



Universidad  
de La Laguna

Escuela Superior de  
Ingeniería y Tecnología  
Sección de Ingeniería Informática

# Trabajo de Fin de Grado

---

## Estudio de 6LoWPAN para su aplicación a Internet de las Cosas

*6LoWPAN study for application to the Internet of Things*

Daniel García García.

---

La Laguna, 9 de junio de 2015

D. **Alberto Hamilton Castro**, con N.I.F. 43773884P profesor Titular de Universidad adscrito al Departamento de Ingeniería de Sistemas y Automática de la Universidad de La Laguna, como tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Estudio de 6loWPAN para su aplicación a Internet de las Cosas”*

ha sido realizada bajo su dirección por D. **Daniel García García**, con N.I.F. 54064993C.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 9 de junio de 2015

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

## **Resumen**

*El protocolo 6LoWPAN surgió como una idea del IETF en 2004 para aprovechar la potencia de direccionamiento de IPv6 junto con la flexibilidad de los dispositivos de bajo procesamiento y consumo, el objetivo es realizar un estudio intenso sobre la utilización y la estructura interna de este protocolo en el internet de las cosas y su viabilidad en los componentes electrónicos adquiridos para tal fin.*

*El primer reto fue, partiendo de base cero y dado lo novedoso del tema, investigar sobre la tecnología IPv6 para luego abordar la 6LoWPAN. Al obtener los dispositivos electrónicos proporcionados por el profesor se empezó a trabajar la plataforma de desarrollo, las librerías y los programas necesarios para implementar una pequeña red 6LoWPAN. Al principio y tras varios esfuerzos por entender la arquitectura de la placa, se consiguió conectar, a través de los interfaces de red inalámbricos, aprovechando la pila de protocolos que ofrece el fabricante. Aún no había capa de red con la que se pudiera hacer enrutamiento, para ello, se opta por la implantación de otro ejemplo proporcionado, que lanza ipv6 sobre redes como las que se probaron. El principal problema fue el tener que utilizar una nueva arquitectura de procesador que no está previsto en la planificación del proyecto, por lo que no se han podido alcanzar los objetivos originales en el plazo previsto.*

**Palabras clave:** 6LoWPAN, IETF, IPv6, bajo procesamiento, bajo consumo, internet de las cosas, arquitectura.

## **Abstract**

*The 6LoWPAN protocol arrived with an idea from IETF in 2004, the objective is use the high power of networking of IPv6 next to the flexibility of the devices with low processing and power consume, the objective is create an intense study on the use and the internal structure of this protocolo of Internet of Things and its viability in the use in some electronic devices.*

*The first challenge was, starting from scratch and an given innovative topic, reserch the IPv6 tecnology and then approach to 6LoWPAN. When the teacher gave me the electronic devices, I began to work in the developer platform, libraries and the basics programs that I need to run a small 6LoWPAN network. Initially, during the efforts to know the architecture of the electronic devices, I achieved connect two devices and both wireless networks interfaces using the factory owner stack protocol. At this moment, I didn't use any network layer for routing, for that, an other example is used to implement this, it employ the IPv6 on past checked networks. The main problem was use a new architecture of proccessor, it wasn't planned in the proyect, for that, the original objectives has not been able to reach in the period referred.*

**Keywords:** 6LoWPAN, IEFT, IPv6, low proccesing, low power, internet of things, arquitecure

# Índice general

<b>Capítulo 1 Introducción.....</b>	<b>1</b>
<b>Capítulo 2 Conceptos Previos.....</b>	<b>2</b>
2.1 Internet de las Cosas.....	3
2.2 Modelo OSI.....	4
2.2.1 Capa física.....	4
2.2.2 Capa de Enlace.....	4
2.2.3 Capa de Red.....	4
2.2.4 Capa de Transporte.....	4
2.2.5 Capa de Sesión.....	4
2.2.6 Capa de Presentación.....	4
2.2.7 Capa de Aplicación.....	4
2.3 Entidades de normalización.....	6
2.3.1 IETF.....	6
2.3.2 IEEE.....	6
2.4 6LoWPAN.....	7
2.4.1 Introducción.....	7
2.4.2 Data Link Layer.....	7
2.4.2.a IEEE 802.15.....	8
2.4.2.b Dirección EUI-64 MAC.....	9
2.4.2.c CSMA/CA.....	10
2.4.2.d Cabecera 802.15.4.....	10
2.4.3 Ipv6 Network Layer.....	11
2.4.3.a Capacidad del protocolo.....	11

2.4.3.b Clases de dirección.....	12
2.4.3.c Compresión de la cabecera IP.....	13
2.4.3.d Seguridad.....	16
2.4.3.e Control de red.....	17
2.4.3.f Autoconfiguración.....	19
<b>Capítulo 3 Plataforma física.....</b>	<b>22</b>
3.1 Familias de microcontrolador.....	22
3.1.1.a ARM.....	22
3.1.1.b AVR.....	22
3.2 Distribuidor de Productos.....	22
3.3 Microcontroladores.....	23
3.3.1 Dispositivos ARM.....	23
3.3.2 Dispositivos AVR.....	23
3.4 Módulos de radio Integrados.....	23
3.5 Placas auxiliares.....	24
3.5.1 Placas de expansión de puertos.....	24
3.5.2 Herramienta de desarrollo y depuración.....	25
<b>Capítulo 4 Desarrollo.....</b>	<b>26</b>
4.1 Primera fase.....	26
4.2 Segunda fase.....	27
4.3 Tercera fase.....	27
4.3.1 Pila 6LoWPAN.....	27
4.3.2 Tipos de dispositivos.....	29
4.3.3 Topologías de red.....	29
4.4 Apoyo.....	31
4.4.1 Herramientas en Windows.....	31
4.4.2 Herramientas en Linux.....	31
<b>Capítulo 5 Conclusiones y líneas futuras.....</b>	<b>33</b>
<b>Capítulo 6 Summary and Conclusions.....</b>	<b>34</b>
<b>Capítulo 7 Presupuesto.....</b>	<b>35</b>
<b>Capítulo 8 Bibliografía.....</b>	<b>36</b>

# Capítulo 1

## Introducción

Dentro del estudio que compone el Trabajo de fin de Grado se ha decidido enfocar hacia un tipo de tecnología en desarrollo llamada **6LoWPAN**. A groso modo componen las interacciones entre sistemas embebidos de diferentes arquitecturas de bajo coste y bajo consumo en entornos de redes inalámbricas de área personal (WPAN).

Destacar la necesidad de comprender el amplio rango de conceptos que se encuentran entre la capa física de los dispositivos y la capa de red con la que se interconectan. Ante la necesidad de comprender qué se nos presenta como problema inicial, el estudio previo es clave para la culminación eficiente del proyecto, de ahí que se opte por modular en puntos importantes.



# Capítulo 2

## Conceptos Previos

Es necesario conocer una serie de conceptos previos.

MCU	Microcontrolador, dispositivo limitado en cuanto a repertorio de instrucciones que realiza funciones también limitadas.
JTAG	JTAG es un estándar para probar las interconexiones entre los chips de circuitos integrados en una placa de circuito impreso .
WPAN	Define las redes de área personal con un rango no más de 10 metros, vienen definidas por la baja tasa de transmisión de datos que permite el aprovechamiento de la vida útil de los dispositivos.
WLAN	Componen las redes de área local permiten la interconexión de dispositivos a mayor escala y mayor distancia, unos 150 metros aproximadamente.
Frecuencia de onda	Número de repeticiones por unidad de tiempo, a mayor frecuencia menor longitud de onda.
Ancho de banda	Es la cantidad de información que es capaz de enviar o recibir un dispositivo, se mide en bits/s.

MAC	Compone la dirección con la que se comunica el dispositivo en capas de red físicas, de ahí que se conozca como dirección física.
MTU	El la máxima unidad de transferencia.
Payload	Son los bits que acompañan a una cabecera, es decir, son los datos que una vez leídos serán útiles.

Redes.
Sistemas Embebidos.
Radio.

## 2.1 Internet de las Cosas

Es un concepto que viene asociado a la interconexión digital de dispositivos cotidianos. El nombre le viene gracias a Kevin Ashton en 1999, se le conoce por desarrollar en el departamento Auto-ID Labs del MIT el estándar global de comunicación de RFID.

Existen muchas definiciones para este concepto pero no existe una forma estandarizada de explicar qué es. A modo de resumen el Internet de las Cosas es la comunicación de una cantidad elevada de dispositivos de bajo consumo que mediante su interacción en un entorno limitado permite la automatización y control del mismo.

En este proyecto se estudia las capacidades de interconexión y direccionamiento pero queda claro que las aplicaciones que se pueden montar sobre estas capas son bastante elevadas siempre teniendo en cuenta el entorno donde se realicen. SmartCities, Domótica, automatización industrial, ... lo que cambia es el entorno, la filosofía es la misma. Inteligencia Ubicua o ambiental, Web semántica, ... son conceptos que se desarrollarán a partir de esta base.

## **2.2 Modelo OSI**

Es un modelo que fue creado por la Organización Internacional de Normalización (ISO, International Organization of Standardization), permite estandarizar las diferentes fases que deben experimentar los datos para viajar por la red. Está compuesto por siete capas que se explicarán a continuación.

### **2.2.1 Capa física**

Viene siendo la capa más baja y más cercana al hardware. Abarca un gran número de dispositivos y su conexión en diferentes topologías de red, pueden llegar a comunicar en diferentes medios de transmisión tales como aire o cable.

### **2.2.2 Capa de Enlace**

Presenta la capa de unión entre el software y el hardware permitiendo abstraer el medio sobre el que se está transmitiendo, de esta manera no importa en que medio se encuentre siempre y cuando entienda esta capa. Con esta capa sola ya se puede realizar una comunicación pero muy primitiva.

### **2.2.3 Capa de Red**

Permite la interconexión de diferentes dispositivos sin necesidad de que estén necesariamente conectados entre ellos. Maneja la forma de comunicarlos de manera eficiente por la red.

### **2.2.4 Capa de Transporte**

Es la capa encargada de conectar el emisor y receptor a través de puertos y realizar el envío correctamente.

### **2.2.5 Capa de Sesión**

Permite mantener y controlar la comunicación y que en caso de problemas puede ser activada de nuevo sin problema.

### **2.2.6 Capa de Presentación**

Esta capa se preocupa de que los datos que lleguen de cualquier origen puedan ser reconocidos desde cualquier equipo receptor.

### **2.2.7 Capa de Aplicación**

Esta capa es la más cercana a los desarrolladores de aplicaciones donde pueden trabajar con los datos sin preocuparse de la gestión

de la conexión.

# LA PILA OSI

## **Nivel de Aplicación**

Servicios de red a aplicaciones

## **Nivel de Presentación**

Representación de los datos

## **Nivel de Sesión**

Comunicación entre dispositivos de la red

## **Nivel de Transporte**

Conexión extremo-a-extremo y fiabilidad de los datos

## **Nivel de Red**

Determinación de ruta e IP (Direccionamiento lógico)

## **Nivel de Enlace de Datos**

Direccionamiento físico (MAC y LLC)

## **Nivel Físico**

Señal y transmisión binaria

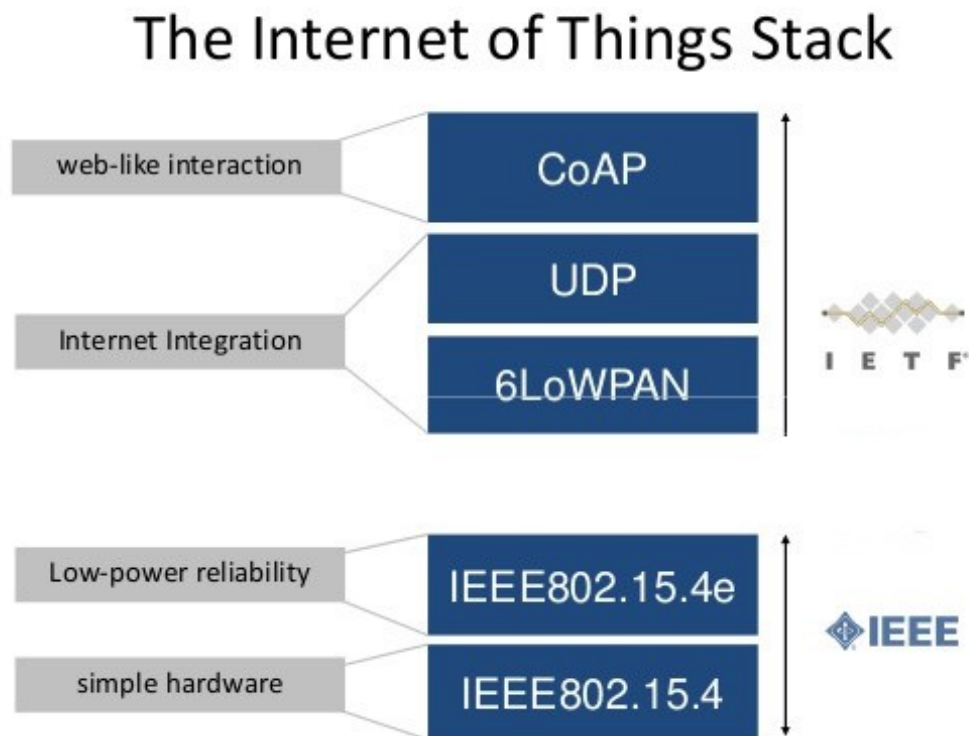
## 2.3 Entidades de normalización

### 2.3.1 IETF

Internet Engineering Task Force se compone de un grupo de personas que velan porque Internet funcione mejor generando documentos que expliquen el funcionamiento del mismo

### 2.3.2 IEEE

El Instituto de ingenieros electronicos y electricos, Institute of Electrical and Electronics Engineers en inglés, es una asociación mundial de ingenieros y técnicos cuyo objetivo es la estandarización y desarrollo en áreas técnicas, está constituida como una organización sin ánimo de lucro y dividida en sociedades especializadas en campos concretos.



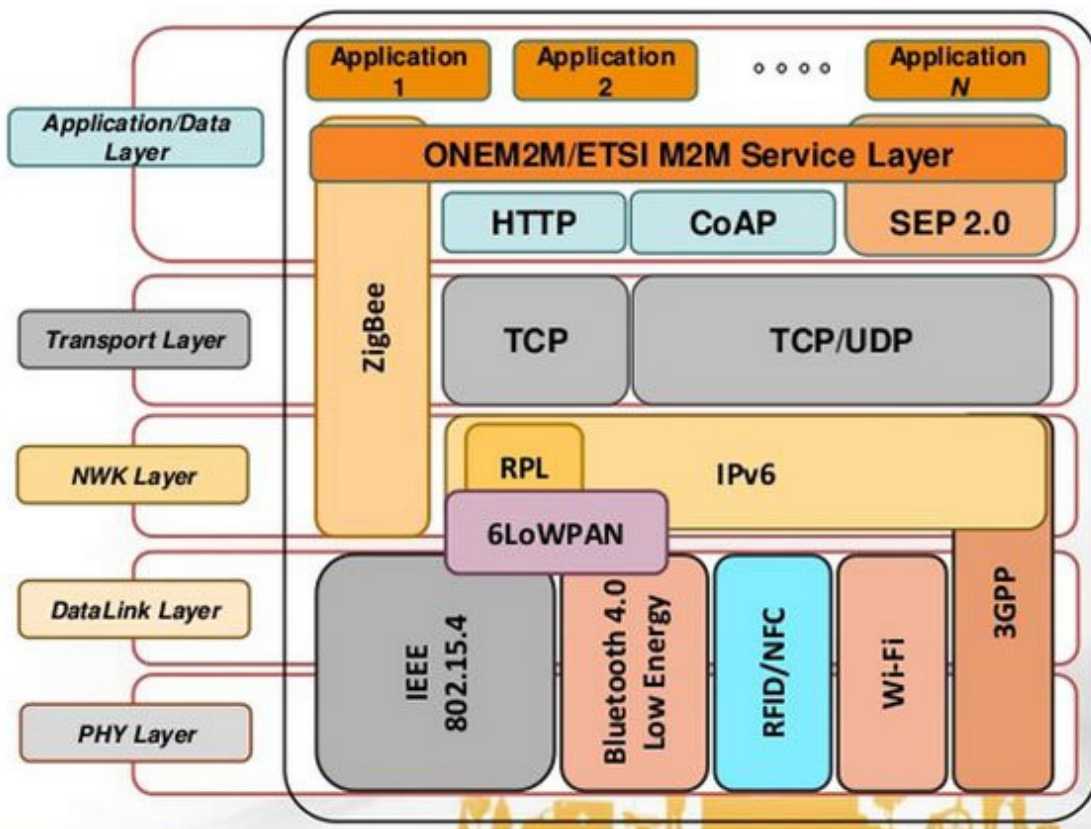
Mientras que el IEEE se encarga más de la parte física y de la correcta conexión entre los componentes, IETF trabaja para que la comunicación sea eficiente y eficaz.

## 2.4 6LoWPAN

### 2.4.1 Introducción

Es un grupo de trabajo en el área de Internet del IEFT, el nombre viene del acrónimo inglés Ipv6 Over Low power Wireless Personal Area Networks que viene siendo Ipv6 para redes de área personal inalámbricas de bajo consumo.

Tras 10 años de desarrollo se puede decir que se ha creado un protocolo que puede ser aplicado a dispositivo simples, con limitado procesamiento y baja potencia para ser integrado en el Internet de las Cosas.



Como se ve en la [Figura](#), 6LoWPAN se encuentra entre las capas de enlace y red, pudiendo llegar incluso a la capa de transporte. Compite directamente con protocolos PAN como Zigbee, Bluetooth y RFID.

### 2.4.2 Data Link Layer

Se emplea en la familia de estándares **IEEE 802**, que compone el nivel básico de red, añadiéndose sobre la capa física, permite el

control de acceso al medio de comunicación, de ahí su nombre. En el modelo OSI se encuentra ubicada en la capa dos de la jerarquía y respecto a sus **funcionalidades** se puede encontrar:

- Corrección de errores de transmisión.
- Gestionar la transmisión entre dispositivos que se encuentran en un mismo canal de comunicación, regulando el acceso al canal físico.
- Agrega flag de control a las tramas para delimitar el inicio y fin de las mismas.
- La identificación de los dispositivos viene definida por la dirección MAC.

Se pueden encontrar diferentes tipos de medios donde se pueden comunicar los dispositivos, para ello, el estándar se subdivide a su vez en más protocolos dependiendo del medio. Por ejemplo el IEEE 802.11 abarca a las redes WiFi, el IEEE 802.3 para redes Ethernet y la que se estudia que es la IEEE 802.15 para redes WPAN.

#### **2.4.2.a IEEE 802.15**

Se especializa en redes WPAN presenta una forma de interconexión de dispositivos de corto alcance, se dice que el espacio que cubren estas redes es similar al que puede abarcar la voz de una persona.

Este grupo de trabajo se puede dividir a su vez en 5 subgrupos especializados en diferentes materias. Entre los más importantes tenemos:

- IEEE 802.15.1 para redes WPAN con Bluetooth.
- IEEE 802.15.3 para redes WPAN de Alta velocidad:  
Define redes con una velocidad de transmisión de hasta 3Gb/s para la comunicación de multimedia.
- **IEEE 802.15.4 [1]** para redes PAN de Baja velocidad:  
Creada para sistemas con baja transmisión de datos, procesamiento bajo y necesaria larga duración en la alimentación. Existen diferentes estándares dentro de este protocolo, uno es Zigbee y otro **6LoWPAN** para redes WPAN en Ipv6. Este estándar define la capa física, la frecuencia, potencia, modulación y condiciones del enlace inalámbrico. Además cuenta con la capa MAC que trabaja con el formato

de los datos y con los protocolos superiores que están incluidos en la aplicación final.

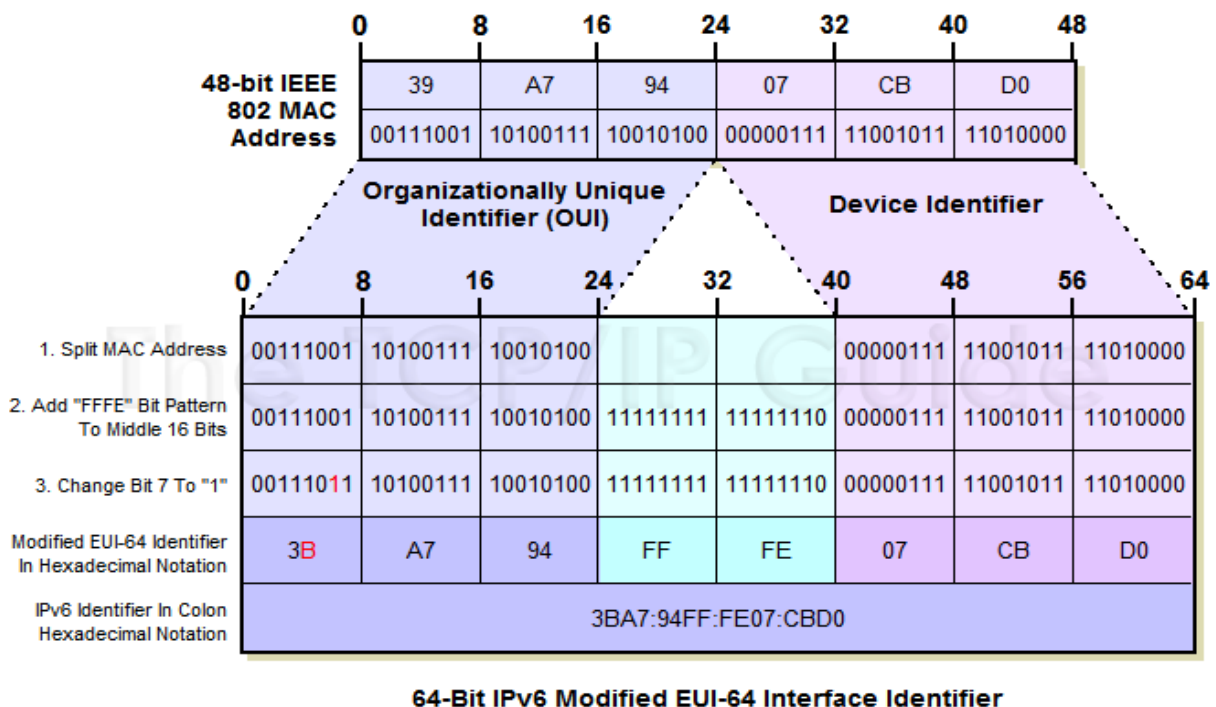
Es de interés destacar que el protocolo Zigbee comparte estandar con el protocolo que se estudia, pero es necesario pagar por usarlo.

### 2.4.2.b Dirección EUI-64 MAC

En la capa dos del modelo OSI se pueden diferenciar los dispositivos a través del identificador generado por el IEEE, el EUI-64. Está compuesto por 64 bits que permite identificar de manera unívoca un dispositivo en la red 802.15.4 (a diferencia de otras redes), esto genera la libertad de no necesitar implícitamente de dirección en la capa de red en comunicaciones básicas. De esta forma, es posible reutilizar dicha dirección en las capas superiores de red y evitamos el overhead de las cabeceras.

- Formación de la dirección EUI-64 [2]:

Se parte de la dirección MAC del dispositivo que tiene 48 bits a partir de ahí se realizan los pasos que muestra la [Figura](#).





### 2.4.2.c CSMA/CA

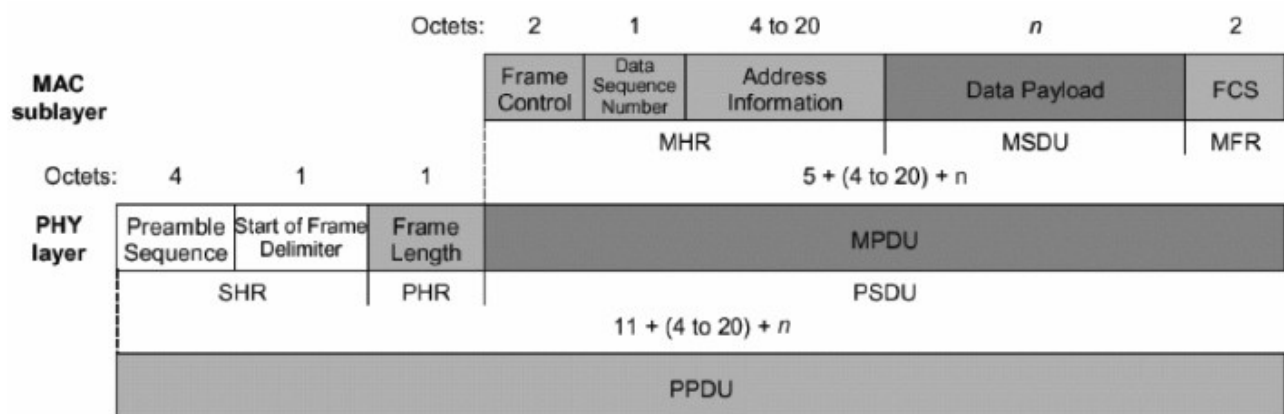
El control de flujo lo lleva la capa de enlace a través de un algoritmo que a grandes rasgos lo que realiza por nodo es la comprobación de la saturación del medio mediante la escucha previa del canal de comunicación y el envío posterior.

### 2.4.2.d Cabecera 802.15.4

Si partimos de la [Figura](#) se puede ver que existen dos partes diferenciadas, la cabecera MAC y la PHY.

La cabecera PHY es la primera cabecera que lee el dispositivo cuando le llega un paquete, contiene bits de control para la lectura del resto de la cabecera. Cuando lee el Frame Length se puede decir que sube a un capa superior en el modelo OSI siendo capaz de trabajar a capa2 (L2).

El dispositivo leerá el resto del paquete a través del byte que se le pasa anteriormente. Aquí se debe tener en cuenta que el MPDU de un paquete en 6LoWPAN debe ser menor que el MTU, es decir, el máximo número de bytes que se debe transmitir en un paquete en una red WPAN debe ser 127, este número está estipulado por el IEEE en sus documentos.



Cuando se pasa a la capa dos se debe saber que van a haber unos cuantos bytes de control y las direcciones inicio, destino. En este caso se tienen 2 bytes para el identificador de red PAN del destino junto a 8 bits de dirección EUI-64 más lo propio para la dirección fuente, esto da un total de 20 bytes.

Existe un componente de seguridad en esta capa que viene siendo la capa SAL (Security Abstraction Layer) que lo que permite

es añadir una capa de seguridad al paquete, se le añaden 20 bytes adicionales a partir de las direcciones, permitiendo dar integridad y seguridad a los datos del paquete. Esta capa permite trabajar , en algunos casos, con dispositivos específicos de cifrado contenidos dentro del hardware del mismo.

### 2.4.3 Ipv6 Network Layer

Define la capa de red del modelo OSI, permite la conectividad y la selección de ruta entre los diferentes dispositivos conectados en la red. Es capaz de conseguir que un dispositivo de origen sea capaz de comunicarse con uno destino sin que estén directamente conectados.

A modo de aclaración previa es necesario destacar que el protocolo Ipv6 que se está investigando ha sido modificado para evitar el alto número de paquetes que se envían por la red y ahorrar energía.

#### 2.4.3.a Capacidad del protocolo

Ipv6 ofrece una gran cantidad de direcciones para cada dispositivo de la orden de  $2^{128}$ , lo ideal para el objetivo que busca 6LoWPAN [3]. IETF ha elegido el protocolo IP por diversos motivos argumentados en su documento RFC 4919, donde se recopilan la información general, supuestos, planteamiento del problema y metas.

A modo de resumen, el uso de ip viene sujeto al hecho de que existen actualmente infraestructuras que usan dicho protocolo, además de que funciona y está testado. También citan que todo lo relativo al IP, es totalmente libre y abierto en cuanto a sus especificaciones. Sin olvidar mencionar que también provee de herramientas de análisis y diagnóstico para el correcto funcionamiento.

Ipv6 permite la reducción de la longitud de la dirección de 128 bits con diferentes métodos.

- Cuando se encuentran ceros juntos se sustituyen por uno solo.

FE80:CD00:**0000:0**CDE:1257:**0000**:211E:729C

FE80:CD00:**0**:CDE:1257:**0**:211E:729C

- Se pueden suprimir listas de ceros por “::”

FE80:CD00:**0000:0000:0000:0000**:211E:729C

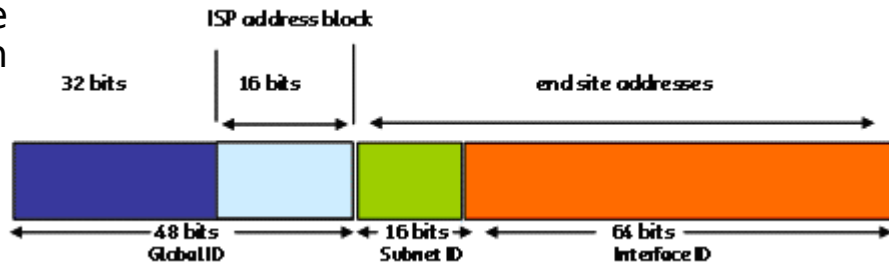
FE80:CD00::**211E:729C**

- La dirección de localhost.

**0000:0000:0000:0000:0000:0000:0000:0001**

**::1**

La subred de se encuentra implícitamente en la dirección, como se muestra en la [Figura](#).



### 2.4.3.b Clases de dirección

#### Unicast

Define la comunicación entre un par de dispositivos.

Para cada dispositivo se genera automáticamente una dirección de enlace local, se suelen usar para configuraciones iniciales y no pueden ser redireccionadas por el router.

Además existe otro tipo de dirección que se denomina site local permite direccionar todo el tráfico pero solo dentro de direcciones link local, es algo similar a una red privada en Ipv4. Estas últimas tienen como prefijo de dirección el FFC0 y las link-local con FE80.

#### Multicast

Consiste el envío de información a un grupo determinado de dispositivos. Vienen identificadas en el protocolo como las direcciones que comienzan por FF, por ejemplo FF01::1. Está estipulada ya una tabla de ips multicast que pueden ser por ejemplo las direcciones multicast para routers, link-local address, dns, etc.

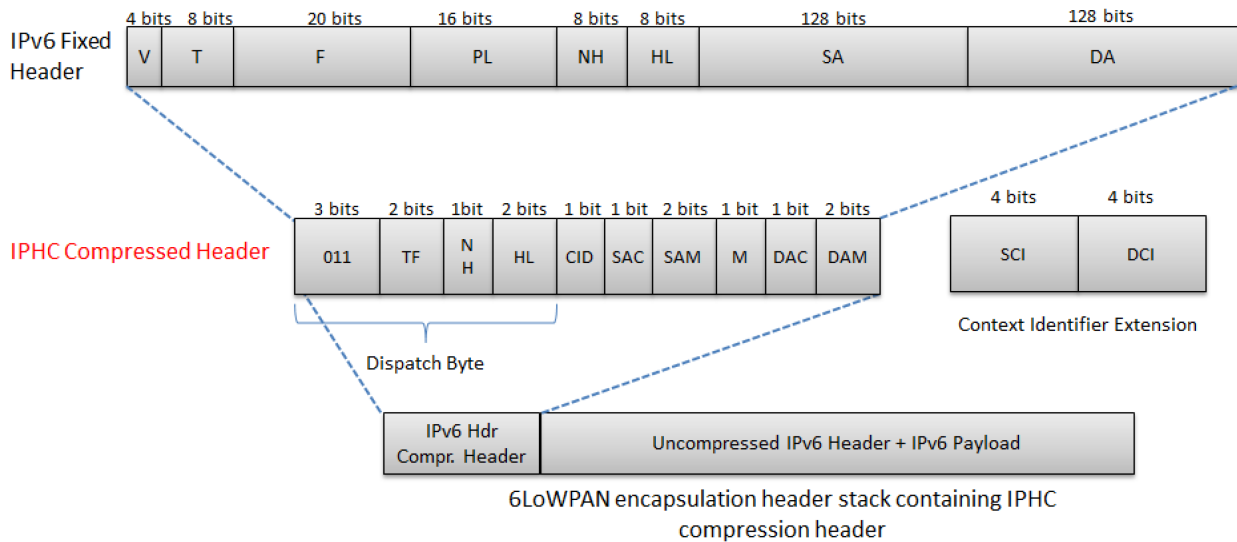
Sustituye de manera eficiente al broadcast de las Ipv4.

#### AnyCast

Presentan la novedad con respecto a Ipv4, Anycast permite hacer peticiones a cualquier dirección dentro de un Grupo Multicast. Estas peticiones se usan para descubrir vecinos o routers anexos en la red.

### 2.4.3.c Compresión de la cabecera IP

En un sistema de red con un alto exponente de dispositivos conectados es necesario reducir el uso que cada dispositivo hace de la red, para ello se ha llegado a generar un nuevo tipo de cabecera estandarizado por el IETF que permite comprimir de los 42 bytes(cabecera superior en la [Figura](#)) de tamaño de cabecera de una ipv6 normal a sólo 6 bytes en su mínima implementación.



Fueron necesarios varios intentos para generar una cabecera estable para el 6LoWPAN, se probó con la creación de la HC pero sólo funcionaba con direcciones link-local, no era lo que se esperaba y se decidió crear la IPHC. El objetivo de esta cabecera es unificar en el menor tamaño posible toda la capacidad de direccionamiento que hay en Ipv6. Para ello la cabecera cuenta con un campo Dispatch, compuesto por 3 bytes en su manera más extendida, permite saber al dispositivo qué tipo de paquete va a leer en los próximos bits. Los 3 bits más significativos seleccionan el tipo de cabecera que lo continúa.

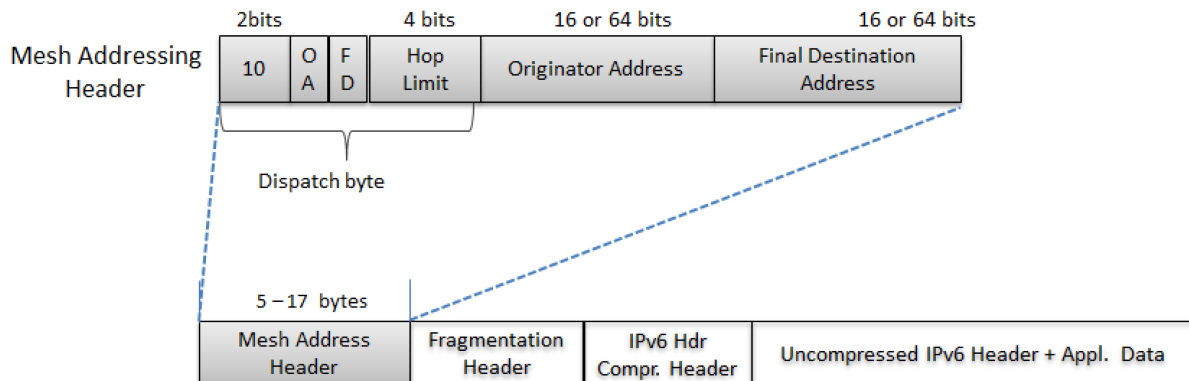
Pattern	Header Type
00 xxxxxx	NALP - Not a LoWPAN frame
01 000001	IPv6 - Uncompressed IPv6 Addresses
01 000010	LOWPAN HC1 - LOWPAN HC1 compressed IPv6
01 000011	reserved - Reserved for future use
...	reserved - Reserved for future use
01 001111	reserved - Reserved for future use
01 010000	LOWPAN BC0 - LOWPAN BC0 broadcast
01 010001	reserved - Reserved for future use
...	reserved - Reserved for future use
01 111110	reserved - Reserved for future use
01 111111	ESC - Additional Dispatch byte follows
10 xxxxxx	MESH - Mesh Header
11 000xxx	FRAG1 - Fragmentation Header (first)
11 001000	reserved - Reserved for future use
...	reserved - Reserved for future use
11 011111	reserved - Reserved for future use
11 100xxx	FRAGN - Fragmentation Header (subsequent)
11 101000	reserved - Reserved for future use
...	reserved - Reserved for future use
11 111111	reserved - Reserved for future use

Figure 2: Dispatch Value Bit Pattern

Aquí se explican los dos tipos de dirección más importante de la Figura superior:

- Mesh Addressing:

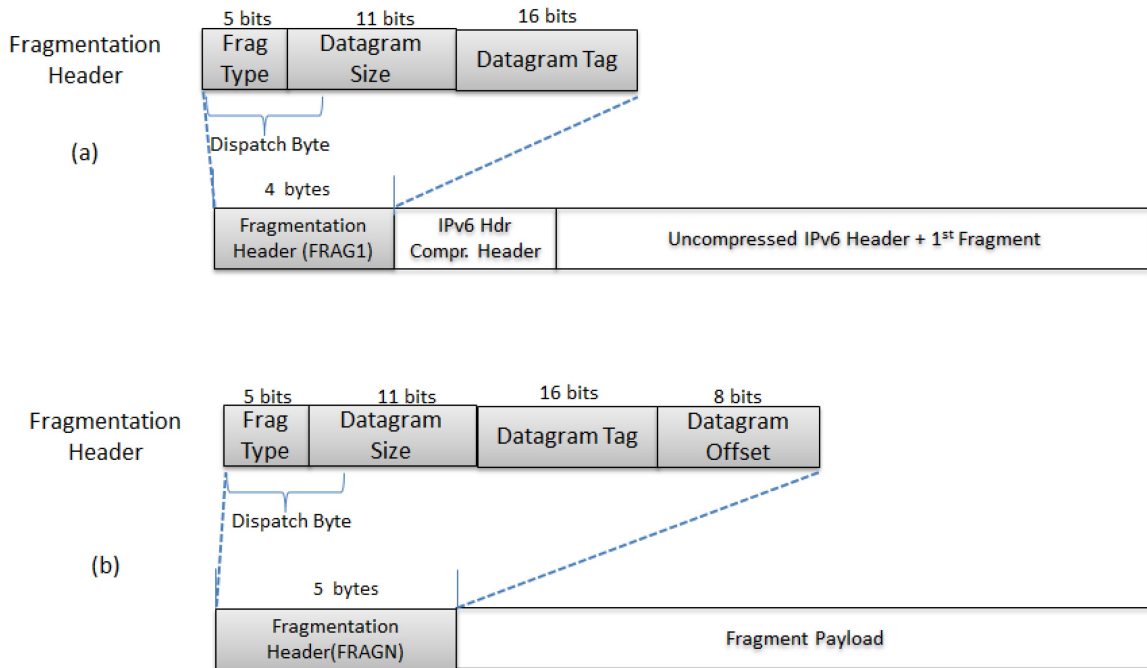
En esta cabecera entran el límite de saltos que permite la red con topología de malla. Destacar que el hop limit sería de alguna forma el campo TTL de ipv4.



6LoWPAN encapsulation header stack containing Mesh address header, fragmentation header and IPv6 Header compression header

- Fragmentation Header:

Cuando un paquete supera el MTU establecido es necesario llevar a cabo la fragmentación del paquete. El procedimiento consiste en crear un primer paquete con un tag de 16 bits, además de una codificación diferente en el dispatch. Luego los paquetes que vayan seguidos además de la cabecera del primer paquete tienen un byte de offset.

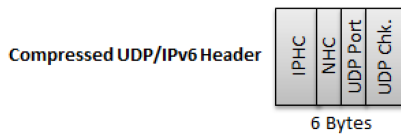


El campo NHC (Next Header Compresion) presenta el campo de compresión de cabeceras para protocolos de nivel superior en las capas OSI, así mismo, es necesario destacar que en la capa de transporte solamente se usa el protocolo UDP. El motivo es sencillo, el protocolo TCP en su máxima reducción llegaría a 20 bytes en comparación con los 8 bytes que ocupa la cabecera UDP, además, no sólo por el tamaño de la cabecera sino por su tipo de orientación a la red se puede considerar que podría ocasionar grandes saturaciones. Por ello se opta por UDP aunque la pérdida de datos puede ser un problema.

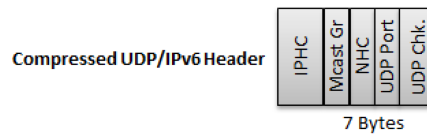
En la [Figura](#) se puede ver los diferentes tipos de cabecera que pueden acompañar a un payload, todo depende siempre del tipo de mensaje. Se debe tener en cuenta que las cabeceras tienen bits específicos que permiten saber qué tipo de dirección se toma para el envío, esta cabecera puede decir si lo próximo que vaya a leer sea una dirección que se pueda generar a través de la dirección

link-local. Además de poder pasarse la dirección entera y una comprimida.

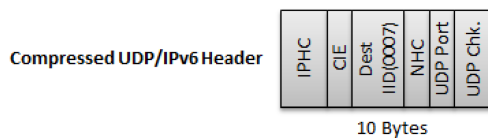
**Case I: Both Source and IPv6 addresses are link local addresses. Eg:**  
 Source IP Address = fe80::3022:001e:33fb:cc45  
 Destination IP Address = fe80::0411:0050:-56C0:1



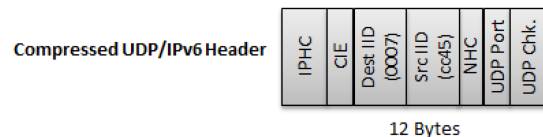
**Case II: Source IPv6 address is link-local and Destination is multicast. Eg:**  
 Source IP Address = fe80::3022:001e:33fb:cc45  
 Destination IP Address = ff02:1



**Case III: Both source and destination addresses are global addresses and source address suffix can be derived from Link Layer. Eg:**  
 Source IP Address = 2001:2312:1204:2201:3022:001e:33fb:cc45  
 Destination IP Address = 2001:4822:2d42::7



**Case IV: Both source and destination addresses are global addresses and their suffix cannot be derived from link layer header. Eg:**  
 Source IP Address = 2001:2312:1204::cc45  
 Destination IP Address = 2001:4822:2d42::7



Aquí se tiene un trozo de la definición de cabecera ipv6, se encuentra en el código que nos proporciona el proveedor Dresden Elektronik.

```
typedef struct
{
    uint8_t version_traffic;
    uint8_t traffic_flow;
    uint16_t flow;
    uint8_t len[2];
    uint8_t next_hdr;
    uint8_t hop_limit;
    ipv6_addr_t src_addr;
    ipv6_addr_t dest_addr;
} ip_hdr_t;
```

Como se puede ver no se respeta el trabajo que ha realizado el IETF para reducir el tamaño de la cabecera IP.

#### 2.4.3.d Seguridad.

El propio protocolo Ipv6 proporciona sus propias medidas de seguridad mediante IPsec.

Ipssec permite cifrar las comunicaciones además de permitir la autenticación de cada paquete IP que viaja por la red. Puede llegar a usarse entre un par de nodos, un par de gateways o entre gateways y nodos. Permite dar a la conexión entre los dos nodos la capacidad autenticación evitando un ataque de falsificación de identidad (Reply Attack).

Lamentablemente la seguridad en los sistemas embebidos tiene un coste crítico de recursos, cifrar las comunicaciones, permitir la integridad de los paquetes y la autenticidad de los mismos necesita procedimiento extra y eso conlleva un gasto energético. Por tanto las capas IPsec son opcionales en esta compresión de protocolo, lo que permite evitar las modificaciones son el checksum de la capa de enlace y el de la capa de transporte.

La capa de enlace además ofrece una capa de software en la MAC Stack que permite la interacción directa con módulos de soporte de cifrado simétrico del tráfico. Para saber si se soporta o no este servicio es necesario ver las especificaciones del fabricante, más adelante en la parte hardware del proyecto se puede ver que hay MCUs que tienen este tipo de funcionalidad.

#### **2.4.3.e Control de red.**

IPv6 define gran variedad de mejoras respecto a IPv4 entre las que tenemos:

- Descubrir routers.
- Resolución de direcciones.
- Determinación de ruta.
- Detección de direcciones duplicadas o cambios.
- Redirección y balanceo de carga.

Todo esto se logra gracias a la mejora del protocolo **ICMP** pasando ya a la versión 6.

Como se puede ver el protocolo sigue teniendo las mismas funcionalidades que en su versión padre, en cambio, se han añadido nuevos mensajes al protocolo como la anunciación del router (Router Advertisement) o la solicitud del mismo (Router Solicitation). La inclusión del direccionamiento Anycast permite, además, la solicitud de dispositivos (Neighbor Solicitation) como la promoción como dispositivo (Neighbor Advertisement).

El routing de los paquetes a través de la red es realizado por los



nodos con el rol de router, capturan los paquetes y mediante la generación del paquete Redirect permite calcular el mejor salto hacia el nodo destino.

Hay diferentes tipos de cabeceras en los ficheros fuente que da Dresden Elektronik, he detectado los siguientes tipos.

#### ICMPv6\_RS → Router Solicitation[5]

```
typedef struct
{
    uint8_t  type;
    uint8_t  code;
    uint16_t checksum;
    uint32_t reserved;
} icmp6_rs_hdr_t;
```

#### ICMPv6\_RA → Router Advertisement

```
typedef struct
{
    uint8_t  type;
    uint8_t  code;
    uint16_t checksum;
    uint8_t  hop_limit;
    uint8_t  config;
    uint16_t router_lifetime;
    uint32_t reachable_time;
    uint32_t retrans_timer;
} icmp6_ra_hdr_t;
```

#### ICMPv6\_NS → Neighbor Solicitation[4]

```
typedef struct
```

```

{
    uint8_t  type;
    uint8_t  code;
    uint16_t checksum;
    uint32_t reserved;
    ipv6_addr_t target_addr;
} icmp6_ns_hdr_t;

```

De los cuatro mensajes posibles el fabricante descarta el Neighbor Advertisement, parece porque el coste en consumo energético de este mensaje es elevado, al tener que estar enviando tramas cada tiempo estimado.

La cabecera de los paquetes ICMP es básicamente la inclusión de bits en el Next Header. Y la posterior cabecera del protocolo, como se ve en la [Figura](#), nótese de nuevo que los tamaños estipulados por el IETF no están siendo respetados en el código que ofrece Dresden Elektronik.

Compressed with GENERIC: 4 Octect Savings

```

      0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
| 1 1 1 1 1 1 1 1 |      Type      |      Checksum      |
+-----+-----+-----+-----+
|  Options ...      |
+-----+-----+

```

### 2.4.3.f Autoconfiguración.

La autoconfiguración permite liberar la carga de configuración para el administrador de red, en este caso podríamos contar con miles de dispositivos conectados y la necesidad de asignarles parámetros e identificarlos únicamente es fundamental, Ipv6 permite automatizar dicha configuración de dos maneras

diferentes.

### **Stateless**

En este tipo de configuración pueden existir dos formas:

- Ausencia de router:

Cuando el nodo se encuentra sin router lo primero que hace es autogenerarse su propia ip link-local para comunicarse a los dispositivos cercanos. El proceso es sencillo, solamente coge su EUID-64 y le concatena el prefijo link-local.

- Presencia de router:

Al contar con un router la configuración depende más de éste, es el dispositivo enrutador quien envía tramas con los prefijos de la subred en la que se encuentra el nodo, a partir de esta información es capaz de autogerar su ip-global. A partir de la generación de esta IP se pasa a la comprobación de duplicidad de ips, para evitar problemas en la red.

### **Stateful**

La autoconfiguración Stateful permite usar los servicios proporcionados por el servidor DHCPv6 o Relay Agents situados en la red . Esto permite que a partir de una configuración manual previa, la red pueda ser autoconfigurada sin la necesidad de hacer un esfuerzo posterior.

El servicio DHCP para 6loWPAN ha sido modificado de tal manera para que el consumo de los nodos sea lo mínimo posible. Para ello fue necesario reducir el número de mensajes posibles a cuatro:

- Solicit.

Permite la petición de una ip a un servidor DHCP o un relay agent cercano.

- Rebind:

Este mensaje permite alargar la vida de las direcciones que ya han sido asignadas.

- Information Request:

Este mensaje sirve para que el cliente obtenga información sin tener asignada una dirección propia.

- Reply:

Es el mensaje de respuesta que contiene las direcciones asignadas y los parámetros de configuración en respuesta de uno de los mensajes anteriores.

Los dos tipos de configuración pueden ser usados a la par, por implementación son complementarios, la ausencia de DHCP puede permitir el funcionamiento de la red por la autoconfiguración stateless. Si llegara a existir el dhcp se llevaría a cabo la autoconfiguración statefull y sería totalmente independiente de la anterior.

# Capítulo 3

## Plataforma física.

Se opta por una plataforma sólida sobre la que trabajar, en este caso, Dresden Electronics(\*) ofrece redes 6LoWPAN implementadas en su DeveloperKit(\*).

### 3.1 Familias de microcontrolador.

Antes que nada, un microcontrolador es un circuito integrado programable que tiene como propósito funcionalidades limitadas y estudiadas para un fin concreto.

#### 3.1.1.a ARM

Gama de microcontroladores de arquitectura RISC (Reduced Instruction Set Computing), la arquitectura trabaja con conjuntos de direcciones de 32 y 64 bits (dependiendo del modelo), se trata del dominante actual del mercado en telefonía móvil usado en tabletas, teléfonos móviles, videoconsolas, ... .

#### 3.1.1.b AVR

Es una familia de microcontroladores RISC de 8 bit de tamaño de palabra. Presentan todas las herramientas necesarias que permiten el buen rendimiento, 32 registros de 8 bits para el almacenamiento de variables en procesamiento, pipeline para el aprovechamiento de las unidades funcionales, puertos de comunicación con el exterior como JTAG, SPI, salidas analógicas y digitales.

### 3.2 Distribuidor de Productos.

Dresden Elektroniks es una empresa con base en Alemania que permite la compra de dispositivos hardware y software además del

montaje de placas y valoración de sistemas.

A groso modo, Dresden Elektroniks crea y desarrolla productos específicos a través de diferentes proveedores importantes en el mercado.

## 3.3 Microcontroladores.

### 3.3.1 Dispositivos ARM.

Se cuenta en este caso con el **deRFarm7** [6], es un dispositivo que cuenta con un microcontrolador **arm7tdmix512** [7] con un tamaño palabra de 32bits, además trabaja a 48MHz y cuenta con 512Kb de memoria flash y 128Kb de RAM.

En los ejemplos que trae con Dresden Elektroniks su papel fundamental es de coordinador de red, hace uso del sistema operativo para sistemas embebidos **FreeRTOS** que viene configurado para el dispositivo de tal manera que pueda hacer tareas críticas a tiempo real. Estas tareas son nateo a internet con ipv4 y tareas dentro de la red WPAN.

### 3.3.2 Dispositivos AVR

Los dispositivos AVR que se tienen son diferentes modelos de fabricación pero en esencia los mismos en cuestión de hardware.

Los **deRFmega128**[9], modelos 22A00 cuenta con un procesador **atmega128RFA1**[8] tiene un tamaño de palabra de 8 bit y funcionan a 16MHz , cuenta con una memoria de 128 KB de flash.

Una funcionalidad interesante que nos puede ofrecer este dispositivo es la capacidad de soportar el firmware OTA (Over-the-air) permite la programación del dispositivo de manera inalámbrica.

Los dispositivos pueden funcionar en modo router y en modo device.

## 3.4 Módulos de radio Integrados

Cada dispositivo tiene entre sus partes fundamentales un módulo de radio integrado junto con una antena de cerámica. Transmiten a una frecuencia de 2,4GHz y dependiendo del modelo pueden llegar a un rango hasta los 200 metros en campo abierto.

El dispositivo ARM cuenta con un módulo añadido que es el AT86RF231 al contrario que los dispositivos AVR que ya traen incrustado un Radio Transmitter RF-CMOS. Ofrece una tasa de transferencia de **2Mb/s** al igual que los módulos de atmega128rfa1. Ambos dispositivos cuentan además con un módulo AES que cifra las comunicaciones muy importantes en temas de seguridad de los datos.

Son totalmente compatibles con el stack IEEE 802.15.4 desde el 6LoWPAN al ZigBee PRO.

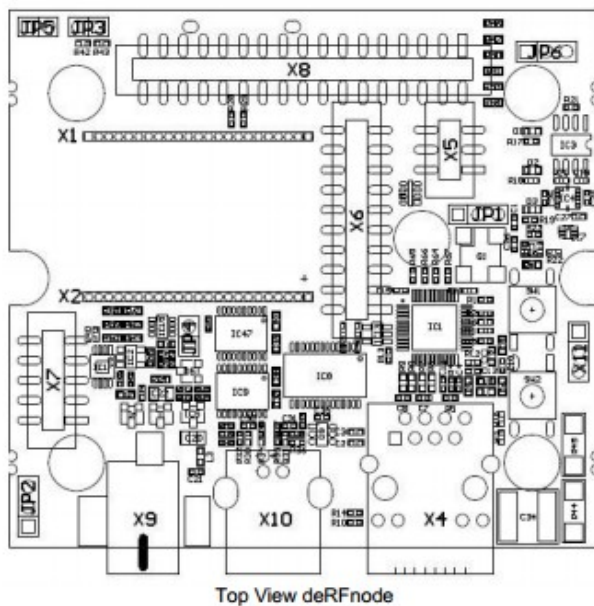
## **3.5 Placas auxiliares**

### **3.5.1 Placas de expansión de puertos**

Para evitar tiempo innecesario diseñando una capa de desarrollo se ha optado por solicitar dos placas que permiten la alimentación, el desarrollo y la comunicación hacia el exterior.

Se cuenta con dos placas una permite la comunicación mediante el puerto Ethernet hacia un dispositivo de red, su nombre es de RFgateway y la placa que permite la comunicación serial con cualquier dispositivo externo de RFnode. Ambas placas son exactamente iguales salvo porque una tiene dispositivos controladores para ethernet y otra para serial (ftdi, ..).

### Board Overlay



Top View deRFnode

### Overlay

- X1 & X2: Radio Module
- X4: Ethernet RJ45
- X5: Debug & Trace
- X6: JTAG for ARM
- X7: JTAG for AVR
- X8: 34 Pin User Interface
- X9: DC
- X10: USB Type-B
- JP1: Acc. INT or SW2
- JP2: Bat or DC/USB
- JP3: VBAT monitor
- JP4: RST Supervisor
- JP5: VCC module
- JP6: SW1

Como se ven en las especificaciones se tienen 4MB de memoria flash extra, sensores de temperatura, luminosidad y movimiento. Además nos provee de 3 LEDs y 2 pulsadores. Los puertos vienen preconfigurados en un fichero con el nombre de la placa dentro del directorio BSP de los ficheros de Dresden Elektronik.

Las placas permiten desarrollar tanto para AVR como para arm y tienen puertos JTAG diferentes para cada uno. El ARM cuenta con 20 pines (x6) y el AVR con 10 pines (x7).

Hablando de los jumper el más importante es el JP2 permite variar entre la alimentación de las baterías y alimentación desde USB.

### 3.5.2 Herramienta de desarrollo y depuración

Para realizar el proceso de inserción del programa precompilado a la memoria flash del MCU es necesario tener un dispositivo intermedio que genere las comunicaciones necesarias para cargar los datos. Para ello se ha conseguido el dispositivo **AVR JTAGICE mkIII** [12] permite comunicarse con él a través del puerto USB del PC y el puerto JTAG de la placa **deRFnode/deRFgateway** por parte del MCU. El dispositivo lo podemos usar en entornos Linux y Windows.

Además de ofrecer la posibilidad de cargar programas en la plataforma AVR se tiene, además, la posibilidad de depurar programas a un grano lo suficiente fino.



# Capítulo 4

## Desarrollo

### 4.1 Primera fase

La mayoría del tiempo de proyecto ha sido de investigación teórica parte fundamental para el desarrollo práctico consecuente de dicha investigación.

El primer objetivo marcado fue familiarizarse con el direccionamiento IPv6, como se ve reflejado en el Capítulo 2.4.3 apartados a, b y f se habla ampliamente del protocolo ipv6.

Una vez asimilados los nuevos conceptos sobre el protocolo era momento de aplicarlos, sin embargo, como explica el Modelo OSI la capa de red se encuentra a un nivel superior y es necesario seguir estudiando capas inferiores.

En ese momento se tenían dos Dispositivos AVR y era necesario además, conocer bien de cerca que ofrecen. El objetivo es intentar programar código propio precompilado para guardarlo en la memoria flash del dispositivo y al encenderlo poder ejecutarlo.

A través de la ayuda de la documentación de la placa de desarrollo, se encontró la manera de activar leds en la placa de desarrollo. En Windows y con el AVRStudio se creó un programa blink sencillo. Hasta aquí ningún problema hasta que se intentó pasar dicho programa a la flash del MCU AVR. El primer intento fue via usb, esta decisión fue tomada premeditadamente lo que llevó al retraso y a los problemas. La unidad que proveía al MCU de comunicación serial no era operativa sin una programación previa. Fue cuando se descartó la opción y se buscó otro puerto más común en la programación como es el puerto JTAG. Aquí se vio que se necesitaba un dispositivo extra y se realizaron las compras oportunas con el consecuente retraso. El dispositivo era el AVR

JTAGICE mkIII [12] (se habla en el Capítulo 3.5.2), una vez conseguido no hubo problema de cargar el código compilado en la flash y ver el resultado. Para ello fue necesario ver en qué puerto específico JTAG se conectaba y detectar el dispositivo en el software de AVR Studio.

## **4.2 Segunda fase**

El objetivo seguía siendo comunicar dos MCUs vía inalámbrica, en la primera fase se obtuvo la forma de cargar un programa en la flash, así que, solamente hay que buscar software de la capa IEEE 802.15.4.

En este caso, la propia empresa ofrece en su web la capa MAC (véase el Capítulo Herramientas en Windows) para sus dispositivos y varios ejemplos de uso. Tras las pertinentes búsquedas, se ha usado un ejemplo básico maestro-esclavo dentro del directorio de instalación en Applications/MAC\_Examples/App\_1\_NoBeacon .

Una vez cargado cada rol en cada dispositivo se vió que se realizaba la comunicación gracias al parpadeo sincronizado después de que el led del dispositivo esclavo permaneciera inactivo unos segundos al conectarle la alimentación, esto fue el indicio de que se estaba buscando maestro al que conectarse.

Otro objetivo dentro de esta fase ha sido investigar formas adicionales de desarrollo para la plataforma y se ha comprobado que también es posible desarrollar en Herramientas en Linux, tanto compilación, depuración y flasheo.

## **4.3 Tercera fase**

### **4.3.1 Pila 6LoWPAN**

Investigando en los ejemplos que se tienen en el paquete que ofrece el distribuidor, se puede ver una carpeta con el nombre SLOW en el apartado Applications. A su vez dentro se ven dos

ejemplos, en el pdf [11] explica de que va cada uno de manera extensa.

TCPPortEcho viene a ser un gateway ipv4 con NAT para desplegar redes simulando 6lo mientras que el ejemplo

Control4Slow permite la conexión ipv6 y el control con la aplicación deCON6. El interés de las redes 6loWPAN es tener redes de muchos dispositivos conectados a Internet.

Empezar por los roles que pueden realizar cada uno, entre los que tenemos tres, el rol de dispositivo, el rol de router y el rol de coordinador.

- Dispositivo:

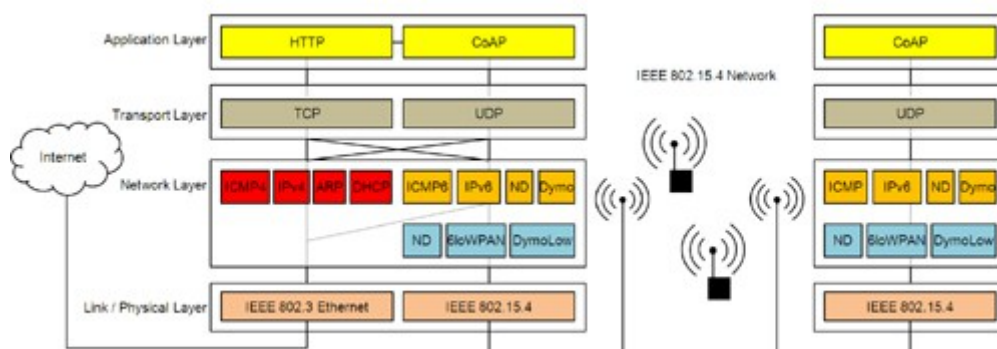
El dispositivo se encarga de la medida de algún tipo de parámetros o bien la actuación sobre algún tipo de dispositivo al que esté conectado.

- Router:

A través de los mensajes multicast dirigidos a todos los nodos alcanzables el router es capaz de promocionarse en la red, cuando un nuevo dispositivo se conecta, envía la trama de vuelta y el router almacena su local-link, para posteriores enrutamientos.

- Coordinador:

El papel del coordinador es centralizar las comunicaciones de la red, ser el nodo principal de comunicaciones.



En la [Figura](#) se puede apreciar las capas de protocolos presentes en los dos dispositivos, el coordinador a la izquierda y el dispositivo o enrutador a la derecha.

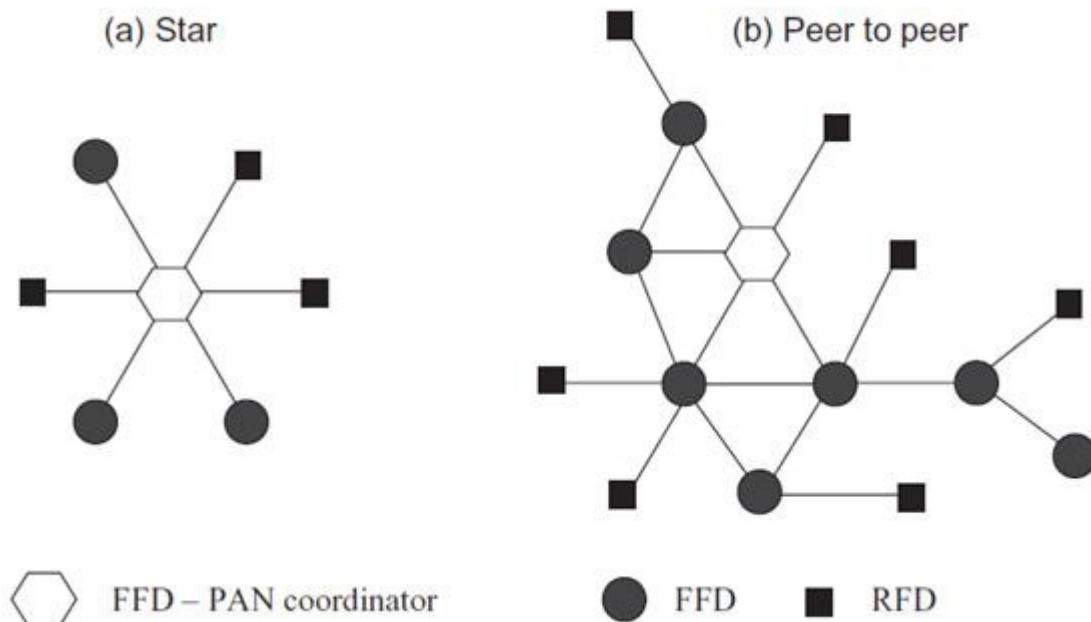
### 4.3.2 Tipos de dispositivos

- Full-function devices:

Se pueden comunicar a cualquier otro dispositivo y esta continuamente activo para servir de paso para la comunicación. Este papel lo suelen realizar coordinadores de red, trabajan con la red en IEEE 802.15.4 y luego son capaces de procesar, además del 6LoWPAN, protocolos como el Ethernet o Wifi para interconectar hacia internet. Este dispositivo es interpretado como el Coordinador del caso teórico.

- Reduced-function devices:

Son dispositivos muy simples que envían información cada cierto tiempo y teniendo siempre en cuenta el ahorro de energía. Componen a los routers y los dispositivos finales.



### 4.3.3 Topologías de red.

Las posibles topologías (ver en la Figura) que se pueden encontrar frecuentemente son las de **malla** y **estrella**. Se diferencian básicamente en el enrutamiento de los paquetes. En la topología estrella se envían los datos al nodo central, él se encarga del mantenimiento y el buen funcionamiento de la red. En redes malla lo que prima es la comunicación entre dispositivos (p2p) con el fin de comunicarse de punto a punto de la red.

Un poco dejando a parte el inciso teórico se optó por formar una topología estrella que es la que trae en los ficheros de ejemplos. En este caso, los roles habían variado respecto a la fase dos, ahora habían tres roles, se examinó el rol dispositivo y router, ambos pertenecían a la arquitectura AVR, sin embargo, el coordinador estaba establecido para una arquitectura ARM que no estaba testada.

Se buscó información técnica sobre el paso del programa ejecutable a la memoria flash del ARM ya que el programador previamente adquirido no lo soporta. Se necesitaba un dispositivo similar al JTAGICE. En este punto ya era demasiado tarde para realizar la búsqueda y la compra. En ese momento se buscaron alternativas entre las cuales estaba compilar el código fuente para AVR y conseguir un dispositivo que soportara JTAG para ARM.

Se comenzó a investigar sobre la posibilidad de compilar el código para ARM, cosa que ciertamente era bastante complicada. Tras investigar bastante tiempo se descubrió que el propio código llevaba incluido el sistema FreeRTOS, son un conjunto de librerías de código que permite programar tareas a tiempo real en sistemas embebidos. Se comprobó finalmente en las especificaciones de FreeRTOS que no había soporte para el procesador el AVR y la opción quedó descartada.

El siguiente paso fue buscar una placa con JTAG y que permitiera el desarrollo para ARM. Se tuvo la suerte de obtener una placa de depuración llamada NEO1973 V3 con puerto JTAG. La placa servía para desarrollar software para un móvil open hardware llamado OpenMoko, además, permitía ser conectado al ordenador por el puerto usb y hacer las opciones de debugger.

Para usar esta placa y que fuera efectiva, se tuvo que buscar un software que manejando por usb hiciera las operaciones necesarias para cargar un binario en la flash del ARM. Fue cuando OpenOCD encajó perfectamente, es más, vienen preparados los ficheros para usarse con el modelo exacto del MCU, en este caso, ARM7TDMIX512. Sin embargo los constantes intentos para llevar a cabo el paso del binario al microcontrolador han sido erróneos y no se ha podido llevar a cabo el objetivo que en un principio se había marcado.

## 4.4 Apoyo.

### 4.4.1 Herramientas en Windows

AVR Studio 6 ofrece soporte para toda su gama de dispositivos AVR, permite generar los ficheros binarios que se van a ejecutar, además de ofrecer la interfaz de depuración y flasheo.

Para instalar la pila 6loWPAN es necesario seguir los pasos que salen en la página de Dresden Elektronik[10]. Son dos paquetes ejecutables que deben ser ejecutados por orden y que cargan ficheros en el pc además de ejecutar scripts para la compilación de ejemplos.

En el pdf [11] que ofrece Dresden Elektronik se puede ver paso a paso como insertar un programa en el dispositivo AVR, Páginas 53-57.

### 4.4.2 Herramientas en Linux

En linux se pueden tener las mismas opciones que en Windows con la diferencia de que todas las herramientas se encuentran en línea de comandos. Lo que aquí se explica es la comprobación del dispositivo JTAGICEmkIII para su correcta utilización.

Los paquetes fundamentales que se deben tener instalados para la compilación cruzada de las aplicaciones que se programen son:

- libusb-1.0-0 (Instalada por defecto en el sistema)
- libusb-dev
- libusb-1.0-0-dev
- gcc-avr
- avrdude (para la programación de placas)
- avarice (para la depuración por JTAG)

Para instalar los diferentes paquetes, en debian, sólo será necesario sustituir el final de la siguiente sentencia bash por el nombre del paquete.

```
sudo apt-get install XXXXXX
```

Uso de **avrdude** para comprobar la detección de la placa.

Conecta el usb y compruebe con **lsusb**

Resultado: ... ID 03eb:2110 Atmel Corp.

Insertar el comando:

**Ejecuta:** `"sudo avrdude -c jtag3 -p m128rfr2"`

Si la opción que se pone en "-c" o "-p" es diferente del registro de placas ISP compatibles, saldrá una lista con ese registro para la posterior corrección.

Salida correcta:

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ###...##### 100%
```

```
avrdude: Device signature = 0x1ea701
```

```
Double check chip, or use -F to override this check.
```

```
avrdude done. Thank you.
```

# Capítulo 5

## Conclusiones y líneas futuras

Se ha visto en los puntos anteriores la gran variedad de parámetros que se tienen en cuenta para las redes 6LoWPAN, desde el rendimiento y el software de los dispositivos que componen una red, pasando por conocer la saturación del medio hasta el consumo que cada dispositivo. El papel fundamental del protocolo a parte de dar conectividad ipv6 es abstraer al usuario que programa aplicaciones para iot de lo mencionado anteriormente.

Está claro que en el mercado hay y habrán gran variedad de dispositivos que permiten usar este tipo de protocolos. Es necesario llevar adecuadamente y de manera correcta todos los puntos tratados anteriormente. Un problema que hubo y que me parece muy grave es que la pila de protocolo que da Dresden Elektroniks no viene adecuada a lo que propone el IETF, sería una posible línea de trabajo mejorar este punto.

Dependiendo de la aplicación que se le de a este tipo de tecnologías será más o menos importante el valor de los datos que se emiten por el aire, en este caso se estudia el protocolo pero si se estudiara la aplicación, la seguridad sería un punto crítico, los ataques Man In The Middle (MITM) y Denegación de Servicio (DOS) podrían ser muy usados para destruir el valor de una red de Internet de las Cosas.

Los dos puntos hablados anteriormente tratan básicamente del desarrollo software, el hardware está más que trabajado y funciona perfectamente, el problema viene a ser la dispersión en el mercado de los fabricantes por buscar un protocolo privado. Esto hace que a cualquier desarrollador que quiera crear algún proyecto personal le sea muy difícil realizarlo.



# Capítulo 6

## Summary and Conclusions

It is seen in the above range of the parameters taken into account for 6LoWPAN networks, from performance and software devices that make a network, to meet the saturation of the medium to the consumption of each device. The fundamental role of the protocol of IPv6 moreover than connectivity, is to abstract the user to program applications for iot.

It is clear that in the market have and will make variety of devices that use this type of protocols. You need to properly and correctly carry all the points discussed above. There was a problem and I think very serious is that the protocol stack that gives Dresden Elektroniks is not appropriate to the proposals of the IETF, would be a possible line of work to improve this point.

Depending on the application to be given to this type of technology will be more or less important to the value of the data that are emitted into the air, in this case the protocol is studied but if the application is studied, the security would be a critical point , attacks Man In The Middle (MITM) and Denial of Service (DOS) could be widely used to destroy the value of a network of Internet of Things.

The point spoken previously talked basically software development, hardware is more than worked and works perfectly, the problem becomes dispersion in the market for manufacturers to seek a private protocol. This means that any developer who wants to create a personal project will be very difficult to do.

# Capítulo 7

## Presupuesto

Se encuentran definidos los dispositivos electrónicos que se han usado durante el desarrollo.

<b>Unidades</b>	<b>Nombre</b>	<b>Precio/u</b>
1	ARM   deRFarm7-25A00	60,16 €
2	deRFmega128-22A00	27,98 €
1	deRFgateway for ARM	76,52 €
2	deRFnode for AVR / ARM	74,08 €
1	AVR   deRFmega256-23T00	29,21 €
1	JTAGICE mkIII	123,40 €
		391,35 €

**Tabla 7.1: Presupuesto**

# Capítulo 8

## Bibliografía.

[Data Link Layer]

[1] [http://en.wikipedia.org/wiki/IEEE\\_802.15.4](http://en.wikipedia.org/wiki/IEEE_802.15.4)

[2] <https://standards.ieee.org/develop/regauth/tut/eui64.pdf>  
[http://www.tcpipguide.com/free/t\\_IPv6InterfacelIdentifiersandPhysicalAddressMapping-2.htm](http://www.tcpipguide.com/free/t_IPv6InterfacelIdentifiersandPhysicalAddressMapping-2.htm)

[6LoWPAN]

[3] <http://tools.ietf.org/html/rfc4919>

<http://tools.ietf.org/html/rfc6282>

<http://tools.ietf.org/html/rfc4944>

<http://www.slideshare.net/rjain51/m-19lpn>

<http://www.mdpi.com/2224-2708/2/2/235/htm>

[Control de red.]

[4] <https://tools.ietf.org/html/draft-oflynn-6lowpan-icmphc-00#section-5.2.1>

[5] <http://tools.ietf.org/html/rfc4861>

[Microcontroladores.]

[6] <https://www.dresden-elektronik.de/funktechnik/products/radio-modules/eval-derfarm7/description/?L=1>

[7] <http://www.atmel.com/Images/doc6120.pdf>

[8] [http://www.atmel.com/Images/Atmel-8266-MCU\\_Wireless-ATmega128RFA1\\_Datasheet.pdf](http://www.atmel.com/Images/Atmel-8266-MCU_Wireless-ATmega128RFA1_Datasheet.pdf)

[9] <https://www.dresden-elektronik.de/funktechnik/products/radio-modules/avr-single-chip->

[modules/description/?L=1&cHash=c9c902ccdb43164696acccf81b62b2bd](#)

[10]<https://www.dresden-elektronik.de/funktechnik/products/software/sources-stacks-firmware/6lowpan/?L=1>

[11][https://www.dresden-elektronik.de/funktechnik/products/boards-and-kits/development-kits/derfdevelopmentkit-6lowpan/description/?L=1&elD=dam\\_frontend\\_push&docID=2124](https://www.dresden-elektronik.de/funktechnik/products/boards-and-kits/development-kits/derfdevelopmentkit-6lowpan/description/?L=1&elD=dam_frontend_push&docID=2124)

[12]<http://www.atmel.com/tools/jtagice3.aspx>