, R.; Gómez-Galán, J.A.; Sánchez-Raya, M.; Gómez-Bravo, F.; Pedro-Carrasco, M. "Design and implementation of a new real-time frequency sensor used as hardware countermeasures". *Sensors*, 13, 11709–11727, 2013.

[Jen05] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal. "Wireless sensor network survey", *Journal Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 12, pp. 2292- 2330, 2008.

[Kar13] Karaklajic, D, Verbauwhede, I., "Hardware Designer's Guide to Fault Attacks". *IEEE Transactions on Very Large Scale Integration Systems*, 21, 2295-2306, 2013.

[Kew06] Kewei Sha, Weisong Shi. "Using Wireless Sensor Networks for Fire Rescue Applications: Requirements and Challenges". *Electro/information Technology, 2006 IEEE International Conference on*, 2006.

[Kim11] Kim, J.S., Lee, J., Serpedin, E., Qaraqe, K. "Robust clock synchronization in wireless sensor networks through noise density estimation". *IEEE Trans. Signal Process*. 2011, 59, 3035–3047.

[Köm99] Kömmerling, O.; Kuhn, M.G. "Design Principles for Tamper-Resistant Smartcard Processors". *Smartcard* 1999, 99, 9–20.

[Kou07] Koubaa, A., Cunha, A., and Alves, M. "A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks", *19th Euromicro Conference on Real-Time Systems*, 125-135, 2007.

[Lad04] Ladd, A. M., Bekris, K. E., Rudys, A. P., Wallach, D. S., & Kavraki, L. E. "On the feasibility of using wireless Ethernet for indoor localization". *IEEE Transactions on Robotics and Automation*, 20(3), 555-559, 2004.

[Lee15] Lee, J.; Moon, S.; Jeong, H.; Kim, W.S. "Robust diagnosis method based on parameter estimation for an interturn short-circuit fault in multipole PMSM under high-speed operation". *Sensors*, 15, 29452–29466, 2015.

[Lim03] Lima, P., Ribeiro, M. I., Custodio, L., & Santos-Victor, J. "The RESCUE Project-Cooperative Navigation for Rescue Robots". *Proc. of ASER*, 3, pp. 13-15, 2003.

[LiX12] Li, X.; Bleakley, C.J.; Bober, W. "Enhanced beacon-enabled mode for improved IEEE 802.15.4 low data rate performance". *Wireless Network*, 18, 59–74, 2012.

[Lor04] K. Lorincz, D. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, S. Moulton. "Sensor networks for emergency response: challenges and opportunities". *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 16-23, 2004.

[LuB09] Lu, B.; Gungor, V.C. "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks". *IEEE Trans. Ind. Electron*. 2009, 56, 4651–4659.

[Mac93] J. MacLellan, S. Lam, and X. Lee, "Resitian in door RF Channel Characterization", *43rd IEEE VTC*, 1993, pp. 210-213.

[Mag17] Maggi, F., Quarta D., Pogliani, M., Polino, M., Zanchettin, A.M., Zanero, S. "*Rogue Robots: Testing the Limits of an Industrial Robot Security*". TrendLabs Research Paper, 2017.

[Mar07] Marques, C.; Cristóvão, J.; Alvito, P.; Lima, P.; Frazão, J.; Ribeiro, I.; Ventura, R. "A search and rescue robot with tele-operated tether docking system". *Ind. Robot 2007*, 34, 332–338.

[Mar13] Martalo, M., Buratti, C., Ferrari, G., and Verdone, R. "Clustered IEEE 802.15.4 Sensor Networks with Data Aggregation: Energy Consumption and Probability of Error", *IEEE Wireless Communications Letters*, 2(1), 70-73, 2013.

[McL08] McLoughlin, I. "Secure embedded system: The threat of reverse engineering". *In Proceedings of the 14th International Conference on Parallel and Distributed Systems*, Melbourne, Australia, 8–10 December 2008; pp. 729–736.

[Min04] Minguez, J., Montesano, L., Montano, L. "An architecture for sensor based navigation in realistic dynamic and troublesome scenarios". *In Proceedings of the Intelligent Robots and Systems International Conference on*, vol. 3, pp. 2750-2756, 2004.

[Mor12] Moreno, H. A., Saltaren, R., Carrera, I., Puglisi, L., & Aracil, R. "Índices de desempeño de robots manipuladores: una revisión del estado del arte". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 9(2), 111-122, 2012.

[Mor09] Morales, J.; Martínez, J.L.; Martínez, M.A.; Mandow, A. "Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner". *J. Adv. Signal Process*. 2009, doi:10.1155/2009/935237.

[Mot17] Motor Controller MD23. Available online: http://www.robot-electronics.co.uk/htm/md23tech.htm.

[Nak15] Nakhaeinia, D.; Payeur, P.; Hong, T.S.; Karasfi, B. "A hybrid control architecture for autonomous mobile robot navigation in unknown dynamic environment". *In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE)*, Gothenburg, Sweden, 24–28 August 2015; pp. 1274–1281.

[Nan05] Nandi, S.; Toliyat, H.A.; Li, X. "Condition monitoring and fault diagnosis of electrical motors—A review". *IEEE Trans. Energy Convers*., 20, 719–729, 2005.

[Nob12] Nobile, C., 2012. Robots Vulnerable to Hacking. http://www.roboticsbusinessreview.com/article/robots_vulnerable_to_hacking/

[Nob17] Nobile, C. "*Robots Vulnerable to Hacking. Robotic Business Review"*. http://www.roboticsbusinessreview.com/article/robots_vulnerable_to_hacking, 2017.

[Oll94] Ollero, A., Mandow, A., Muñoz, V. F., & De Gabriel, J. G. "Control architecture for mobile robot operation and navigation. Robotics and computer-integrated manufacturing", *Robotics and Computer-Integrated Manufacturing*, 11(4), 259-269, 1994.

[Oll95] Ollero, A., Heredia, G. "Stability analysis of mobile robot path tracking. In Intelligent Robots and Systems 95". *In Proceedings of the Human Robot Interaction and Cooperative Robots. IEEE/RSJ International Conference on*, vol. 3, pp. 461-466, 1995.

[Oll99] Ollero, A., Arrue, B. C., Ferruz, J., Heredia, G., Cuesta, F., López-Pichaco, F., & Nogales, C. "Control and perception components for autonomous vehicle guidance. Application to the ROMEO vehicles". *Control Engineering Practice*, 7(10), 1291-1299, 1999.

[Oll01] Ollero, A. "*Robótica: Manipuladores y Robots Móviles*". Marcombo: Barcelona, Spain, 2001.

[opn17] www.opnet.com

[Par13] Park, J.; Jeong, W.; Lee, H.K.; Won, J. "An efficient path planning method for a cleaning robot based on ceiling vision". *In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 11–14, 2013.

[Pee17] Peeters, C.; Guillaume, P.; Helsen, J. "Vibration-based bearing fault detection for operations and maintenance cost reduction in wind energy". *Renew. Energy* 2017.

[Pri07] Prieto, J., Ramos, O., Delgado, A. "Diseño de un gene digital en FPGA y MATLAB con aplicaciones en robótica móvil". *XIII Taller Iberchip IWS-2007*, Lima, 14, 2007.

[Qia15] Qiao, W.; Lu, D. "Survey on wind turbine condition monitoring and fault diagnosis—Part II: Signals and signal processing methods". *IEEE Trans. Ind. Electron.* 62, 6546–6557, 2015.

[Raw14] Rawat, P.; Singh, D.K.; Chaouchi, H.; Bonnin, J.M. "Wireless sensor networks: A survey on recent developments and potential synergies". *J. Supercomput.* 2014, 68, 1–48.

[Ros05] Ross Anderson, Mike Bond, Jolyon Clulow and Sergei Skorobogatov. "Cryptographic processors- a survey". *Proceedings of the IEEE*, vol. 94, no. 2, 2006.

[Rui16] Ruiz-Cárcel, C.; Jaramillo, V.H.; Mba, D.; Ottewill, J.R.; Cao, Y. "Combination of process and vibration data for improved condition monitoring of industrial systems working under variable operating conditions". *Mech. Syst. Signal Process.* 2016, 66, 699–714.

[Ryb00] Rybski, P. E., Stoeter, S. A., Erickson, M. D., Gini, M., Hougen, D. F., & Papanikolopoulos, N. "A team of robotic agents for surveillance". *In Proceedings of the fourth international conference on autonomous agents*, pp. 9-16, 2000.

[Sad06] Sadler, B.M., Swami, A. "Synchronization in sensor networks: An overview". *In Proceedings of the IEEE Military Communications Conference (MILCOM)*, Washington, DC, USA, 23–25,2006; pp. 1–6.

[Sal10] Salman, N.; Rasool, I.; Kemp, A.H. "Overview of the IEEE 802.15.4 standards family of Low Rate Wireless Personal Area Network". *In Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS)*, York, UK, 19–22, 2010.

[Sha10] Shahid Raza, Thiemo Voigt. "Interconnecting WirelessHart and Legacy Hart Networks". *Distributed Computing in Sensor Systems Workshops (DCOSSW), 6th IEEE International Conference on*, 2010.

[Sha15] Shariff, F.; Rahim, A.N.; Ping, W.H. "Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system". *Expert Syst. Appl.*, 42, 1730–1742, 2015.

[She17] Sheppard, B., Thompson, T., (2017) "*Cyber Security for Robots: Scenarios for 2030*", http://www.roboticsbusinessreview.com/article/cyber_security_for_robots_scenarios_for_2030

[Sie16] Siemens Motor. Available online: https://mall.industry.siemens.com/mall/en/uk/Catalog/Product/1LA7063-4AB11 (accessed on December 15, 2016).

[Sim04] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, K. Frampton. "Sensor network-based countersniper system". *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (Sensys)*, Baltimore, MD, 2004.

[Sin13] Singh, G. "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security". *Int. J. Comput. Appl.*, 67, 33–38, 2013.

[Sun10] Sung, W.-T. "Multi-sensors data fusion system for wireless sensors networks of factory monitoring via BPN technology". *Expert Syst. Appl.*, 37, 2124–2131, 2010.

[Sun11] Sung, W.T.; Hsu, Y.C. "Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee". *Expert Syst. Appl.* 2011, 38, 4522–4529.

[Sre11] Sreenithi, V.; Selvabala, N.; Ganesh, B.A. "Implementation of Wireless sensor network based human fall detection system". *Proc. Eng.*, 30, 767–773, 2011.

[Tak06] Takeshita, T., Tomizawa, T., & Ohya, A. "A House Cleaning Robot System-Path indication and Position estimation using ceiling camera. *In SICE-ICASE International Joint Conference*, pp. 2653-2656, 2006.

[Tsy13] Tsypkin, M. Induction Motor Condition Monitoring: "Vibration analysis technique—A twice line frequency component as a diagnostic tool". *In Proceedings of the IEEE International Electric Machines & Drives Conference (IEMDC)*, Chicago, IL, USA, 12–15 May 2013.

[Ver07] Verucchi, C.J.; Acosta, G.G. "Fault detection and diagnosis techniques in induction electrical machines". *IEEE Latin Am.* Trans. 2007, 5, 41–49.

[Val15] Gómez, J. V., Vale, A., Garrido, S., & Moreno, L. "Performance analysis of fast marching-based motion planning for autonomous mobile robots in ITER scenarios". *Robotics and Autonomous Systems*, 63, 36-49, 2015.

[Wen06] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Walsh. "Deploying a wireless sensor network on an active volcano", Data-Driven Applications in Sensor Networks (Special Issue). *IEEE Internet Computing*, vol. 2, pp. 18-25, 2006.

[Wan07] Wantanabe, K., Ise, M., Onoye, T., Niwamoto, H., and Keshi, I. "An Energy-efficient Architecture of Wireless Home Network Based on MAC Broadcast and Transmission Power Control", *International Conference on Consumer Electronics, ICCE. Digest of Technical Papers*, 1-2, 2007.

[Yan09] Yang, W.; Tavner, P.J.; Wilkinson, M.R. "Condition monitoring and fault diagnosis of a wind turbine synchronous generator drive train". *IET Renew. Power Gener.* 2009, 3, 1–11.

[Yic05] J. Yick, B. Mukherjee, D. Ghosal. "Analysis of a Prediction-based Mobility Adaptive Tracking Algorithm". *Proceedings of the IEEE Second International Conference on Broadband Networks (BROADNETS)*, Boston, 2005.

[Yoo10] Yoo, S.-E.; Chong, P.K.; Kim, D.; Doh, Y.; Pham, M.L.; Choi, E.; Huh, J. "Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15". *IEEE Trans. Ind. Electron.*, 57, 3868–3876, 2010.

[Zah07] Zaher, A.S.; McArthur, S.D.J. "A multi-agent fault detection system for wind turbine defect recognition and diagnosis". *In Proceedings of the IEEE Lausanne Power Tech, Lausanne*, Switzerland, 1–5 July 2007; pp. 22–27.

[Zhe06] Zheng, J., and Lee, M. J. "A comprehensive Performance Study of IEEE 802.15.4, Sensor Network Operations", *IEEE Press, Wiley Interscience*, Chapter 4, 218-237, 2006.

[Zen08] Zen, K., Habibi, D. and Ahmad, I. "Improving Mobile Sensor Connectivity Time in the IEEE 802.15.4 Networks", *Telecommunication Networks and Applications Conference*, *ATNAC*, 317-320, 2008.

[Zha08] Zhang, F., Wang, F., Dai, B., and Li, Y. "Performance Evaluation of IEEE 802.15.4 Beacon Enabled Association Process", *22nd International Conference on Advanced Information Networking and Application – Workshops*, 541-546, 2008.

Publicaciones

Artículos en revistas internacionales

1. **Jonathan Medina-García**, Trinidad Sánchez-Rodriguez, Juan Antonio Gómez Galán, Aránzazu Delgado, Fernando Gómez-Bravo and Raúl Jiménez. "A Wireless Sensor System for Real-Time Monitoring and Fault Detection of Motor Arrays". *Sensors*, vol. 17(3), pp. 469-491, Febrero 2017.

2. Raúl Jiménez-Naharro, Fernando Gómez-Bravo, **Jonathan Medina-García,** Manuel Sánchez-Raya and Juan Antonio Gómez-Galán."A Smart Sensor for Defending against Clock Glitching Attacks on the I2C Protocol in Robotic Applications". *Sensors,* vol. 17(4), pp. 677-684, Marzo 2017.

3. F. Gómez Bravo, **J. Medina García**, R. Jiménez Naharro, J. A. Gómez Galán y M. Sánchez Raya. "Plataforma Experimental para el Estudio de la Vulnerabilidad Hradware en los Robots Móviles: el Bus I2C como Caso de Estudio". *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 14, no 2, pp. 205–216, Abril 2017.

4. C. Rubia-Marcos, **J. Medina-García**, J. Galán, D. Daza, and R. G. Carvajal. "Low Activity Mechanism for Mobile Sensor/Actuator Networks based on IEEE 802.15.4". *Wireless Personal Communications*, vol. 97, no.1, pp. 197-212, Nov. 2017.

Comunicaciones en congresos nacionales e internacionales

1. F. Gómez Bravo, Raúl Jimenez Naharro, **J. Medina García**, J. A. Gómez Galán, M. Sanchez Raya. "Sobre la Vulnerabilidad de los Robots Móviles Frente a los Ataques Hardware". *XXXVI Jornadas de Automática*. Bilbao, 2-4 Septiembre 2015.

2. F. Gómez Bravo, Raúl Jimenez Naharro, **J. Medina García**, J. A. Gómez Galán, M. Sanchez Raya. "Hardware Attacks on Mobile Robots: I2C Clock Attacking". *Second Iberian Robotics Conference*. Lisboa, 19-21 Noviembre 2015.

3. F. Gómez-Bravo, M. Sánchez-Raya, J.A. Gómez-Galán, R. Jiménez-Naharro, M. J. Aznar, R. López de Ahumada, **J. Medina-García**, J.M. Martín Ramos, M. Pedro Carrasco. "Desarrollo de vehículos autónomos recolectores de Fresas", *Jornadas Nacionales de Robótica, Spanish Robotics Conference*. Valencia, 8-9, Junio 2017.