

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO DE FIN DE GRADO

PLATAFORMA INALÁMBRICA PARA LA
WEB FÍSICA

GRADO EN INGENIERÍA DE
TECNOLOGÍAS DE TELECOMUNICACIÓN

JOSÉ ANTONIO YÉBENES GÁLVEZ
MÁLAGA, 2018

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD DE MÁLAGA

Titulación: Grado en Ingeniería de Tecnologías de Telecomunicación

Reunido el tribunal examinador en el día de la fecha, constituido por:

D./D^a. _____

D./D^a. _____

D./D^a. _____

para juzgar el Trabajo de Fin de Grado titulado:

Plataforma inalámbrica para la web física

del alumno D./D^a. *José Antonio Yébenes Gálvez*

dirigido por D./D^a. *Ignacio Herrero y José Manuel Cano*

ACORDÓ POR _____ OTORGAR LA

CALIFICACIÓN DE _____

y, para que conste, se extiende firmada por los componentes del tribunal, la presente diligencia.

Málaga, a _____ de _____ de _____

El Presidente:

El Vocal:

El Secretario:

Fdo.: _____ Fdo.: _____ Fdo.: _____

Universidad de Málaga
Escuela Técnica Superior de Ingeniería de
Telecomunicación

PLATAFORMA INALÁMBRICA PARA LA WEB FÍSICA

REALIZADO POR
José Antonio Yébenes Gálvez

DIRIGIDO POR
Ignacio Herrero y José Manuel Cano

Dpto. de: Tecnología Electrónica (DET)

Palabras clave: IOT, web, red, sensores, inalámbrica, física, objetos, internet

Titulación: Grado en Ingeniería de Tecnologías de Telecomunicación

Resumen: Aquí se explica un breve resumen del proyecto, en qué consiste, cuál es su propósito, etc.

Puede usarse más de un párrafo si es necesario.

Málaga, 2 de febrero de 2018

A mi familia,
por facilitarme cumplir mis objetivos.
A Lucía,
por corregir mi rumbo cuando lo pierdo.

Acrónimos

API	<i>Application Programming Interface</i>
APL	<i>Application</i>
BLE	<i>Bluetooth Low Energy</i>
CCS	<i>Code Composer Studio</i>
CM0	<i>Cortex M0</i>
CM3	<i>Cortex M3</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
DFE	<i>Directed Frame Exchange</i>
ETSIT	Escuela Técnica Superior de Ingeniería de Telecomunicación
H2M	<i>Human-to-Machine</i>
HP	<i>Hewlett-Packard</i>
ICall	<i>Indirect Call Framework</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IOT	Internet de las cosas o <i>Internet Of Things</i>
ISM	<i>Industrial, Scientific and Medical</i>
LBT	<i>Listen Before Talk</i>
M2M	<i>Machine-to-Machine</i>
MAC	<i>Medium Access Control</i>
MPU	microprocesador

MCU	microcontrolador
MT	<i>Managment and Test</i>
MVC	Modelo Vista Controllador
NPI	interfaz de protocolo de red
NWK	<i>Network</i>
PAM	red de área personal o <i>personal area network</i>
PHY	<i>Physical</i>
QR	<i>Quick Response</i>
RF	Radiofrecuencia
RTOS	sistema operativo en tiempo real
TFG	Trabajo Fin de grado
UART	<i>Universal Asynchronous Receiver/Trasnsmitter</i>
uBle	<i>Micro BLE Stack</i>
UMA	Universidad de Málaga
URL	<i>Uniform Resource Locator</i>
Wavenis-OSA	Wavenis Open Standard Alliance
WSN	<i>Wireless Sensor Networks</i>

Índice

Acrónimos	IX
I Introducción	1
1 Objetivos	3
2 Metodología	5
Memoria	6
Concentrador	6
Servidor	6
Nodo	7
II Estado del arte y la técnica	9
3 Web física	11
4 Redes inalámbricas e internet de las cosas	15
Visión general de las soluciones existentes	16
5 Plataformas para comunicaciones inalámbricas	19
ESP8266	19
Zig-Bee	19
6 Plataformas basadas en Linux	21
BeagleBone Black	21
Raspberry Pi	21

III	Desarrollo del proyecto	23
7	Descripción general	25
	Esquema General	25
	Nodo	25
	Concentrador	26
	Raspberry Pi	27
	Servidor	27
8	TI 15.4-Stack	29
	Introducción	29
	Elección de arquitectura	29
	Banda de frecuencias y tasa de transmisión	31
	Alcance	31
	Indirect Call Framework	31
	uBle	32
	Modos de operación	33
9	Nodo	35
	Introducción	35
	Arquitectura Hardware	35
	TI-RTOS	37
	Arquitectura de la aplicación	37
	Función de inicio	38
	Arquitectura general de la aplicación	38
	Web física	39
10	Concentrador	41
	Introducción	41
	Diagrama de bloques y modelo de la interfaz	41
	Descripción del SDK	42
	Aplicación	44
11	Servidor	45
	Backend	45
	Frontend	45
IV	Pruebas y funcionamiento	47
12	Prueba en entorno real	49
	Localización	49

13 Prueba de consumo	51
Introducción	51
Medida del consumo	51
Resultados	52
14 Explicación de funcionamiento	55
Iniciar el sistema	55
 V Conclusiones y líneas futuras	 57
Conclusiones y líneas futuras	59
Conclusiones	59
Líneas futuras	59
 VI Apéndices	 61
15 Apéndice	63
15.1 Primera sección	63
 Bibliografía	 66

Índice de figuras

3.1	Logo de la web física	11
3.2	Ejemplos de comunicación cercana	12
7.1	Esquema general del proyecto	26
8.1	Configuración como dispositivo único y como coprocesador	30
8.2	Aplicación ICall - Abstracción del protocolo	32
8.3	Ejemplo de mensajes ICall	33
9.1	Diagrama de bloques de Simplelink™CC13x0	36
9.2	Diagrama de bloques de la aplicación	37
9.3	Diagrama de estados de la función de inicio	38
9.4	Flujo de la aplicación	39
10.1	Arquitectura de software a alto nivel de las aplicaciones TI 15.4-Stack 2.1.0 Linux®	42
10.2	Estructura del directorio TI 15.4-Stack 2.1.0 Linux®	43
13.1	Programas de LabView	52
13.2	Consumo en mA del nodo a la máxima frecuencia de muestreo	52

Índice de Tablas

8.1	Bandas permitidas en TI 15.4-Stack y las frecuencias de sus canales .	31
9.1	Formato de la trama <i>Eddystone-URL</i>	40

Parte I

Introducción

Capítulo 1

Objetivos

La web física es un término que describe la forma de comunicar cualquier objeto físico con la web. A partir este enfoque, es posible navegar y controlar objetos en el mundo a través de dispositivos móviles. Esto ofrece a los usuarios la manera de realizar sus tareas diarias utilizando los objetos de su entorno. Para utilizar este enfoque, lo primero que hay que elegir es la manera con la que el objeto se comunicará con el usuario, tal como códigos QR o etiquetas RFID. De los diferentes enfoques que tiene la web física, el que se va a tratar en este proyecto es el basado en proximidad inalámbrica.

El modelo que plantea la web física, los objetos pueden necesitan un canal por donde enviar y recibir datos. Hay diferentes técnicas enmarcadas dentro del Internet de las cosas o *Internet Of Things* (IOT). El problema se agrava cuando los objetos están distribuidos y no tienen cerca un punto de acceso a internet o tienen que funcionar con baterías. En estos casos, buscar una solución de comunicación inalámbrica de gran alcance y bajo consumo es prioritario.

Con estos conceptos en mente, el presente trabajo de fin de grado va a abordar el concepto de web física, implementando una red inalámbrica que permita comunicarse con los objetos con el fin de ser una red versátil. Finalmente, esta tecnología se aplicará en un caso concreto de uso.

Capítulo 2

Metodología

Para la correcta realización de este Trabajo Fin de grado (TFG) se han elegido diversas herramientas y metodologías.

Cuando se aborda un proyecto de investigación, la primera fase consiste en investigar y aprender sobre la materia. Para esto se han utilizado tres portales principalmente: Google Académico, IEEE Xplore[®] y la web de Texas Instruments. De estos portales se ha extraído la información necesaria para el desarrollo de este proyecto.

Los proyectos software normalmente necesitan un control de versiones, para poder trazar qué cambios se han realizado en el proyecto y poder volver a versiones anteriores si fuera necesario. Para este proyecto se ha utilizado Github que es un servidor para alojar proyectos utilizando el sistema de control de versiones GIT. Al inicio del proyecto se crearon cuatro repositorios diferentes donde se ha guardado la mayor parte de la información del TFG. Los repositorios son enumerados a continuación:

TFG_memoria (https://github.com/joseayebenes/TFG_Memoria) Almacena la información sobre esta memoria escrita en L^AT_EX.

TFG_HostCollector (https://github.com/joseayebenes/TFG_HostCollector) Este repositorio contiene el proyecto del concentrador o nodo central de la red.

TFG_Sensor (https://github.com/joseayebenes/TFG_Sensor) Contiene el código que ejecutan los nodos de la red.

TFG_Gateway (https://github.com/joseayebenes/TFG_Gateway) En este se guarda el proyecto del servidor.

Cada uno de los repositorios alberga partes diferentes del proyecto y cada una de estas partes han tenido herramientas y metodologías diferentes.

Memoria

La memoria es la parte final de un proyecto, es donde se resumen la investigación y se presentan los resultados. Para su realización se ha utilizado L^AT_EX, que es un sistema de composición de textos y el entorno TeXstudio para facilitar el trabajo y compilación del documento. Además se han utilizado herramientas como StarUML o Adobe Illustrator para la creación de los diagramas e imágenes.

Concentrador

Esta parte está destinada a ser ejecutada en un entorno Linux, por lo que ha sido programado en C++, usando el editor de textos SublimeText para la edición de los archivos de código.

Uno de los inconvenientes al realizar esta parte era trabajar en un sistema operativo distinto al instalado en el ordenador, por lo que en las primeras fases del proyecto se utilizó una máquina virtual con el sistema operativo Ubuntu en VirtualBox, aunque la falta de recursos hacía lento el desarrollo. Como solución, se comenzó a trabajar utilizando la Raspberry Pi que finalmente se incluiría en el proyecto creando unos *scripts* que automatizaban la tarea de descargar el código de GitHub, compilar y ejecutar. De esta forma se puede programar en cualquier sistema operativo y al subir los cambios a GitHub, la Raspberry Pi se los descargaba, compilaba y ejecutaba.

Servidor

El servidor está basado en Node.js[®], que es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. El código en JavaScript del servidor se realizó utilizando SublimeText.

La parte del *FrontEnd*, es decir, la parte con la que el usuario interactúa, está basada en el *FrameWork* AngularJS, que es una *FrameWork* de JavaScript de código abierto basado en el patrón de diseño Modelo Vista Controlador (MVC).

En la fase final, el servidor se alojó en un servidor dedicado de Amazon WebServices basado en Ubuntu Server, donde se crearon *scripts* similares a los del concentrador para automatizar los despliegues de los cambios realizados.

Nodo

El nodo está programado en C usando el IDE CodeComposerStudio y librerías del fabricante.

Parte II

Estado del arte y la técnica

Capítulo 3

Web física

El primer concepto que hay que conocer para afrontar el proyecto es el de Web Física. La Web Física es un término que describe el proceso de presentar objetos cotidianos en internet[1]. Este enfoque ofrece a los usuarios móviles la posibilidad de gestionar sus tareas diarias en el uso de objetos cotidianos. Los objetos comienzan a ser inteligentes y remotamente controlables. Este modelo permite a los usuarios móviles navegar y controlar los objetos físicos que rodean al dispositivo móvil. Además, esto ayuda al desempeño de tareas diarias dependiendo de los objetos cercanos [2].



Figura 3.1: Logo de la web física

Podemos mencionar en este contexto a los conocidos códigos *Quick Response* (QR), los códigos QR son un código de barras en dos dimensiones, a menudo utilizados para mapear *Uniform Resource Locator* (URL) con objetos físicos [3].

Las etiquetas inalámbricas son uno de los enfoques más utilizados para el marcado de objetos físicos. Las etiquetas inalámbricas pueden soportar protocolos como *Bluetooth Low Energy* (BLE) y *WiFi*. Los protocolos mencionados son soportados por la mayoría de los teléfonos móviles modernos.

En la Web Física, personas, lugares y objetos tienen páginas web que proveen información y mecanismos de interacción. Sin embargo, es la amplitud y la profundidad de la pila que rodea a la web, que hacen de esta una atractiva visión para la evolución del IOT.[1]

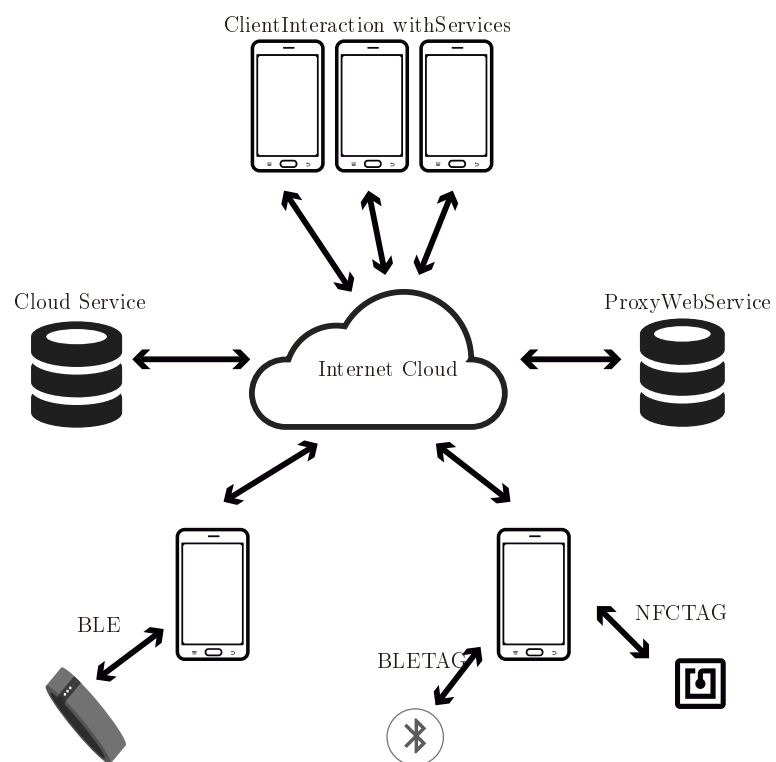


Figura 3.2: Ejemplos de comunicación cercana

Las páginas web son una fantástica tecnología para interacción *Human-to-Machine* (H2M), pero muchos de los casos de uso del IOT son interacciones *Machine-to-Machine* (M2M). Los formatos de datos usados por **Schema.org** y otros, permiten a los agentes de usuario y servicios en la nube analizar los datos para eventos, organizaciones, personas, lugares, productos y así sucesivamente, acutando sobre ellos de forma interactiva y proactiva. [1]

Uno de los primeros proyectos en fomentar esta idea fue *Hewlett-Packard* (HP) con *Cooltown*, que usaba balizas infrarrojas para transmitir URLs. Más recientemente, BLE proporciona una similar baliza de bajo consumo que puede emitir URL en paquetes periódicos (www.uribeacon.org). [1]

Capítulo 4

Redes inalámbricas e internet de las cosas

Contenido

Visión general de las soluciones existentes	16
---	----

El futuro de internet tiene como meta integrar diferentes tecnologías de comunicación, cableadas e inalámbricas, con el objetivo de contribuir sustancialmente a mejorar el concepto de IOT [4]. Aunque hay muchas maneras de describir el IOT, podemos definirlo como una red con objetos interconectados con direcciones únicas, basadas en un protocolo estándar de comunicación.[5]

Los sensores de bajo costo han facilitado la proliferación de *Wireless Sensor Networks* (WSN) en muchos escenarios como monitorización medioambiental, agricultura, salud, y construcciones inteligentes. WSN están caracterizadas por una alta heterogeneidad porque están basadas en diferentes soluciones, propietarias y no propietarias. Este gran rango de soluciones está retrasando actualmente desarrollos a gran escala de estas tecnologías a fin de que se obtenga una gran red virtual de sensores que permita integrar todos las existentes redes de sensores.[6]

Las redes de sensores basadas en sistemas cerrados o propietarios son islas conectivamente hablando, con limitadas comunicaciones con el mundo exterior. Por lo general, es necesario usar *gateways* con conocimiento específico de la aplicación para exportar los datos de la WSN a otros dispositivos conectados a Internet. Además, no hay comunicación directa entre diferentes protocolos a menos que se implementen complejas conversiones específicas para la aplicación en los *gateways* o *proxies*.

Visión general de las soluciones existentes

En este apartado presentaremos una rápida visión general de las principales tecnologías usadas para WSN [7]. Analizaremos las soluciones que no están basadas en los protocolos de internet.

ZIGBEE

ZigBee es una tecnología de red inalámbrica desarrollada por la ZigBee Alliance para baja tasa de transmisión de datos y aplicaciones de corto alcance [8]. La pila de protocolos ZigBee está compuesta por 4 principales capas: la capa *Physical* (PHY), la capa *Medium Access Control* (MAC), la capa *Network* (NWK) y la capa *Application* (APL). PHY y MAC de ZigBee están definidas por el estándar *Institute of Electrical and Electronics Engineers* (IEEE) 802.15.4, mientras el resto de la pila está definida por la especificación ZigBee.

Esta versión inicial del IEEE 802.15.4, en la que ZigBee está basado, funciona en la bandas de 868 MHz (Europa), 915 MHz (Norteamérica) y 2.4 GHz (global).

Una nueva especificación de ZigBee es RF4CE [9], que tiene una simplificada pila de red para topologías en estrellas solamente, ofreciendo una solución simple para el control remoto de electrónica de consumo.

Z-WAVE

Z-Wave es un protocolo inalámbrico desarrollado por ZenSys y promovida por la Z-Wave Alliance para automatizaciones residenciales y pequeños entornos comerciales. El principal objetivo de permitir transmisiones seguras de cortos mensajes desde una unidad de control hasta uno más nodos en la red [10]. Z-Wave esta organizado de acuerdo a un arquitectura cumpuesta por 5 capas principales: PHY, MAC, transferencia, enrutado y capas de aplicación.

Z-Wave opera principalmente en la banda de 900 MHz (868 MHz en Europa y 908 MHz en Estados Unidos) y 2.4 GHz. Z-Wave permite tasas de transmisión a 9.6 kb/s, 40 kb/s y 200 kb/s.

INSTEON

INSTEON [?] es una solución desarrollada por SmartLabs y promovida por la INSTEON Alliance. Una de las distintivas características de INSTEON es el hecho de como define una topología de red compuesta de Radiofrecuencia (RF) y *power*

line links. Los dispositivos pueden ser RF, *power-line*, o pueden soportar ambos tipos de comunicación.

INSTEON opera a 904 MHz como frecuencia central, con una tasa de datos bruta de 38.4 kb/s.

Los dispositivos INSTEON son parejas, lo que significa que cualquiera de ellos pueden tener el rol de emisor, receptor o repetidor. La comunicación entre ambos dispositivos que no estén en el mismo rango se logra mediante un enfoque «multisalto» que usa los repetidores en un esquema de sincronización temporal.

WAVENIS

Wavenis es un protocolo inalámbrico desarrollado por Coronis System para el control y monitorización de aplicaciones en entornos exigentes, incluida la domótica y la automatización de edificios. Wavenis actualmente está promovida y gestionada por la Wavenis Open Standard Alliance (Wavenis-OSA). Está definido por la funcionalidad de las capas física, de enlace y de red [11]. Los servicios de Wavenis pueden ser accedidos desde capas superiores mediante una *Application Programming Interface* (API).

Wavenis opera principalmente en las bandas de 433 MHz, 868 MHz y 915 MHz, que son bandas reservadas para *Industrial, Scientific and Medical* (ISM) en Asia, Europa y Estados Unidos. Algunos productos también operan en la banda de 2.4 GHz. Las tasas de transmisión mínimas y máximas dadas por Wavenis son 4.8 kb/s y 100 kb/s, respectivamente, con 19.2 kb/s como valor típico.

TI 15.4-STACK

El TI 15.4-Stack es una plataforma completa libre de derechos de autor para desarrollar aplicaciones que requieren una solución inalámbrica con topología en estrella, un extremado bajo consumo, largo alcance, fiable, robusto y seguro.

Este capítulo 8 se explicará en detalle las características de este protocolo y el uso en este proyecto.

Capítulo 5

Plataformas para comunicaciones inalámbricas

Contenido

ESP8266	19
Zig-Bee	19

ESP8266

Zig-Bee

Capítulo 6

Plataformas basadas en Linux

Contenido

BeagleBone Black	21
Raspberry Pi	21

BeagleBone Black

Raspberry Pi

Parte III

Desarrollo del proyecto

Capítulo 7

Descripción general

Contenido

Esquema General	25
Nodo	25
Concentrador	26
Raspberry Pi	27
Servidor	27

Esquema General

El presente trabajo de fin de grado se puede representar como se observa en la figura 7.1, donde están representados los diferentes elementos que componen la arquitectura.

Del diagrama se extrae que hay cuatro dispositivos distintos: nodo, concentrador, Raspberry Pi y servidor. Cada uno de estos elementos tiene su propia línea de ejecución diferente de los demás. También se observa como están interconectados los diferentes dispositivos usando protocolos diferentes según sea la comunicación. En los siguientes apartados se repasarán la función de cada uno de los dispositivos y como se comunican con los demás.

Nodo

El nodo es uno de los extremos de la red, y es el encargado de comunicarle al usuario la URL tal y como define la web física. Este está basado en el microcontro-

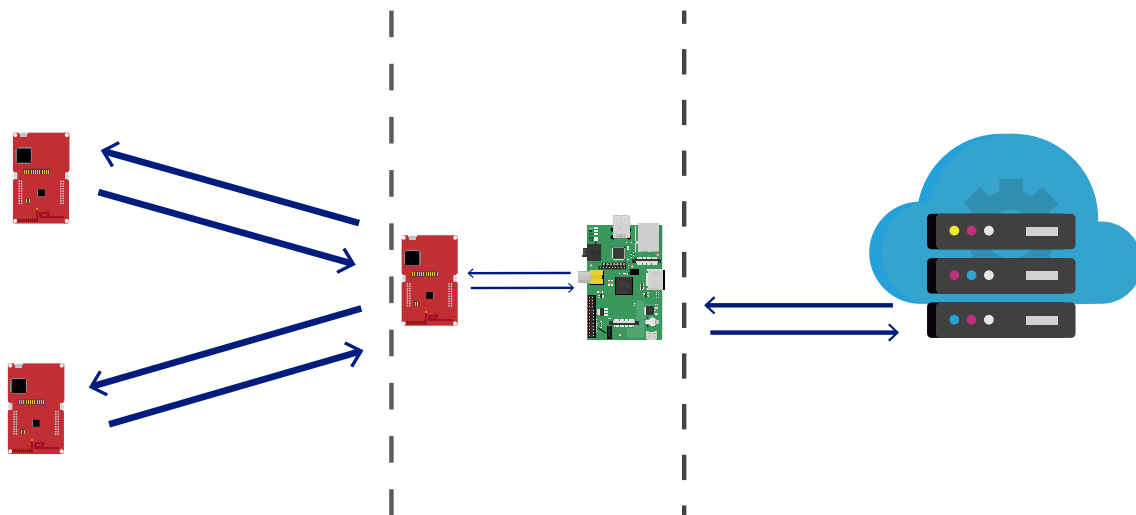


Figura 7.1: Esquema general del proyecto

lador CC1350 SimpleLink™ de Texas Instruments que nos permite comunicación en dos bandas de frecuencia diferentes, en nuestro caso 868MHz y 2.4Ghz.

Con el uso de estas dos bandas de frecuencia, el nodo se comunicará con el usuario usando la banda de 2.4GHz y el protocolo bluetooth. Y con el resto de la red usando la banda de 868MHz y el protocolo TI 15.4 Stack.

Usando la red inalámbrica TI 15.4 Stack, el nodo podrá enviar y recibir información desde otros puntos de la red, como por ejemplo cambiar la url que está enviando o enviar datos de sensores.

Concentrador

El concentrador es el nodo central de la red TI 15.4 Stack, este se encarga de comunicarse con los nodos a 868MHz. Este dispositivo ejecuta un código precompilado, que implementa una capa 802.15.4e/g MAC/PHY y proporciona una interfaz basada en el protocolo *Management and Test* (MT) que conecta el dispositivo con el host linux, en nuestro caso una Raspberry Pi.

Raspberry Pi

Servidor

Capítulo 8

TI 15.4-Stack

Contenido

Introducción	29
Elección de arquitectura	29
Banda de frecuencias y tasa de transmisión	31
Alcance	31
Indirect Call Framework	31
uBle	32
Modos de operación	33

Introducción

El TI 15.4-Stack es una plataforma completa libre de derechos de autor para desarrollar aplicaciones que requieren una solución inalámbrica con topología en estrella, un extremado bajo consumo, largo alcance, fiable, robusto y seguro.

Este capítulo explicará en detalle los diferentes modos de configuración de la red soportadas por el TI 15.4-Stack.

Elección de arquitectura

TI 15.4-Stack se puede utilizar con diferentes arquitecturas. En la figura 8.1 se observan 2 diferentes arquitecturas permitidas por el TI 15.4-Stack.

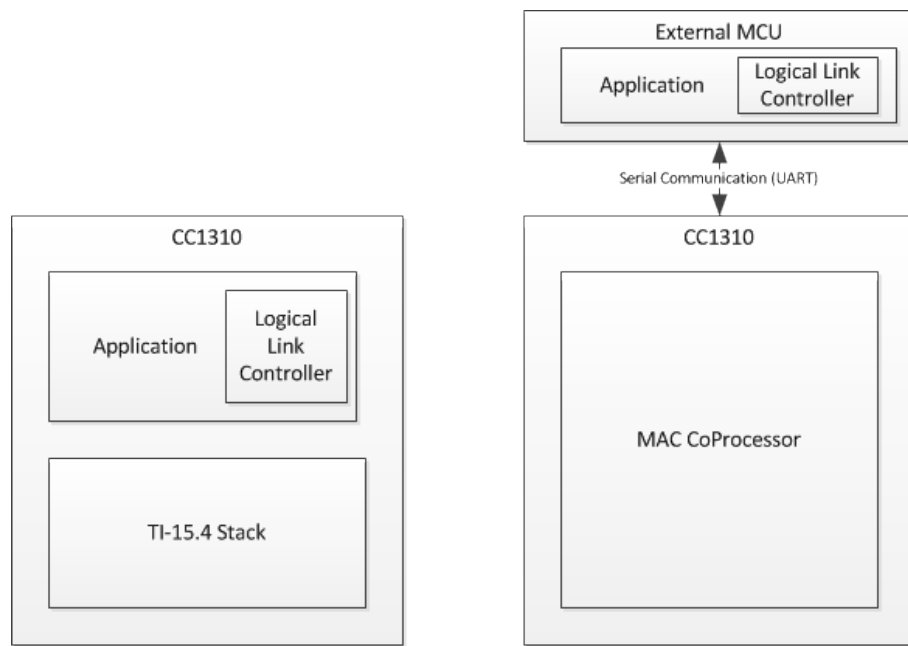


Figura 8.1: Configuración como dispositivo único y como coprocesador

- Una configuración como dispositivo único la podemos observar en la figura 8.1 (izquierda). La aplicación y la pila del protocolo son implementados en el CC1350 como una solución *single-chip*. Esta configuración es la más simple y común cuando se usa el CC1350 para nodos de la red. Esta configuración es la técnica más económica y de menor consumo.
- Como coprocesador se observa en la figura 8.1 (derecha). La pila de protocolos se ejecuta en el CC1350 mientras la aplicación es ejecuta en un microprocesador (MPU) o microcontrolador (MCU). La aplicación se comunica con el CC1350 usando el interfaz de protocolo de red (NPI) sobre una comunicación serie *Universal Asynchronous Receiver/Trasnsmitter* (UART). Esta configuración es usada en aplicaciones que añadan comunicación inalámbrica de rango alcance o en un ordenador sin los requerimientos para implementar los complejos protocolos asociados a una red inalámbrica.

Para el desarrollo de este TFG se utilizarán ambas arquitecturas, usando la arquitectura de dispositivo único para los nodos y la basada en coprocesador para el nodo central o concentrador.

Banda de frecuencias y tasa de transmisión

La elección de una banda y una tasa de transmisión elegirse configurando el apropiado atributo (PHY ID). Las opciones son explicadas en la tabla 8.1.

PHY ID	Tasa de datos	Frecuencia canal 0	Nº canales	Espacio canales
0	250 kbps	2405 MHz	16	5 MHz
1	50 kbps	902.2 MHz	129	200 kHz
30	50 kbps	863.125 MHz	34	200 kHz
128	50 kbps	403.3 MHz	7	200 kHz
129	5 kbps	902.2 MHz	129	200 kHz
130	5 kbps	403.3 MHz	7	200 kHz
131	5 kbps	863.125 MHz	34	200 kHz
132	200 kbps	902.4 MHz	64	400 kHz
133	200 kbps	863.225 MHz	17	400 kHz

Tabla 8.1: Bandas permitidas en TI 15.4-Stack y las frecuencias de sus canales

Se han utilizado las frecuencias de 2.4GHz y 868MHz por ser bandas de uso libre en España. Usando la banda de 868Mhz para la comunicación de largo alcance y la de 2.4GHz para enviar los mensajes de la web física.

Alcance

Indirect Call Framework

ICall es un módulo que provee un mecanismo para que la aplicación se comunique con los servicios del TI 15.4-Stack, así como con ciertos servicios primitivos proporcionados por el sistema operativo en tiempo real (RTOS). *Indirect Call Framework* (ICall) permite que la aplicación y la pila del protocolo operen eficientemente, comunicandose y compartiendo recursos en un entorno unificado RTOS.

El componente central de la arquitectura ICall es el *dispatcher*, que facilita la comunicación entre la aplicación y las tareas del TI 15.4-Stack.

La figura 8.3 muestra un ejemplo de como un comando se envía desde la aplicación hasta el TI 15.4-Stack, con su correspondiente respuesta.

ICall_init() inicializa la instancia del módulo ICall y la llamada *ICall_createRemotetasks()* crea una tarea, con una función de entrada en una dirección conocida. Después de inicializar el ICall, la tarea de la aplicación se registra con el módulo ICall usando *ICall_registerApp()*. Durante la ejecución de la tarea de la aplicación, esta envía un

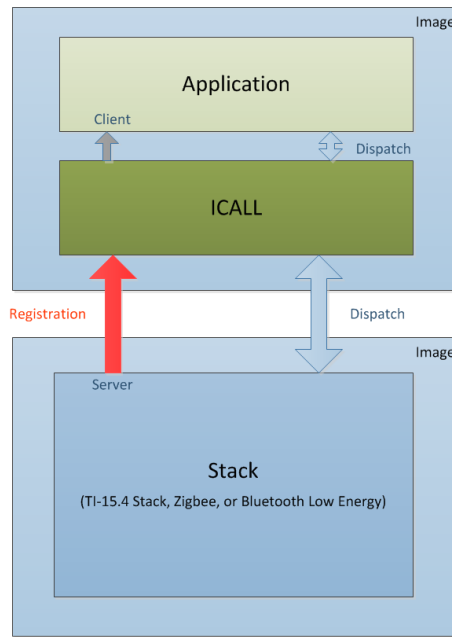


Figura 8.2: Aplicación ICall - Abstracción del protocolo

comando del protocolo como por ejemplo *ApiMac_mlmeSetreqArray()*. El comando no se ejecuta en el hilo de la aplicación, en lugar, el comando es encapsulado en un mensaje ICall y es dirigido a la tarea del TI 15.4-Stack a través del módulo ICall. Mientras la aplicación espera al correspondiente mensaje. Cuando el TI 15.4-Stack finaliza la ejecución del comando, la respuesta es enviada a través del módulo ICall a la aplicación.

uBle

Micro BLE Stack (uBle) es una variante del paquete BLE-Stack para los dispositivos con conectividad Sub1-GHz y 2.4GHz, como el CC1350. Este paquete permite a las aplicaciones ser encontradas, escanear o actuar como monitor de conexión. El paquete uBle utiliza el *MultiMode RF Driver*. El *MultiMode RF Driver* permite a las aplicaciones usar ambos modos donde otro protocolo de comunicación es integrado junto al uBle.

Restricciones y requisitos

El Micro BLE Stack tiene las siguientes restricciones y requisitos:

- Las opciones de diseño dependen de una parcial integración de ICall para

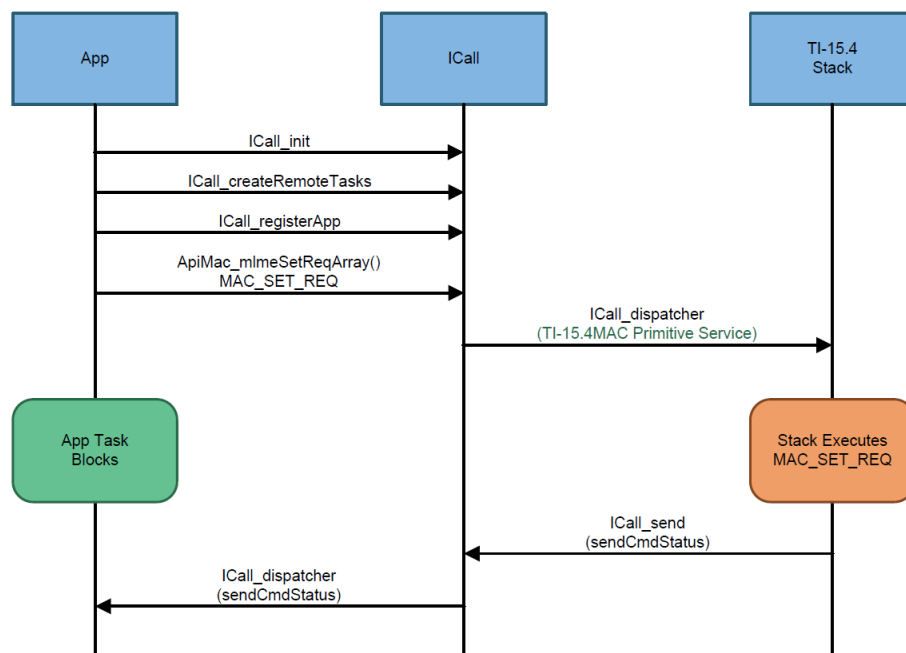


Figura 8.3: Ejemplo de mensajes ICall

guardar recursos del sistema. En el caso de uso del módulo ICall, se tiene que utilizar el sistema de gestión de pila del módulo ICall.

- No puede haber interacción humana-computador porque no existe separación entre el controlador y el host.
- Las opciones de privacidad no son soportadas.
- Para minimizar el consumo de memoria y eliminar redundantes cambios de contexto el uBle no utiliza diferentes tareas en TI-RTOS.
- Solo configuraciones utilizando el *MultiMode RF Driver* pueden ser usada con otros protocolos RF.

Modos de operación

Modo Beacon

Las especificaciones IEEE 802.15.4 definen un modo de operación *beacon-enabled* donde el dispositivo coordinador de la **PAN!** (**PAN!**) transmite *beacons* para indicar su presencia y permite que otros dispositivos encuentren la red de área personal

o *personal area network* (PAM) y se sincronicen. Los *beacons* proporcionan información sobre las especificaciones de la super-trama, que ayuda a los dispositivos con la intención de unirse a la red a sincronizarse y conocer los parámetros de la red antes de comenzar el proceso de unión. La super-trama está dividida en periodos activo e inactivos. Durante el periodo activo, los dispositivos se comunican usando el procedimiento *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) excepto en la banda de 863MHz donde se usa el procedimiento **LSB!** (**LSB!**) para el acceso al canal. Los periodos inactivos permiten a los dispositivos en la red conservar energía.

Modo NonBeacon

Las especificaciones IEEE 802.15.4 definen un modo de operación *non-beacon* donde el coordinador de la red no envía *beacons* periódicos. El modo *non-beacon* es un modo de operación asíncrono donde los dispositivos se comunican usando el mecanismo CSMA/CA.

Modo FrequencyHopping

Las aplicaciones que son desarrolladas usando el TI 15.4-Stack pueden ser configuradas para operar en redes con saltos de frecuencia donde los dispositivos de la red cambian de frecuencia. Este modo de funcionamiento está basado en el modo *Directed Frame Exchange* (DFE) de las especificaciones de Wi-SUN FAN.

Capítulo 9

Nodo

Contenido

Introducción	35
Arquitectura Hardware	35
TI-RTOS	37
Arquitectura de la aplicación	37
Función de inicio	38
Arquitectura general de la aplicación	38
Web física	39

Introducción

La aplicación implementa el dispositivo de la red, que le permite conectarse a la red creada por el concentrador. El sensor periódicamente envía reportes de datos en intervalos configurados por el concentrador y este responde con mensajes de rastreo.

Arquitectura Hardware

ARM Cortex M0 (Núcleo radio)

El núcleo *Cortex M0* (CM0) en el CC1350 es responsable de la interfaz audio, y traduce complejas instrucciones del núcleo *Cortex M3* (CM3) en bits que son enviados a través del enlace radio. Para el protocolo TI 15.4-Stack, el CM0 implementa la capa PHY de la pila de protocolos.

El *firmware* del núcleo de radio no está destinado a ser usado o modificado por la aplicación del desarrollador.

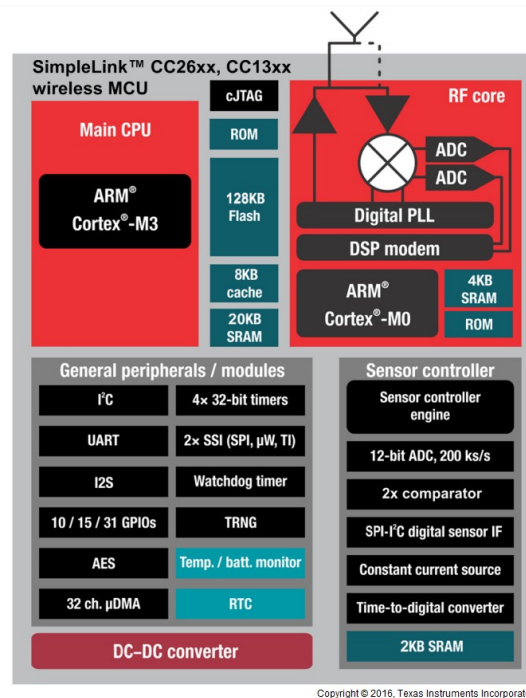


Figura 9.1: Diagrama de bloques de Simplelink™CC13x0

ARM Cortex M3 (Núcleo del sistema)

El núcleo CM3 está diseñado para ejecutar la pila del protocolo inalámbrico desde la capa de enlace hasta la capa de aplicación de usuario. La capa de enlace actúa como interfaz del núcleo de radio como un módulo software llamado *RF driver*.

Flash, RAM y periféricos

El CC1350 contiene en el sistema 128KB de memoria flash programable, 20KB de SRAM, y un amplio rango de periféricos. La memoria flash se divide en partes que se pueden borrar de 4KB. El CC1350 también contiene 8kB de caché SRAM que puede ser utilizada para extender la capacidad de la RAM o puede funcionar como una caché normal para incrementar el rendimiento de la aplicación. Otros periféricos incluidos son UART, I2C, I2S, AES, TRNG, temperatura y monitor de la batería.

TI-RTOS

TI-RTOS es un entorno operativo para proyectos TI 15.4-Stack en dispositivos CC1350. El kernel TI-RTOS es una versión adaptada del kernel SYS/BIOS y funciona como un sistema operativo con controladores en tiempo real, con prioridades, multitarea y herramientas para la sincronización y planificación.

Arquitectura de la aplicación

En la figura 9.2 se muestra el diagrama de bloques de la aplicación del nodo.

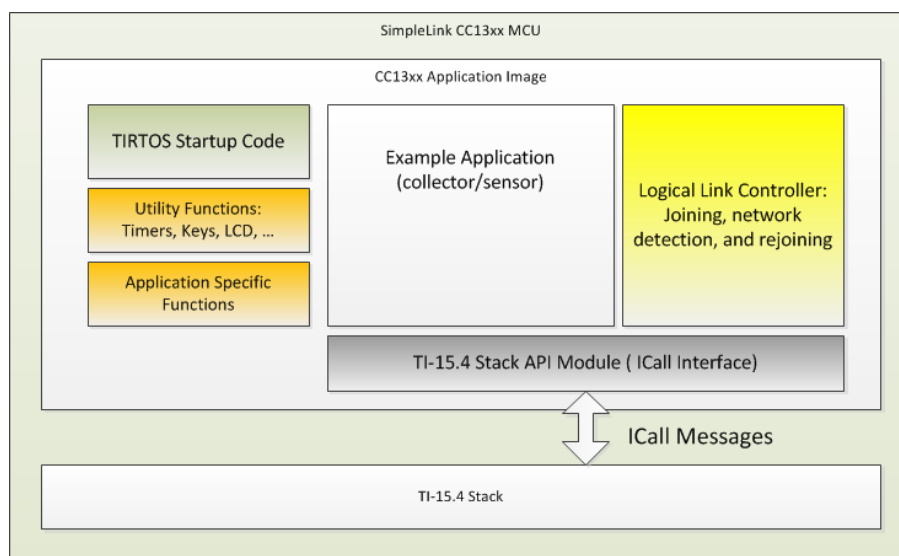


Figura 9.2: Diagrama de bloques de la aplicación

Descripción a alto nivel de varios bloques de la figura 9.2:

Application: Este bloque

Logical Link Controller: Implementa varias funciones específicas del IEEE 802.15.4 o Wi-SUN (para una configuración *frequency-hopping*) para formación, conexión y reconexión de la red.

TI-RTOS Start-up Code: Inicializa la aplicación.

Utility Functions Provee varias utilidades para usar LCD, temporizadores, botones y más.

Application-Specific Functions: Implementa funciones como guardado de datos, y provee una interfaz para gestionar pulsaciones botones o mostrar información esencial en un LCD.

TI 15.4-Stack API Module (API MAC Module): Este módulo proporciona una interfaz para gestión y los servicios de datos del 802.15.4 stack mediante el módulo ICall.

Función de inicio

La función *main()* dentro del archivo *main.c* es el punto de inicio de la ejecución de la aplicación. En este punto los componentes relaciones con la placa son inicializados. Las tareas se configuran en esta función, inicializando los parámetros necesarios como su prioridad y su tamaño en la pila. En el paso final, las interrupciones se habilitan y el planificador *SYS/BIOS* se inicia llamando a *BIOS_start()*.

Arquitectura general de la aplicación

Esta sección describe como la tarea de la aplicación esta estructura en más detalle.

Función de inicio de la aplicación

Después de que la tarea sea construida y el planificador *SYS/BIOS* se inicie, la función que se le pasa durante la construcción de la tarea es ejecutada cuando la tarea está lista.

Las funciones de gestión de la energía son inicializadas aquí y el módulo ICall se inicia con la función *ICall_init()*. La dirección IEEE address (programada por TI) es obtenida desde la memoria flash. La tarea de la aplicación (Aplicación Sensor) es inicializada y ejecutada.

Sensor_init() establece varios parámetros de configuración, así como:

- Inicializa las estructuras para los datos

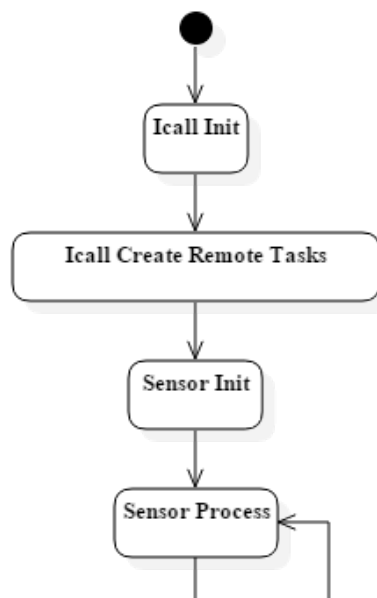


Figura 9.3: Diagrama de estados de la función de inicio

- Inicializa el TI 15.4-Stack
- Configura la seguridad y el *Logical Link Controller*
- Registra las funciones de retorno MAC

Tarea principal

Después de la función de inicialización, la tarea entra en un bucle infinito ejecutando siempre las mismas tareas, se puede ver en la figura

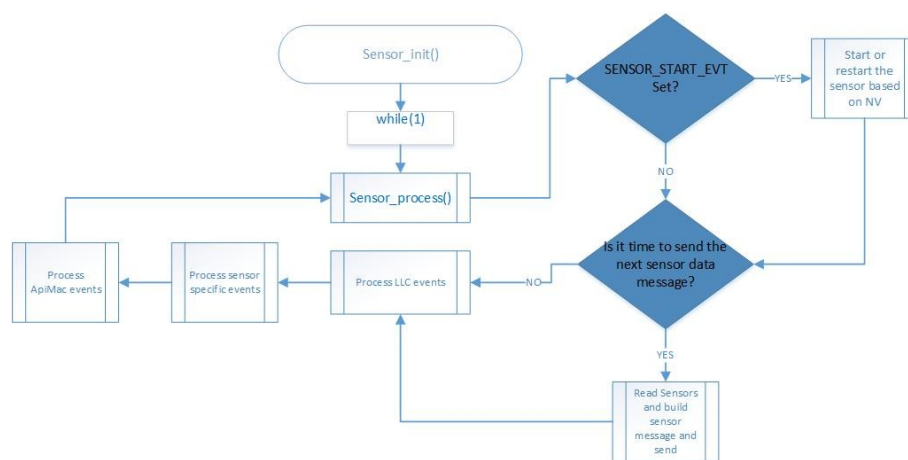


Figura 9.4: Flujo de la aplicación

Web física

La librería uBle permite a la aplicación enviar un paquete en modo *broadcast* a todos los usuarios que estén en el radio de acción del dispositivo. Este paquete de datos es el que utilizamos para notificar a los usuarios con la información de la web física.

Los paquetes bluetooth usan la trama *Eddystone-URL*, estas forman parte del núcleo de la web física. Una vez el usuario la decodifica la trama podrá acceder a la URL si tiene conexión a internet.

Byte offset	Campo	Descripción
0	Tipo de trama	valor = 0x10
1	Potencia TX	Potencia de TX calibrada a 0 m
2	Prefijo de la URL	Prefijo de la web
3+	URL codificada	Longitud de 1-17 bytes

Tabla 9.1: Formato de la trama *Eddystone-URL*

Especificaciones de la trama

Potencia TX La potencia de transmisión es la potencia recibida a 0 metros, en dBm, en el rango de valores de -100 dBm a +20dBm con una resolución de 1 dBm.

Prefijo de la URL El prefijo de la url define la expansión utilizada por la url, por ejemplo “http://www.” o “https://” son codificadas por los bytes 0x00 o 0x03 respectivamente.

Sufijo de la URL El esquema de URL HTTP está definida por RFC 1738, por ejemplo “https://goo.gl/S6zT6P”, y es usada para designar recursos accesibles usando HTTP.

Capítulo 10

Concentrador

Contenido

Introducción	41
Diagrama de bloques y modelo de la interfaz	41
Descripción del SDK	42
Aplicación	44

Introducción

Diagrama de bloques y modelo de la interfaz

Esta sección describe la arquitectura de alto nivel basada en coprocesador, los componentes software, y la arquitectura general del sistema (véase figura 10.1). El coprocesador es una entidad que implementa el estándar MAC IEEE 802.15.4e/g en un chip dedicado y provee una interfaz serie por la que un procesador externo controla y procesa las operaciones del coprocesador.

El coprocesador se centra en una arquitectura escalable con una división perfecta donde el procesador ejecuta las capas sobre el IEEE 802.15.4e/g MAC/PHY.

En esta aplicación, el programa se ejecuta en una plataforma basada en Linux. Aunque los componentes de alto nivel , pueden ser conceptualmente aplicados a otras plataformas no basadas en Linux. Los componentes desarrollados serán descritos más adelante.

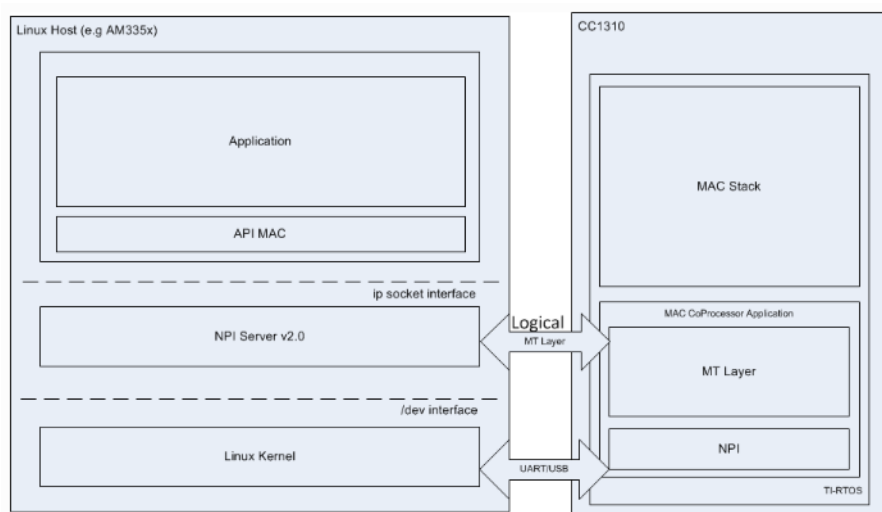


Figura 10.1: Arquitectura de software a alto nivel de las aplicaciones TI 15.4-Stack 2.1.0 Linux®

La interfaz entre el procesador y el coprocesador están definidas como capas lógicas que están separadas en esta arquitectura: una capa física (por ejemplo, USB o UART), una capa lógica de enlace, y la capa de presentación.

Componentes software:

Aplicación del coprocesador: Es el programa ejecutandose en el dispositivo CC1350. Esta aplicación implementa una capa 802.15.4e/g MAC/PHY y proporciona una comunicación serie.

Kernel Linux : El kernel Linux provee los controladores para la interfaz serie que está disponible en un puerto físico (por ejemplo, USB).

Aplicación TI 15.4-Stack: Este módulo implementa la aplicación usando el protocolo 802.15.4e/g y la estructura del modelo MT.

Descripción del SDK

La figura 10.2 muestra la estructura del directorio de instalación del TI 15.4-Stack 2.1.0. A continuación se explica una descripción de alto nivel de cada carpeta:

components: Contiene las siguientes librerías:

common: Rutinas para características del sistema operativo, como lectura y escritura de ficheros.

h



Figura 10.2: Estructura del directorio TI 15.4-Stack 2.1.0 Linux®

nv: Simula una memoria no volátil, como la usada en sistemas empotrados.

api: Interfaz de mensajes API MAC y MT

docs: Documentos como la guía de desarrollo y la guía de comandos MAC para el coprocesador.

example: Aplicación de ejemplo

cc13xx-sbl: Herramientas para la actualización para los dispositivos CC13x0.

collector: Aplicación de ejemplo que demuestra como iniciar una red, permitir la conexión de dispositivos y recoger datos desde dispositivos remotos.

gateway: Una aplicación basada en Node.js™ que crea un servidor local y muestra la información de la red y los datos de los nodos.

npi_server2: Interfaz socket para comunicarse con el coprocesador.

google: Contiene un makefile para descargar e instalar el compilador de *Google protocol buffer*.

firmware Precompilados ficheros .hex para el coprocesador.

prebuilt Compilación para ejecutar la aplicación de ejemplo en una BeagleBone Black.

scripts Contiene fragmentos de ficheros makefile usados para compilar la aplicación de ejemplo.

Aplicación

Capítulo 11

Servidor

Contenido	
Backend	45
Frontend	45

Backend

Frontend

Parte IV

Pruebas y funcionamiento

Capítulo 12

Prueba en entorno real

Contenido

Localización	49
------------------------	----

Localización

Capítulo 13

Prueba de consumo

Contenido

Introducción	51
Medida del consumo	51
Resultados	52

Introducción

Para una exitosa implementación del IOT uno de los requisitos más importantes es un consumo eficiente del sistema.

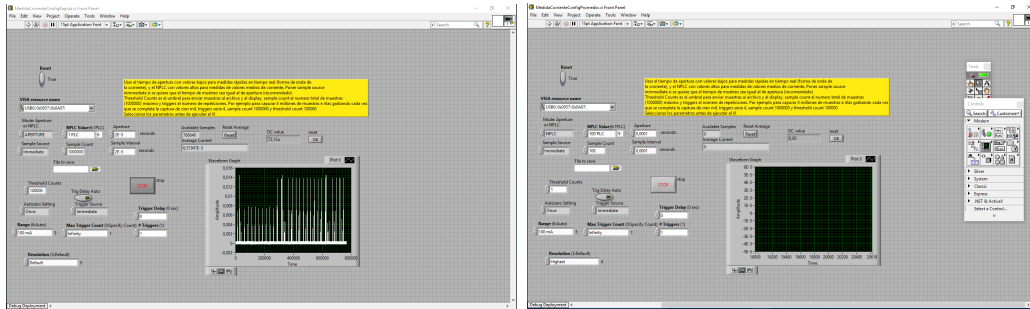
La comunicación inalámbrica entre los nodos comienza a ser un problema cuando la fuente de alimentación es limitada. Un sistema que pueda funcionar con una sola pila AAA durante años es lo ideal en el IOT [12]. Por ello, durante el desarrollo de este proyecto se ha utilizado como protocolo inalámbrico el IEEE 802.15.4, que destaca su bajo consumo.

Medida del consumo

Para realizar la medida de consumo se ha utilizado el multímetro digital Keysight 34411A, que proporciona 6 dígitos y medio de resolución y una velocidad de muestreo de 50000 muestras/s. [13]

Para el control del multímetro se ha utilizado dos programas de LabView, uno de ellos permite obtener datos a la máxima velocidad de muestreo aunque eso limite

la resolución a 4 dígitos (figura 13.1a) y el otro programa permite tomar datos promediados en intervalos de tiempo (figura 13.1b).



(a) Captura rápida

(b) Captura promedio

Figura 13.1: Programas de LabView

Resultados

Aunque el objetivo de esta prueba es conocer el consumo medio de corriente del nodo. Antes de nada se necesita conocer que fondo de escala utilizar en el multímetro para la medida en promedio, para ello se hace uso del programa de captura a máxima frecuencia de muestreo, donde se obtiene la corriente instantánea máxima que consume el nodo, unos 15 mA (figura 13.2).

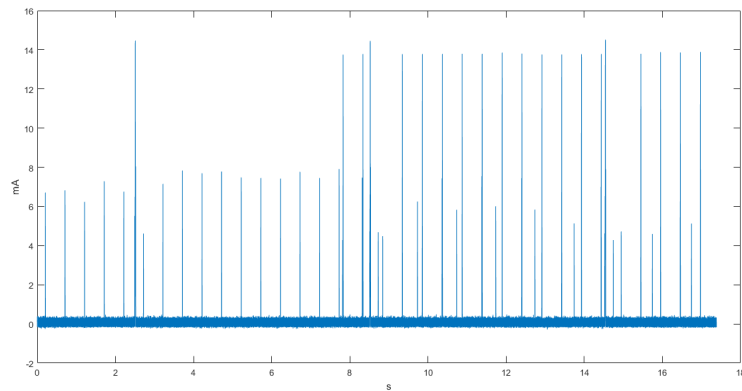


Figura 13.2: Consumo en mA del nodo a la máxima frecuencia de muestreo

Finalmente, utilizando el programa de medida promediada de LabView con un fondo de escala de 100mA, realizamos una medida de la corriente promedio durante 50 ciclos de subidas al servidor. Con esta configuración se obtiene que el consumo

medio de un nodo es de $169.6 \mu A$, lo que equivale a más de 5 meses de funcionamiento utilizando una pila de botón CR 2477.

Capítulo 14

Explicación de funcionamiento

Contenido

Iniciar el sistema	55
------------------------------	----

Iniciar el sistema

Parte V

Conclusiones y lineas futuras

Conclusiones y líneas futuras

Conclusiones

Después de todo el desarrollo del proyecto, es pertinente hacer una valoración final del mismo, respecto a los resultados obtenidos, las expectativas o el resultado de la experiencia acumulada.

En esta sección se exponen todos esos conceptos y enuncian unas conclusiones finales.

Además, considerando también el estado de la técnica, se pueden deducir líneas futuras de trabajo, proponer otros puntos de vista o cualquier otra sugerencia como postámbulo del presente trabajo, para ser considerada por el lector o el tribunal evaluador.

Líneas futuras

José Antonio Yébenes Gálvez
2 de febrero de 2018

Parte VI

Apéndices

Capítulo 15

Apéndice

Contenido

15.1 Primera sección	63
--------------------------------	----

15.1. Primera sección

Bibliografía

- [1] Roy Want, Bill N. Schilit, and Scott Jenson. Enabling the internet of things. *Computer*, 2015.
- [2] Dmitry Namiot and Manfred Sneps-Sneppe. The physical web in smart cities. *Advances in Wireless and Optical Communications*, 2015.
- [3] Yue Liu, Ju Yang, and Mingjun Liu. Recognition of QR code with mobile phones. In *Control and Decision Conference*.
- [4] Gerd Kortuem, Fahim Kawsar, and Vasughi Sundramoorthy. Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 2009.
- [5] Luca Mainetti, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. *Telecommunications and Computer Networks*, 2011.
- [6] Michele Zorzi, Alexander Gluhak, and Sebastian Lange. From today’s intranet of things to a future internet of things: a wireless and mobility-related view. *IEEE Wireless Communicationss*, 2010.
- [7] Carles Gomez and Josep Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 2010.
- [8] Zigbee Alliance. Zigbee home automation public application profile. Technical report, Zigbee Alliance, 2007.
- [9] Zigbee Alliance. The zigbee rf4ce standard. Technical report, Zigbee Alliance, 2009.
- [10] Z-Wave. Z-wave protocol overview. Technical report, Z-Wave, 2007.
- [11] Ana Belén García Hernando, José Fernán Martínez Ortega, Juan Manuel López Navarro, Aggeliki Prayati, and Luis Redondo López. *Problem Solving for Wireless Sensor Networks*. Springer, 2008.

- [12] Mahmoud Shuker Mahmoud and Auday AH Mohamad. A study of efficient power consumption wireless communication techniques/modules for internet of things (iot) applications. *Advances in Internet of Things*, 6(02):19, 2016.
- [13] Keysight. *Keysight Technologies 34410A and 34411A Multimeters. Data sheet.*