

Capítulo 2: Descripción de la tecnología empleada

1. Qfix:

1.1. Descripción del fabricante:

Qfix es una empresa de origen alemán fundada en 1998 cuyo principal nicho de mercado está orientado para ámbitos educativos con el objetivo de la enseñanza de electrónica y robótica a un público joven. A lo largo de su desarrollo como empresa fue abarcando otras ramas de mercado como robótica a nivel educativo de enseñanzas superiores y a particulares con un conocimiento de electrónica y programación avanzados, habiéndose establecido en 2006 como proveedora de material didáctico para algunas universidades Alemanas. En la actualidad está promoviendo una incursión dentro de la robótica profesional con robot todoterreno, módulos de dirección omnidireccionales y escáneres láser.

El producto ofrecido es similar en concepto y en calidad al de otras marcas más conocidas como Lego MindStorm pero con un ahorro considerable de los productos Qfix frente a los productos de sus competidores. Los elementos que componen los sets robóticos distribuidos por Qfix están fabricados con electrónica de consumo con un núcleo central basado en microcontroladores Atmel, esta cualidad permite una alta universalización de sus productos facilitando la compatibilidad de otros dispositivos vendidos por el fabricante o realizados por el usuario.

El software necesario para la programación de las placas controladoras de los kit

robóticos podría realizar utilizando cualquier compilador con librerías propias para programación de núcleos basados en arquitectura AVR; no obstante, el fabricante distribuye gratuitamente un software a modo de entorno de desarrollo que aglutina un editor de texto para los programas con un reconocedor de palabras reservadas básico que otorga al texto de un sangrado y coloreado que evade la arcaica monotonía permitiendo una comprensión rápida del código. El lenguaje de programación para este software está orientado a objetos y como lenguaje de programación se utiliza C++ de forma que pueden ser utilizadas librerías genéricas de C++ o las propias desarrolladas por el usuario.

1.2. Dispositivos disponibles:

Qfix dispone de los siguientes modelos de kit de robots a nivel educacional:

- Minibot:

Qfix presenta a Minibot como modelo el más asequible que permite el manejo de una tecnología robótica profesional. El kit contiene una plataforma de dirección diferencial con dos motores eléctricos y caja de reducción, una placa controladora con conexión USB y un sensor de blancos que permite la detección y movimiento a lo largo de una línea negra. Adicionalmente se puede añadir una pantalla LCD y hasta ocho sensores a la placa. La placa controladora utiliza un microcontrolador Atmega32 y las comunicaciones habilitadas son mediante USB y mediante puerto I²C.

- Crash-Bobby:

Qfix presenta a Crash-Bobby como un kit robótico perfecto para comenzar a trabajar más profundamente el mundo de la robótica. Contiene una plataforma con tracción diferencial de aluminio, tres sensores de proximidad y una placa controladora idéntica a la de Minibot.

- Omnibot:

Qfix presenta a Omnibot como un kit para nivel educacional tope de gama. Consiste en una base de aluminio circular de 21cm con tres ruedas omnidireccionales tractoras. Dispone también de dos sensores de distancia y uno de contraste de color.

1.3. Justificación de la elección y características del dispositivo elegido:

Puesto que el objetivo principal del proyecto es sentar las bases para crear un escenario con más elementos y verificar la compatibilidad con distintas tecnologías se ha decidido adquirir un robot de bajas prestaciones pero que posibiliten la interconexión mediante algún protocolo de comunicación.

El robot base elegido sobre el que posteriormente se añadirán nuevos elementos se muestra en la siguiente figura:



Ilustración 5: Imagen de MiniBot de Qfix

El elemento más importante de este kit robótico es la placa controladora de la que cabe destacar la conexión I²C para la comunicación con otras placas y la conexión para los motores que implementa por defecto alimentación PWM.

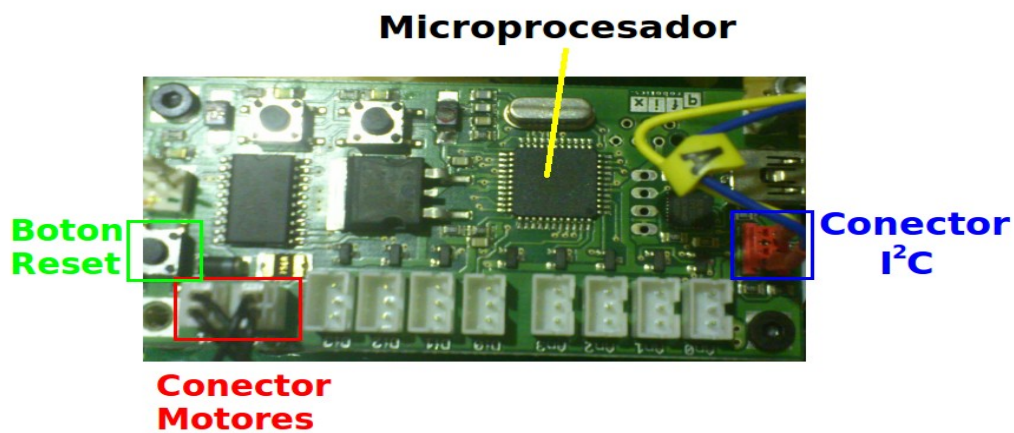


Ilustración 6: Placa controladora de MiniBot

2. Libelium:

2.1. Descripción del fabricante:

Libelium es una empresa fundada en 2006 como una compañía Spin Off de la Universidad de Zaragoza cuyo principal nicho de mercado está orientado al diseño y fabricación de tecnología hardware para la implementación de redes sensoriales inalámbricas y redes malladas para que integradores de sistemas, ingenierías y consultorías puedan llevar a cabo soluciones fiables a usuarios finales con el mínimo tiempo de desarrollo y salida al mercado.

Su diseño se caracteriza por una gran modularidad que permite integrar al sistema únicamente los módulos que requeridos pudiendo éstos ser cambiados o ampliados según las necesidades. El aprendizaje de esta plataforma es rápido y se encuentra cubierto por una extensa documentación tanto en forma de manuales como de foro que pone en contacto a toda la comunidad de desarrolladores.

2.2. Dispositivos disponibles:

Actualmente este fabricante dispone de dos familias de dispositivos cuyas características se detallan a continuación:

» WaspMote:

Libelium presenta este dispositivo como una mota sensorial de bajo consumo que permite la creación de redes sensoriales inalámbricas en la que se pueden integrar mas de 50 sensores diferentes. Adicionalmente dispone de un dispositivo denominado GateWay cuya finalidad está orientada a escuchar las transmisiones en la red o la información que le sea enviada.

Esta Mota fabricada por libelium tiene integradas las siguientes características:

- Microcontrolador Atmega 1281 con una frecuencia de trabajo de 8MHz
- Memoria de trabajo SRAM de alta velocidad de 8KB
- Memoria no volátil de rápido acceso EEPROM de 4KB
- Memoria flash cuya finalidad principal es albergar el programa de 128KB.
- Ranura para la inserción de una tarjeta micro SD de capacidad máxima 2GB.
- Real Time Clock DS3231SN de alta resolución que permite una cuantificación temporal suficientemente precisa como para programar alarmas e interrupciones. Adicionalmente el RTC tiene integrado un sensor de temperatura.
- Múltiples entradas y salidas: 2 UART's, comunicación I²C, comunicación SPI, comunicación USB, 7 puertos analógicas con conversor A/D de 10bits, 8 puertos digitales, un puerto digital programable como PWM

Adicionalmente dispone de distintos módulos que pueden ser integrados tales como:

- Módulo ZigBee de baja y alta potencia acordes con IEEE 802.15.4 de 2.4 GHz, 868Mhz ó 900MHz
- Módulo GSM/GPRS cuatribanda
- Módulo GPS
- Módulo sensorial compuesto por una placa de sensores apta para albergar dispositivos proporcionados por el fabricante.
- Módulo de almacenamiento mediante tarjeta de memoria microSD.
- Dispositivo GateWay que genera un puente desde las motas hasta un ordenador vía puerto USB utilizando comunicación serie.

» Meshlium:

Libelium presenta este dispositivo como un router multitecnología que integra modulo Wifi mesh, ZigBee, GPRS, GPS y bluetooth soportado por una máquina basada en un

Linux Debian embebido. Su utilidad principal es manejar redes complejas con múltiples tecnologías con un único dispositivo permitiendo conectar una red moteada con los usuarios mediante distintas plataformas, así el usuario de la red puede recibir información sobre la red a través del teléfono móvil, a través de un servidor FTP o a través de una pagina web de desarrollo propio en internet.

Este Frutero fabricado por libelium tiene integradas las siguientes características:

- Procesador 500MHz de 32bits.
- Memoria RAM de tecnología DDR de 256MB.
- Memoria de disco entre 8Gb y 32Gb
- Radio Wifi Atheros AR5213A acorde compatible son IEEE 802.1 1b/g con antena omnidireccional.
- Radio WifiMesh de banda Dual de 2.4/5 GHz Atheros AR5213A compatible con IEEE 802.1 1a/b/g con antena omnidireccional.
- Radio ZigBee Pro seleccionable (802.15.4, ZigBee, 868, 900, XSC...)
- Radio Bluetooth BT100 1.1 acorde con IEEE 802.15.1 con antena 5dBi dipolar.
- Modulo GSM/GPRS cuatribanda modelo hilo.
- Modulo GPS modelo A2095 de Vincotech con salida NMEA

Atendiendo a las características de este dispositivo se pueden crear redes entre dispositivos bluetooth o ZigBee cuyos datos medidos podrían ser enviados a un ordenador local perteneciente a la red, a ordenadores remoto perteneciente una red de otro Meshlium o incluso depositar los datos en internet utilizando el gestor de base de datos que incluye el dispositivo:

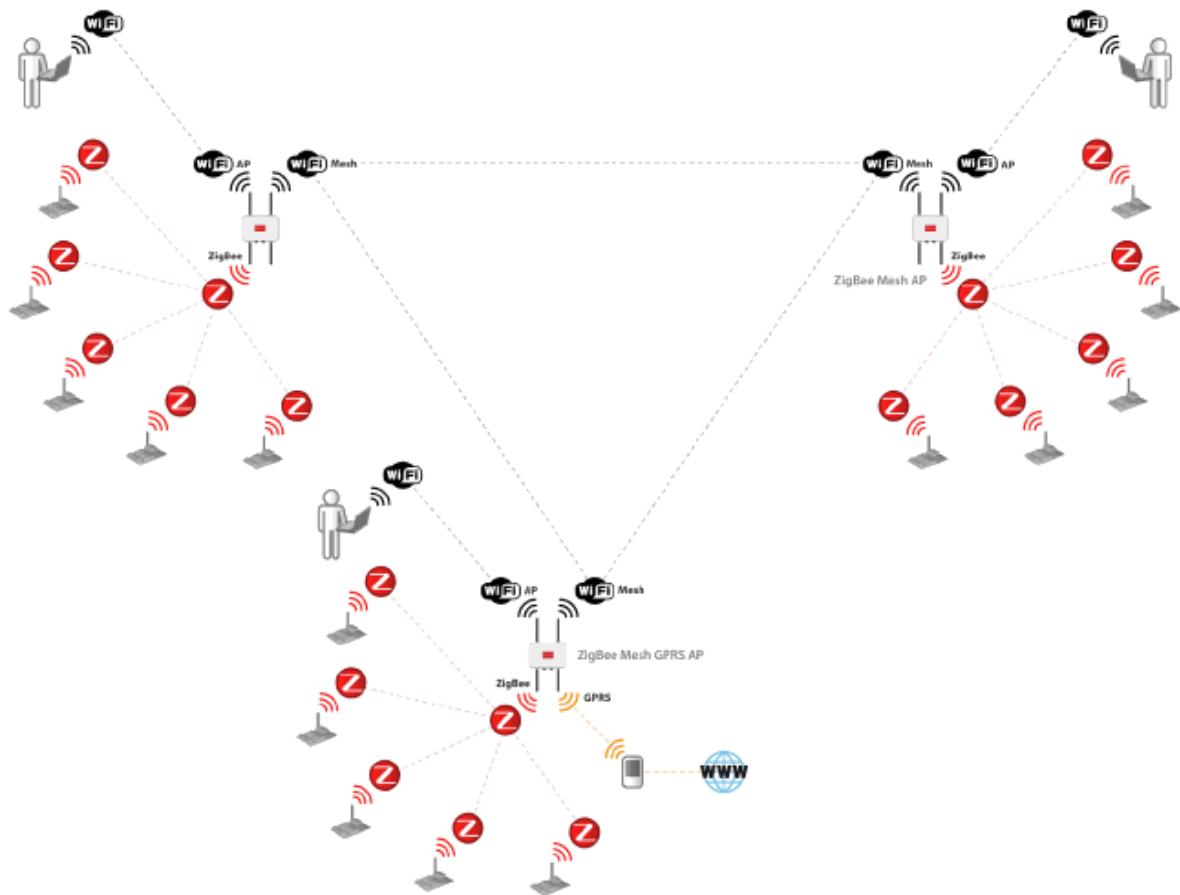


Ilustración 7: Ejemplo de red inalámbrica multitecnología

2.3. Justificación de la elección y características del dispositivo elegido:

2.3.1. Justificación de la elección:

En la fase preliminar de este proyecto, no se tenía la inequívoca certeza de la compatibilidad de la placa del robot Qfix con la tecnología arduino de libelium. Por ello se decidió intentar componer el escenario con la tecnología que existía en el laboratorio. No obstante, no fue el hecho de disponer de este material en el laboratorio la única razón por la cual fue elegido; sino que cumple, e incluso en algunos puntos supera, todas las especificaciones y características que hacen de esta plataforma un candidato excepcional tal y como se reflejará más adelante y a lo largo de todo el proyecto.

Tras comprobar la compatibilidad e interconexión de las plataformas de Qfix y de libelium se consigue completar una red sensorial con funcionalidad completa, pudiendo sustituir la mota concentradora de información por un router Meshlium dotando al sistema con funcionalidades extras como se apuntará en el apartado de posibles mejoras y líneas de investigación futuras.

2.3.2. Contenido del kit desarrollador WaspMote:

Al inicio del proyecto se disponía en el laboratorio de un kit de desarrollador de WaspMote de libelium cuyo contenido es:

- 5 WaspMote
- 1 WaspMote GateWay
- 5 Baterías de litio
- 5 Baterías auxiliares
- 6 Cables USB
- 6 Radios Xbee + antena
- 5 Tarjetas microSD
- 5 Placas de sensores.

2.3.3. Funcionalidades:

2.3.3.1. Descripción general:

En principio se incluye una foto de la parte frontal del WaspMote en la que se resaltan determinados elementos de interés:

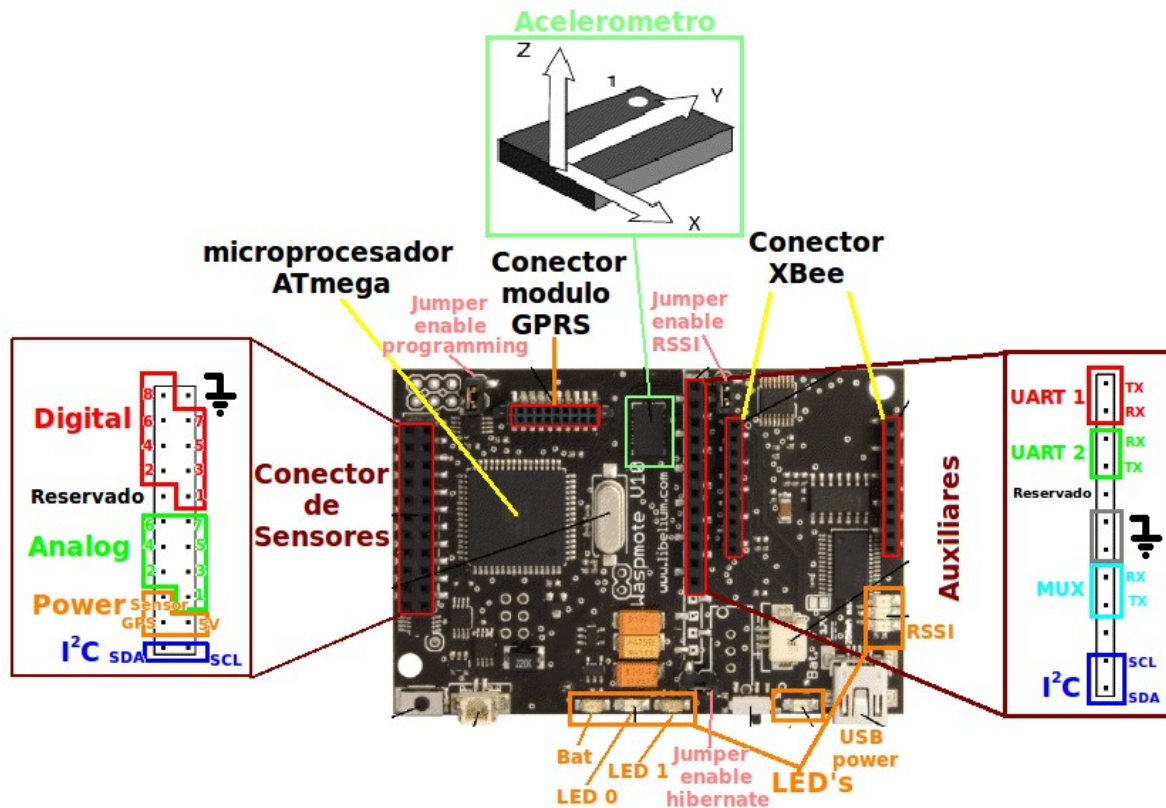


Ilustración 8: Detalle delantero placa WaspMote

Cabe resaltar de esta imagen los elementos más representativos utilizados en este proyecto. En el conector de elementos auxiliares se encuentran los terminales SCL y SDA que conforman el puerto I²C, descrito en el capítulo de protocolos, encargado de comunicar el WaspMote con el Minibot. Se obtendrá información sobre la aceleración de la placa WaspMote transportada por el Minibot utilizando el chip tri-axial integrado de WaspMote. Es fundamental el papel que juegan los jumpers, sirviendo como medida de protección ante programación del dispositivo o como protección para evitar que entre en modo de hibernación.

La foto de la parte trasera del WaspMote con sus elementos resaltados es la siguiente:

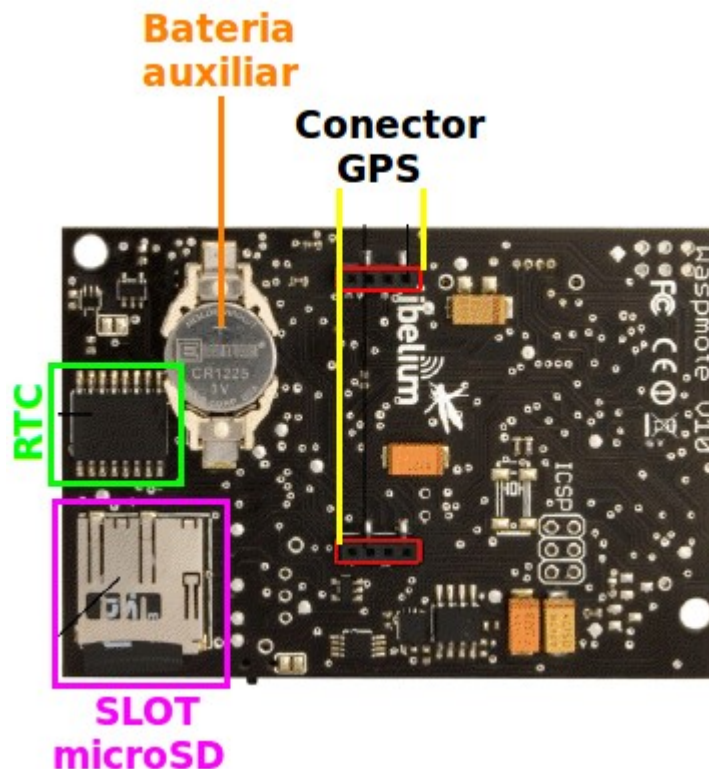


Ilustración 9: Detalle trasero placa WaspMote

Siendo de especial interés el microchip de Real Time Clock que permitirá generar eventos que despierten del letargo a la placa para enviar la información que se le requiera de forma precisa. Debido al efecto de la temperatura en el cristal de cuarzo que este chip necesita para medir el tiempo el fabricante integró un sensor de temperatura para poder contrarrestar la deriva producida en la medición del tiempo, al poder acceder a la lectura de este sensor es posible obtener la temperatura entre un rango de -45°C y 85°C del WaspMote cada vez que se requiera.

2.3.3.2. Programar el dispositivo:

La carga de un programa nuevo en el dispositivo se realizará a través de la versión de libelium de la IDE genérica de Arduino. Para conseguir actualizar el programa introducido en la flash de la mota es condición necesaria que el puente

de programación habilitada esté colocado y que no estén conectados a la mota la pila de alimentación auxiliar, la placa de eventos ni el modulo Xbee.

Los pasos a realizar son los siguientes:

- 1.- Colocar el puente de programación.
- 2.- Desconectar la batería auxiliar y modulo Xbee
- 3.- Conectar la placa al USB
- 4.- Poner en interruptor de la mota en posición ON
- 4.- Ejecutar la IDE
- 5.- Identificar la mota con el puerto serie correspondiente
- 4.- Cargar el programa en la IDE
- 5.- Compilar el programa
- 6.- Cargar el programa compilado en la mota
- 7.- Apagar la mota o pulsar el botón de reiniciar para permitir la ejecución del código introducido.

2.3.3.3. Modos de funcionamiento:

Puesto que uno de los objetivos de las motas es permanecer sin intervención humana en un lugar enviando paquetes de datos a otro nodo con la información captada en la posición en la que se encuentra. Pero para permitir esto durante una cantidad de tiempo razonable comienza a ser importante que el dispositivo consuma la menor cantidad de energía posible, y que al ser posible, tenga medios para recargar sus fuentes de energía. En el caso de WaspMote, esta necesidad imperativa de las redes de sensores se ha traducido en distintos modos de funcionamiento que permiten que la mota siga funcionando con un comportamiento aletargado incurriendo así en una reducción del consumo de energía tan demandada en este tipo de dispositivos.

En este apartado se comenta como es posible pasar del modo de funcionamiento habitual de “alto consumo” con una corriente de 9mA a modos aletargados incurriendo en gastos energéticos de 0,7μA. En la siguiente tabla se resumen los

posibles estados, el consumo energético y los elementos que trabajan:

	Consumo	Micro	Ciclo	Interrupciones
ON	9mA	ON	-	Síncronas y Asíncronas
Sleep	62µA	ON	32ms - 8s	Síncrona (Watchdog) y Asíncronas
Deep Sleep	62µA	ON	8s - min/horas/días	Síncrona (RTC) y Asíncronas
Hibernate	0,7µA	OFF	8s - min/horas/días	Síncrona (RTC)

La descripción de los estados es la siguiente:

- **ON:** Modo de funcionamiento normal. Atiende a todo tipo de interrupciones.
- **Sleep:** En este modo se detienen algunas de las funcionalidades del módulo y se pasa a un funcionamiento asíncrono, normalmente dirigido por eventos generados por los módulos sensoriales, de comunicación o temporizadores.
- **DeepSleep:** Es un modo similar al sleep donde el microprocesador no atiende a los pulsos generados por su reloj hasta que una interrupción debida al Real Time Clock o a un sensor es generada despertando así al microprocesador.
- **Hibernación:** En este modo se detienen todas las funcionalidades del módulo pasando a un uso asíncrono, normalmente dirigido por los eventos generados por el Real Time Clock

A continuación se puede observar una imagen que describe el funcionamiento y cambio de entorno entre modos de funcionamiento junto con las ordenes que nos permiten pasar de modo ON a cualquiera de los otros modos y las interrupciones que sacan al WaspMote del estado aletargado:

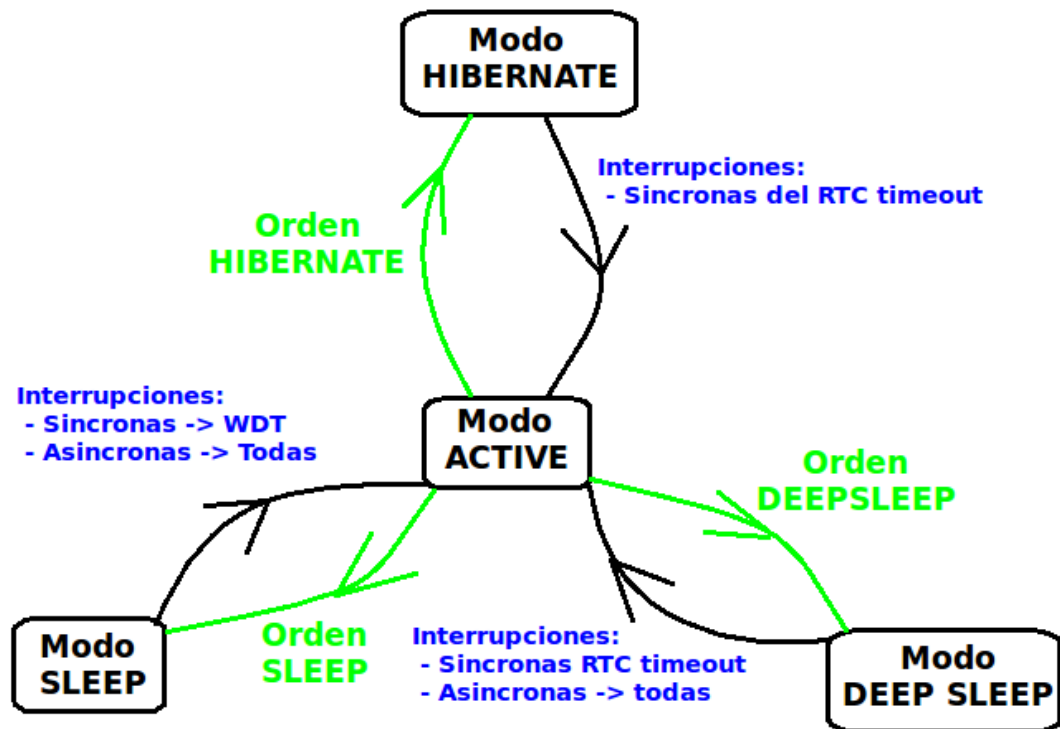


Ilustración 10: Modos de funcionamiento WaspMote