

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



**Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación**

TRABAJO FIN DE GRADO

**Análisis de soluciones inalámbricas para la creación de un
sistema de comunicaciones para su uso en dispositivos IoT**

**Adrian Mihai Rosu
Tutor: Fernando Jesús López Colino**

SEPTIEMBRE 2019

Análisis de soluciones inalámbricas para la creación de un sistema de comunicaciones para su uso en dispositivos IoT

AUTOR: Adrian Mihai Rosu
TUTOR: Fernando Jesús López Colino



Hardware & Control Technology Laboratory

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre de 2019

Resumen

Este Trabajo Fin de Grado está basado en el análisis de tecnologías inalámbricas para su implementación en un sistema de comunicaciones para su uso en dispositivos de IoT (Internet Of Things), el llamado internet de las cosas que consiste, como su propio nombre indica, en habilitar a diferentes objetos con la capacidad de intercambiar datos utilizando una red como por ejemplo Internet. Para hacer esta interacción posible, existen varios protocolos de comunicación inalámbrica disponibles de los cuales se hará su análisis de sus principales características como: consumo, alcance y velocidad de transmisión y se escogerá uno de ellos para realizar una prueba de concepto.

Para la elección del protocolo, se seleccionará un dispositivo asociado a cada uno de los protocolos analizados. El consumo de estos dispositivos será crucial para la elección del protocolo bajo el que funcione. Una vez elegido, la prueba de concepto consistirá en tener un elemento central conectado a un servidor de internet, que recogerá las peticiones del usuario y mandará las órdenes al resto de dispositivos que serán móviles. En este caso en particular, la prueba se ha realizado con un dispositivo central y otro dispositivo móvil. Al finalizar tendremos una página web accesible para el usuario, con la cual podrá interactuar con todos los dispositivos dispuestos.

Palabras clave

Inalámbrico, SiPy, Sigfox, FSK (Frequency Shift Keying).

Abstract

This Bachelor Thesis is based on analysing wireless technologies in order to implement them into a communication system which is made to be used in IoT (Internet Of Things) devices, which consist of enabling different objects with a connection to a network such as Internet as they can communicate with each other. In order to make this possible, there are several available protocols whose main characteristics will be analysed such as: power consumption, range and data rate and then one of them will be chosen to carry out a demonstration.

For the protocol election, one device has been chosen for each analysed protocol. The consumption of these devices will be crucial for selecting the protocol that it uses. Therefore, having chosen the device and the protocol, the demonstration will be about having a central device connected to a web server on the Internet which will be gathering users requests and sending them to the rest of the devices. In particular, we will have two devices: one will be the central device and the other one will be mobile. The mobile device can be placed in a measured range around the central one. Finally, a web page will be available for the user. This page will be able to interact with every device attached.

Keywords

Wireless, SiPy, Sigfox, FSK (Frequency Shift Keying).

Agradecimientos

Quiero empezar agradeciendo la oportunidad de realizar este Trabajo de Fin de Grado a mi tutor Fernando.

Ahora creo que todas las personas que han formado parte de mi día a día durante todos estos años de carrera se merecen mis agradecimientos. Gracias a todas estas personas que me han aportado su granito de arena para llegar a ser siempre mejor persona que el día anterior. Aquí menciono desde a aquellas mujeres de la limpieza que me han dejado pasar al baño, aunque el suelo estuviera recién fregado, hasta a los técnicos de las peceras que han tenido que esperar miles de veces a que pudiera finalizar las prácticas para poder cerrar el laboratorio. Simplemente quiero agradecerles a todos ellos el hacerme más llevadero el día a día en esta escuela.

No cabe duda de que en estos años he realizado muchísimas amistades y he conocido a infinidad de compañeros. Quiero mandar un saludo a todos ellos y decirles que la más mínima conversación que he compartido con ellos también forma parte de mí. En especial me gustaría mencionar a: Jesús, Lucas, Cristian, Jorge, David, Víctor y Patricia. Mis principales compañías cuando el hambre entraba en juego y con los que tantas conversaciones profundas y maduras he compartido en la postmesa.

Quiero dedicar este párrafo a una chica muy especial que supuso un gran cambio en mi forma de ser y en mi forma de pensar. Su nombre, Patricia. Gracias Patri por ser como eres y por haber provocado ese gran cambio en mí, a mejor por supuesto, y por haber sido mi compañera durante estos últimos cuatro años, y los que quedan. Estoy seguro de que sin ti todo este camino hubiera sido enormemente más complicado y aburrido, gracias por estar ahí cuando te he necesitado y espero que lo sigas haciendo.

Por último, pero no menos importante. Estoy muy agradecido a mis padres por todo el sacrificio que les he supuesto desde que nací y todos los malos momentos que tuvieron que soportar para garantizarme un futuro mejor. Gracias a ellos estoy donde estoy a día de hoy, gracias a su sacrificio y a su empeño en que me convirtiera en una buena persona. Me siento muy afortunado de haberme convertido en lo que soy hoy y dedico todo mi esfuerzo en no decepcionarles e intentar demostrarles que todo su sacrificio no ha sido en vano. Gracias por los valores que me habéis inculcado desde pequeño y gracias a Dios por cuidar de mí en los momentos más difíciles.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Organización de la memoria.....	2
2	Estado del arte	3
2.1	Topologías de red	3
2.1.1	Red en malla	3
2.1.2	Red punto a punto.....	3
2.1.3	Red en estrella	3
2.1.4	Red en árbol.....	3
2.2	Non-IP Based WPAN.....	4
2.2.1	Bluetooth	4
2.2.2	Zigbee	6
2.2.3	Z-Wave	6
2.2.4	LiFi	7
2.3	IP-Based WPAN y WLAN.....	7
2.3.1	6LoWPAN	7
2.3.2	Thread.....	8
2.3.3	Protocolos IEEE 802.11 y WLAN	9
2.4	Sistemas Long-Range Communication y Protocolos (WAN).....	10
2.4.1	Cellular connectivity.....	10
2.4.2	WiMAX	11
2.4.3	LoRa y LoRaWAN.....	12
2.4.4	Sigfox	12
2.5	Resumen protocolos	14
2.6	Análisis protocolos	14
3	Diseño.....	17
3.1	Comunicación.....	18
3.1.1	SiPy.....	18
3.1.1	Pantalla OLED 0,91” y HTU21D.....	19
3.1.2	FSK Modulation	19
3.1.3	UART (Universal Asynchronous Receiver-Transmitter).....	19
3.1.4	I ² C (Inter-Integrated Circuit).....	19
3.2	Servidor Web.....	20
3.2.1	HTTP (Hypertext Transfer Protocol) Apache	20
3.2.2	PHP (PHP: Hypertext Preprocessor)	20
3.3	Lenguajes de programación.....	20
3.3.1	Micropython	20
3.3.2	HTML y CSS.....	20
4	Desarrollo	21
4.1	Envío de mensajes Sigfox.....	21
4.1.1	Device2device.py	22
4.2	RGB Led.....	22
4.2.1	Rgb.py	23
4.3	Pantalla OLED 0,91”	23
4.3.1	Screen.py	23

4.4 HTU21D	23
4.4.1 Sensor.py	23
4.5 UART	23
4.5.1 Reception.py	23
4.6 Servidor Web	24
4.6.1 Index.php	24
4.6.2 Serial.php	25
4.7 SiPy-1	25
4.8 SiPy-2	25
5 Integración, pruebas y resultados	27
5.1 Alcance comunicación radiofrecuencia	27
5.2 Tasa real de transferencia	27
5.3 Tiempos de acción	27
5.3.1 Envío FSK SiPy-1 a SiPy-2	28
5.3.2 Actuación sensor I2C	28
5.3.3 Respuesta FSK SiPy-2 a SiPy-1	28
5.3.4 Tiempo total SiPy-1 a SiPy-2	28
5.3.5 Tiempo respuesta usuario	28
6 Conclusiones y trabajo futuro	29
6.1 Conclusiones	29
6.2 Trabajo futuro	29
Referencias	31
Glosario	35
Anexos	I
A Manual de instalación	I
Actualizar dispositivo	I
Activación en Sigfox	I
Entorno de desarrollo	III
PuTTY	III
Conexión WiFi	IV
Conexión FTP	V
Máquina virtual	VII
Servidor Web	VIII
B Montaje	IX
Pantalla OLED 0,19"	IX
HTU21D	X
UART	X

INDICE DE FIGURAS

FIGURA 1- RESUMEN PRINCIPALES TOPOLOGÍAS	4
FIGURA 2-TOPOLOGÍAS BLUETOOTH	5
FIGURA 3-TOPOLOGÍA LiFi	7
FIGURA 4-RED 6LOWPAN [31]	8
FIGURA 5-TEORÍA CELULAR [68]	10
FIGURA 6- TOPOLOGÍA DE SIGFOX [44]	13
FIGURA 7-ESQUEMA DEL DISEÑO INICIAL	17
FIGURA 8-ESQUEMA DEL DISEÑO FINAL	17
FIGURA 9-DISPOSITIVO SiPY [46]	18
FIGURA 10-ANTENA SIGFOX	18
FIGURA 13-PÁGINA PRINCIPAL DEL SERVIDOR WEB	24
FIGURA 14-ACTUALIZACIÓN SiPY	I
FIGURA 15-PORTAL REGISTRO DE USUARIO SIGFOX [63]	II
FIGURA 16-PORTAL REGISTRO DISPOSITIVO SIGFOX [64]	II
FIGURA 17-PAQUETE A INSTALAR EN ATOM.....	III
FIGURA 18-CONFIGURACIÓN CONEXIÓN PUTTY WINDOWS	IV
FIGURA 19-SALIDA DEL TERMINAL PUTTY	IV
FIGURA 20-RED WIFI DEL DISPOSITIVO SiPY	V
FIGURA 21-CONFIGURACIÓN DE SITIO EN FILEZILLA GENERAL.....	V
FIGURA 22-CONFIGURACIÓN DE SITIO EN FILEZILLA OPCIONES DE TRANSFERENCIA	VI
FIGURA 23-DIRECTORIOS CONTENIDOS EN EL DISPOSITIVO SiPY	VII
FIGURA 24-CONFIGURACIÓN REENVÍO DE PUERTOS	VIII
FIGURA 25-SiPY PINOUT [46].....	IX
FIGURA 26-CONEXIONES PANTALLA OLED 0.91”	IX
FIGURA 27-CONEXIONES SENSOR HTU21D.....	X

FIGURA 28-CONEXIÓN UART X

INDICE DE TABLAS

TABLA 1-CLASES DE BLUETOOTH [12] [13] [14] [15].....	5
TABLA 2-RESUMEN PROTOCOLOS 802.11 [36].....	9
TABLA 3-RESUMEN TECNOLOGÍAS MÓVILES [39].....	11
TABLA 4-RESUMEN CARACTERÍSTICAS PROTOCOLOS	14

1 Introducción

1.1 Motivación

Esta memoria de TFG (Trabajo de Fin de Grado) tratará sobre el análisis de diferentes tecnologías que los dispositivos usan para comunicarse en el mundo del IoT (Internet of Things). Actualmente se abusa de enlaces que utilizan una conexión basada en IP (Internet Protocol), lo que conlleva un gasto extra de energía y de información, como se verá más adelante cuando analicemos los dispositivos que hacen uso de esta tecnología. Debido a que estar conectado a internet supone una recepción constante de paquetes que circulan en la misma red y que no van dirigidos a estos dispositivos, sino a otros que se encuentren dentro de ella [1].

Con el tiempo, cada vez más y más elementos se unirán al IoT, existiendo un total de 26,66 mil millones de dispositivos y se prevé un incremento de hasta 75,44 mil millones en torno al año 2025 [2], y desde luego que las conexiones basadas en IP no serán las óptimas para casos como: sensores de luz que envíen información acerca del nivel lumínico y así activar las farolas, sensores de humedad que determinen cuándo se ha de regar el césped, etc. Debido a que para este tipo de aplicaciones se necesitarán sensores que no gasten tanta energía recibiendo paquetes que no van dirigidos a ellos mismos y que sigan funcionando aun cuando no lo haga internet.

La motivación de este TFG es analizar las formas más comunes de comunicación entre los dispositivos y mostrar los pros y los contras, enfocándonos en el consumo energético, el alcance y varios aspectos principales, para determinar cuál de estos métodos es el óptimo.

1.2 Objetivos

El objetivo principal de este proyecto es elegir el protocolo de comunicaciones inalámbricas adecuado para su implantación en una red de sensores.

Para ello, se usará el software oficial del dispositivo y ciertas librerías de otros autores para los sensores externos que se conectarán al dispositivo. La prueba de concepto consistirá en demostrar la comunicación entre dos dispositivos, manejados desde un servidor web, montado en un ordenador. Con todo ello, se conseguirá tener una página web sencilla para el usuario que le permitirá interactuar con los dispositivos, habilitando el envío y recepción de información.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- En el primer capítulo se ha introducido brevemente los objetivos y la motivación de realizar este trabajo de fin de grado.
- En el segundo capítulo se expone el análisis del estado del arte, investigando las múltiples opciones que existen para solventar la comunicación inalámbrica entre varios dispositivos. Destacando las características principales de las mismas como: consumo energético, alcance, velocidad de transmisión y topología.
- En el tercer capítulo se explica el diseño que se va a seguir tras haber hecho el previo análisis en el estado del arte. Se muestra la información necesaria sobre las herramientas que se van a usar en el desarrollo del proyecto.
- En el cuarto capítulo se muestra el desarrollo de cada parte del proyecto y cómo se ha decidido realizar su implementación.
- En el quinto capítulo se realizan las pruebas necesarias para comprobar el correcto funcionamiento del proyecto y medir las tasas de transmisión reales en las diferentes etapas a la hora de la implementación.
- En el sexto capítulo se concluye con los aspectos que más problemas han causado en su desarrollo y posibles alternativas para continuar en un trabajo futuro sobre este proyecto.
- Por último, se añaden los anexos y referencias.

2 Estado del arte

En este apartado se investigará acerca de las tecnologías y protocolos existentes que se usan para llevar a cabo una comunicación inalámbrica en dispositivos de IoT (Internet of Things). Para ello, he tomado como referencia un libro que trata sobre la arquitectura de IoT [3].

2.1 Topologías de red

Antes de exponer las diversas maneras de comunicación inalámbrica en dispositivos IoT, vamos a entender las topologías de red que la mayor parte de estas tecnologías comparten.

Una topología de red es la estructura que se sigue para organizar la comunicación entre una serie de dispositivos. Dentro de una topología de red existe la parte física y la parte lógica. La parte física define la manera en la que van a ir conectados los dispositivos y la parte lógica define la manera en la que se va a gestionar el envío de información de datos [4]. Tomaremos como referencia las principales topologías que existen, expuestas a continuación, durante el transcurso del informe.

2.1.1 Red en malla

La red en malla consiste en conformar una red en la que cada uno de los nodos esté conectado a todos los demás nodos que conforman la red [5]. Este tipo de configuración posibilita más de un camino entre un nodo y otro, de manera que en el caso de que falle alguna conexión se podrá llegar por otro camino. De la misma manera, si uno de los nodos falla la red sigue operativa ya que los demás esquivarán el paso por aquel nodo que ha fallado [6].

2.1.2 Red punto a punto

Las redes punto a punto consisten en un canal creado para la comunicación entre dos nodos. Ambos nodos poseen las mismas funciones, pudiendo ser maestro o esclavo dependiendo del sentido de la comunicación. El canal creado para este tipo de comunicación puede ser de tres tipos diferentes teniendo en cuenta cuál es el sentido de la transmisión de datos [7]:

- **Simplex:** la comunicación se realiza en un único sentido.
- **Half-duplex:** la comunicación se realiza en ambos sentidos. Sin embargo, dicha comunicación no se puede dar simultáneamente en ambos nodos.
- **Full-duplex:** la comunicación se realiza en ambos sentidos y también puede hacerlo simultáneamente.

2.1.3 Red en estrella

En una red de estrella existe un nodo central, llamado concentrador, con el que se conectan el resto de los nodos dispuestos en esta red. De esta manera, se crea un canal de comunicación entre el concentrador y cada uno de los nodos. En este modelo, el concentrador es el que decide la información que pasa de un nodo a otro. Esta es la topología que más se usa [8].

2.1.4 Red en árbol

La red en árbol está basada en la red en estrella. La principal diferencia que existe entre ambas es que en la red de estrella todos los elementos están conectados a un único concentrador, mientras que en la red en árbol no todos los nodos deben estar conectados al mismo concentrador. En la red de árbol se pueden encontrar varios concentradores que

distribuyen los nodos de maneras diferentes y a su vez esos concentradores están conectados al concentrador central [9].

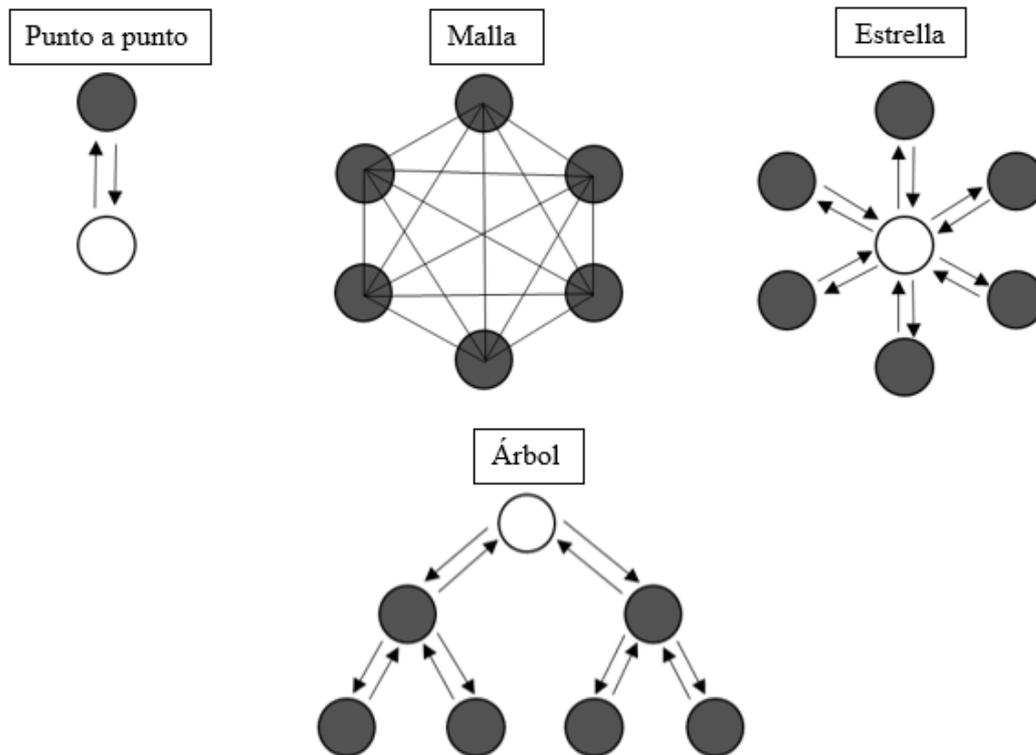


Figura 1- Resumen principales topologías

2.2 Non-IP Based WPAN

Los sensores y demás dispositivos conectados a internet necesitan un método para transmitir y recibir información. En el ámbito del IoT, las comunicaciones entre los sensores o actuadores se pueden realizar a través de WPAN (Wireless Personal Area Network), que consiste en una red ideada para que diferentes dispositivos se comuniquen con un punto central de manera inalámbrica dentro de un rango cercano. Las WPANs pueden ser IP (Internet Protocol) y Non-IP, es decir pueden estar construidas sobre internet o no.

Los sistemas que utilizan Non-IP optimizan el coste y la energía de uso [10].

2.2.1 Bluetooth

Es una tecnología que se ha tenido un uso extensivo en el mundo del IoT, principalmente en dispositivos en modo LE (Low Energy) para la transmisión de balizas, sensores inalámbricos, sistemas de alarma. Bluetooth y todos sus componentes siempre se han desarrollado bajo licencia GPL (General Public License) y son esencialmente libres de uso [11].

Los dispositivos Bluetooth se clasifican en varias clases basadas en la potencia de transmisión:

Clase	Potencia máxima (mW)	Potencia máxima (dBm)	Alcance máximo (metros)
Clase 1	100	20	~100
Clase 2	2,5	4	~10
Clase 3	1	0	~0,1
Clase 4	0,5	-3	~0,5

Tabla 1-Clases de Bluetooth [12] [13] [14] [15]

La topología que sigue Bluetooth está compuesta por piconets y scatternets [16]:

- **Piconet:** Los dispositivos funcionan como maestro-esclavo. Puede haber un solo esclavo o múltiples, pero siempre debe haber un único maestro que determine el canal y la fase.
Un piconet tiene tres bits de direcciones, entonces solo puede haber un máximo de ocho dispositivos, incluyendo al maestro.
- **Scatternet:** Consiste en la unión de uno o más piconets. Los dispositivos de un piconet pueden tomar la función tanto de maestro como de esclavo en otro piconet del mismo scatternet.

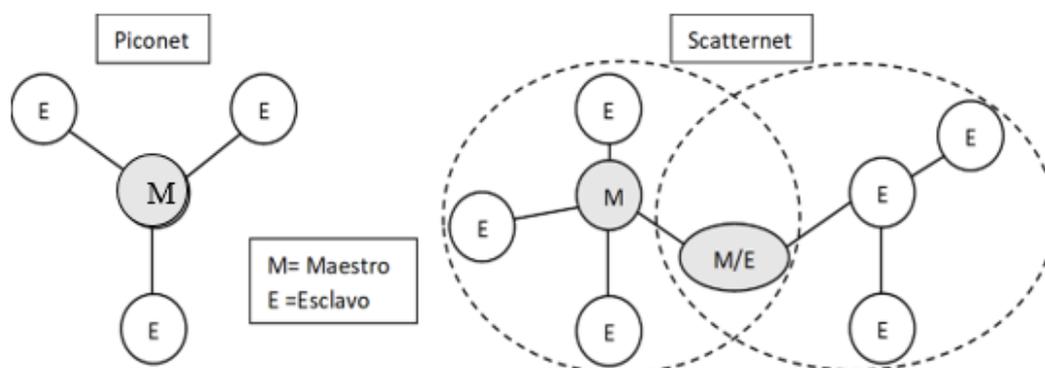


Figura 2-Topologías Bluetooth

En Bluetooth existen básicamente tres modos de funcionamiento [17]:

- **Low Energy mode (LE):** Usa la banda ISM (Industrial, Scientific and Medical) de 2,4 GHz y para evitar interferencias emplea FHSS (Frequency Hopping Spread Spectrum). Opera a una velocidad de un Mbps. Bluetooth cinco permite configurar velocidades de transmisión de datos de: 125 kbps, 500 kbps, 1 Mbps y 2 Mbps más adelante.
- **Basic Rate/Enhanced Data Rate mode (BR/EDR):** También opera en la banda ISM de 2,4 GHz, pero usa diferentes canales de radiofrecuencia que LE y AMP. BR soporta una tasa de bits de un Mbps. EDR proporciona una tasa de datos de entre dos y tres Mbps. Se usa FHSS para prevenir interferencias.
- **Alternative MAC/PHY (AMP):** Es una funcionalidad opcional que usa 802.11 para mayores velocidades de transporte hasta 24 Mbps. Este modo requiere que tanto el

maestro como el esclavo soporten AMP. También se requiere que el sistema tenga BR/EDR para establecer una conexión previa.

Para evaluar el consumo energético que tiene el Bluetooth, he tomado como referencia el componente ATSAMB11-MR510CA [18]. Se trata de un módulo LE que maneja la versión 4.1 de Bluetooth, se alimenta con 3,6 V y tiene un consumo de corriente de 4,5 mA que se da cuando el dispositivo está en modo de recepción. Nos da lugar a un consumo de 16,2 mW.

2.2.2 Zigbee

Al igual que el Bluetooth, Zigbee opera principalmente en la banda ISM a 2,4 GHz. Pero a diferencia del Bluetooth, también puede utilizar la frecuencia de 868 MHz en Europa y 915 MHz en EEUU y Australia. Tiene un rango de alcance de entre 10 y 75 metros [19].

Zigbee está basado en el estándar 802.15.4 [20] por lo que dependiendo de la modulación que se utilice contara con diferentes velocidades de transmisión. Para Europa: BPSK alcanza 20 kbps, O-QPSK (Override Quadrature Phase Shift Keying) alcanza 100 kbps y ASK (Amplitude-shift Keying) alcanza 250 kbps. Soporta tres topologías básicas: estrella, árbol y malla. Por cada dispositivo Zigbee puede haber como máximo 240 nodos.

Para evaluar el consumo energético que tiene Zigbee, he tomado como referencia el componente XB3-24Z8CM-J [21]. Este módulo dispone de dos versiones, la normal y la pro. La versión normal se alimenta con 3,6 V y tiene un consumo máximo de corriente de 40 mA a la hora de transmitir. Esto da lugar a un consumo de 144 mW. Mientras que la versión pro, tiene un consumo de corriente de 135 mA, ya que tiene más potencia de transmisión, por lo que su consumo asciende a 486 mW. Debido a la diferencia de consumo elegiremos la versión normal para hacer la comparativa con los demás protocolos.

2.2.3 Z-Wave

Z-Wave es un protocolo [22] destinado principalmente para la domótica y enfocado al consumidor. Está basado en estándar ITU-T G.9959 [23], que contienen las especificaciones de los dispositivos transmisores de corto alcance en banda estrecha, que incluye las frecuencias sub-1 GHz, menores de 1 GHz. Tiene un alcance de 100 metros [24].

Existen varias bandas, dentro del rango de frecuencias sub-1 GHz, usadas para Z-Wave dependiendo del país. Para Europa existen dos frecuencias centrales: 868,4 MHz y 869,85 MHz. Hay tres velocidades de transmisión que varían con la frecuencia:

- **100 kbps:** 868,4 MHz usa la modulación GFSK (Gaussian Frequency Shift Keying).
- **40 kbps:** 869,85 MHz usa la modulación FSK (Frequency Shift Keying).
- **9,6 kbps:** 869,85 MHz usa la modulación FSK.

La topología que se usa es la red en malla. Cada red puede tener hasta un máximo de 232 nodos incluyendo al maestro.

Para evaluar el consumo energético que tiene Z-Wave, he tomado como referencia el componente ZM5101 [25]. Este módulo se alimenta con 3,3 V típicamente, aunque puede soportar hasta 3,6 V máximo. Cuando el dispositivo se encuentra en transmisión con una potencia de +6dBm, consume 45,4 mA de corriente. Todo esto implica que el dispositivo podría tener un consumo de 149,82 mW.

2.2.4 LiFi

LiFi (Light Fidelity), sistemas de comunicaciones inalámbricas rápido y de bajo coste. Usa la luz led para transmitir información y puede llegar a los 10 Gbps y tiene un alcance de entre cinco y diez metros [26].

El estándar IEEE 802.15.7 [27] define la capa física. Usa la luz visible como medio de transmisión. Puesto que la luz es el medio que se usa, entre emisor y receptor debemos asegurar que no haya nada que obstruya el paso de esta.

Bajo estas condiciones podemos formar redes: punto a punto, de estrella y de difusión (broadcast). En las redes de difusión un dispositivo puede transmitir señales a otros dispositivos sin formar una red con ellos, es una comunicación unidireccional y para la que no se requiere la dirección de destino.

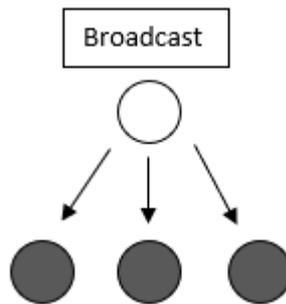


Figura 3-Topología LiFi

2.3 IP-Based WPAN y WLAN

Soportar la capa IP en los protocolos consume muchos recursos que podrían ser usados en otro lugar. Sin embargo, existen beneficios claves en implantar IoT en sistemas que permiten comunicarse a los dispositivos mediante TCP/IP (Transmission Control Protocol/Internet Protocol).

2.3.1 6LoWPAN

El nombre de 6LoWPAN es un acrónimo que proviene de IPv6 over low power WPANs. Este protocolo fue diseñado sobre las especificaciones del RFC 4944 [28] y más adelante se actualizó con el RFC 6282 [29]. La tasa de transmisión de datos es de 250 kbps. El protocolo cuenta con libertad de bandas de frecuencia, usando las bandas de 2,4 GHz y sub-1GHz, compatible con otros como: 802.15.4 [20], Bluetooth, sub-1 GHz protocolos de radiofrecuencia y PLC. Un efecto secundario de usar IPv6 es que proporciona una cantidad teórica de direcciones de 2^{128} . De esta manera se podrá cubrir en torno a 50 mil millones de dispositivos conectados a internet alrededor del año 2020.

Las redes 6LoWPAN son redes en malla que se encuentran en la periferia de redes más grandes. Dentro de esta red en malla se pueden encontrar topologías flexibles, permitiendo punto a punto y redes disjuntas, sin ningún tipo de conexión a Internet, o también pueden estar conectadas al Backbone (conexión principal que une routers secundarios con servidores privados) o a Internet usando routers secundarios/superficiales [30]. Es decir, mediante la

disposición de estos routers secundarios se puede realizar la interconexión entre la topología principal de malla y otra de las topologías flexibles.

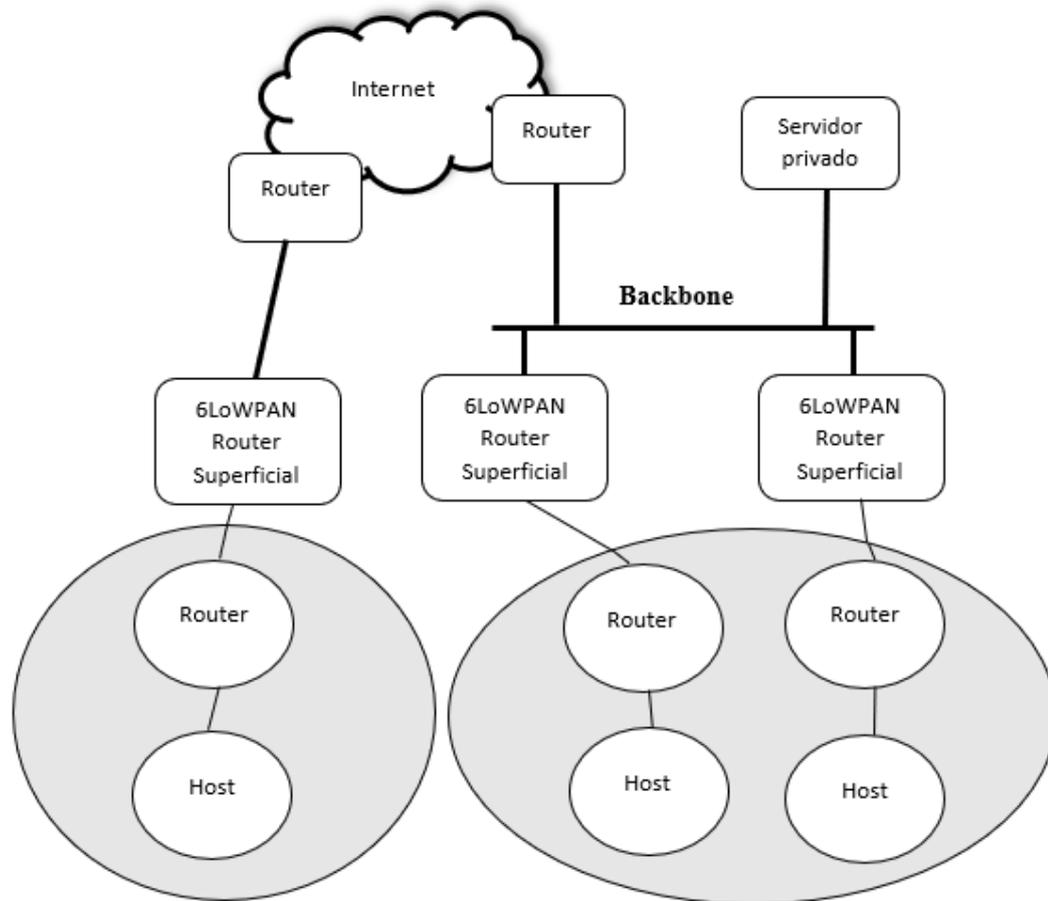


Figura 4-Red 6LoWPAN [31]

Para evaluar el consumo energético que tiene 6LoWPAN, he tomado como referencia el componente ZWIR4512 [31]. Este módulo se alimenta con 2,0 V (sin uso de ADC (Analog-To-Digital converter)) o 2,4 V (usando ADC), el ADC viene implementado en el microcontrolador para transformar la señal analógica en digital, y como máximo soporta 3,6 V. El dispositivo transmite una potencia máxima de +5 dBm en Europa y de +10 dBm para Estados Unidos. Con un consumo de corriente de 175 mA, que nos da como resultado que el dispositivo podría tener un consumo de 630 mW.

2.3.2 Thread

Thread [32] es un protocolo de comunicaciones para IoT relativamente nuevo. Está basado en IPV6 (6LoWPAN) y el protocolo 802.15.4 [20]. Thread también está basado en redes de malla, haciéndolo atractivo para sistemas de iluminación, con un máximo de 250 dispositivos dentro de la misma red. Todas las comunicaciones usan la encriptación AES, operando a una velocidad de 250 kbps en la banda de 2,4 GHz.

Para evaluar el consumo energético que tiene Thread, he tomado como referencia el componente EFR32MG1 [33]. Se alimenta con un voltaje típico de 3,3 V, aunque soporta

unos valores dentro del rango de 1,85 V a 3,8 V. La corriente que consume es de 200 mA lo que da lugar a un consumo de energía de 760 mW.

2.3.3 Protocolos IEEE 802.11 y WLAN

Una de las primeras tecnologías que adoptó por primera vez las bandas libres de licencia ISM, fue la tecnología IEEE 802.11 [34]. La tecnología IEEE 802.11 es una gama de protocolos entre los que se encuentra la definición de WLAN. El éxito de IEEE 802.11 se le puede atribuir a la aproximación al modelo de capas OSI [35].

Dentro de la gama amplia de protocolos de IEEE 802.11, existen diferencias y semejanzas entre ellos. En esta tabla adjunta podemos ver las principales características de cada uno:

IEEE 802.11 Protocol	Frequency (GHz)	Bandwidth (MHz)	Data Rate (Mbps)	Indoor Range (m)	Outdoor Range (m)
802.11	2.4	22	1 to 2	20	20
a	5	20	6 to 54	30	120
	3.7				5000
b	2.4	22	1 to 11	50	150
g	2.4	20	6 to 54	38	140
n	2.4/5	20	7.2 to 72.2	70	250
		40	15 to 150		
ac	5	20	7.2 to 96.3	35	35
		40	15 to 200		
		80	32.5 to 433.3		
		160	65 to 866.7		
ah	2.4/5	1 to 16	347	1000	1000
p	5.9	10	27	NA	400 to 1000
af	0.47 to 0.71	6 to 8	568	NA	6000-100000
ad	60	2160	4260	10	10
ax	2.4/5	20	450 to 10000	35	35
		40			
		80			
		160			

Tabla 2-Resumen protocolos 802.11 [36]

Los sistemas 802.11 soportan tres topologías básicas:

- **Infrastructure:** En esta forma, la estación hace referencia a un dispositivo 802.11 que se comunica con la estación a través de un AP (Access Point) central. Este método es también conocido como BSS (Infrastructure Basic Set Service). Esta topología es una red en estrella.
- **Ad hoc:** Los nodos 802.11 forman una red independiente entre ellos, llamada IBSS (Independent Basic Set Service). Esta es una topología punto a punto.
- **Distribution system:** El sistema distribuido combina dos o más IBSS a través de puntos de interconexión entre ellos.

Para evaluar el consumo energético que tienen los dispositivos que usan 802.11, he tomado como referencia el componente ESP8266EX [37]. Se alimenta con un voltaje típico de 3,3 V, aunque soporta un rango de 2,5 V a 3,6 V. El modo que más corriente consume es el de transmisión 802.11b que emite una potencia de +17 dBm y consume 170 mA. Esto supone un consumo máximo de 612 mW.

2.4 Sistemas Long-Range Communication y Protocolos (WAN)

Ya hemos visto el funcionamiento de WPAN y WLAN. Este tipo de comunicaciones cubren un amplio espectro de los dispositivos IoT. Aunque si queremos llegar a los lugares más remotos, entonces tenemos que utilizar WAN (Wide Area Network).

Dentro de las tecnologías WAN veremos la red celular, LoRa y Sigfox.

2.4.1 Cellular connectivity

La forma más predominante de comunicación es la radio celular y específicamente los datos móviles. Los dispositivos de comunicación móvil han existido desde hace muchos años antes de la tecnología celular, tenían cobertura limitada, compartían el mismo espacio de frecuencia y básicamente eran radios de doble sentido.

La teoría celular, se basa en patrones hexagonales que garantizan la separación de frecuencias entre las celdas vecinas. Este método permite reusar las frecuencias.

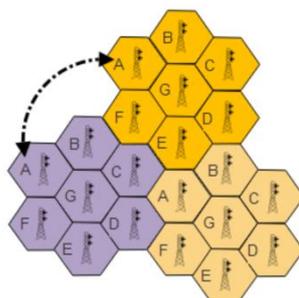


Figura 5-Teoría celular [68]

Dentro de los datos móviles tenemos tecnologías que han ido evolucionando con el tiempo: 1G, 2/2.5G, 3G, 4G y 5G. El 5G está previsto para el 2020 así que aún no se saben datos seguros, ya que están por determinar. Sin embargo, en España a día de hoy se dispone de un despliegue inicial de 5G por parte del operador Vodafone gracias a la red de Huawei [38].

En la siguiente tabla podemos observar la información que caracteriza a cada una de estas tecnologías cotejadas con el ITU:

Feature	1G	2/2.5G	3G	4G	5G
First Availability	1979	1999	2002	2010	2020
Specification	NA	NA	IMT-2000	IMT-Advanced	IMT-2020
Frequency Specification	NA	NA	400 MHz to 3 GHz	450 MHz to 3.6 GHz	TBD
Bandwidth Specification	NA	NA	Stationary: 2 Mbps Moving: 384 Kbps	Stationary: 1 Gbps Moving: 100 Mbps	Min Down: 20 Gbps Min Up: 10 Gbps
Typical Bandwidth	2 Kbps	14.4-64 Kbps	500 to 700 Kbps	100 to 300 Mbps	TBD
Standards and Multiplexing	AMPS	2G: TDMA, CDMA, GSM 2.5G: GPRS, EDGE, 1xRTT	FDMA, TDMA, WCDMA, CDMA-2000	CDMA	CDMA
Core Network	PSTN	PSTN	Packet Switch	Internet	Internet
Switching	Circuit	Circuit for Access network and air network	Packet-based except for air interface	Packet-based	Packet-based
Technology	Analog Cellular	Digital Cellular	Broad bandwidth CDMA, WiMAX, IP-based	LTE Advanced Pro-based	LTE Advanced Pro-based, mmWave

Tabla 3-Resumen tecnologías móviles [39]

Para evaluar el consumo energético que tienen las redes celulares, he tomado como referencia el componente SIM800H [40]. Se alimenta con un voltaje típico de 4,0 V, aunque el dispositivo soporta un rango de 3,4 V a 4,4 V. El pico máximo de corriente que puede alcanzar es de 2 A, pero el valor típico que se puede dar es de 453,57 mA. Luego el consumo de potencia que tiene el dispositivo es de 1,8 W.

2.4.2 WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) es la alternativa al LTE (Long Term Evolution). Ambos son estándares para la conectividad móvil de banda ancha, usan el protocolo de comunicación OFDMA (Orthogonal Frequency-Division Multiple Access).

WiMAX está basado en el estándar IEEE 802.16 [41] y manejado por WiMAX Forum. Utiliza la banda de 2,3 a 3,5 GHz, pero puede alcanzar el rango de 2,1 GHz y 2,5 GHz, como LTE. Puede tener una cobertura de hasta 70 km y transmite a tasas de velocidad: móviles de 100 Mbps y fijas de 1 Gbps.

Las topologías compatibles con WiMAX son: punto a punto y red en malla.

Para evaluar el consumo energético que tiene WiMAX, he tomado como referencia el componente MGA-22103 [42]. Se alimenta con un voltaje típico de 3,3 V, aunque el dispositivo soporta un rango de 3 V a 5 V. En el modo de máxima potencia transmite una potencia de 25 dBm y consume 437 mA, mientras que en el modo de baja energía utiliza una corriente de 85 mA transmitiendo 0 dBm. Luego, esto supone un consumo de 280,5 mW.

2.4.3 LoRa y LoRaWAN

LPWAN (Low Power Wide Area Network) incluye tecnologías privadas y no apoyadas por 3GPP (3rd Generation Partnership Project). Nosotros nos centraremos en dos de estas tecnologías LPWAN: LoRa (Long Range Radio) y Sigfox.

El protocolo LoRa tiene un rango de alcance de 2 a 5 km en áreas urbanas y de 15 km en áreas suburbanas. La arquitectura está basada en las siguientes bandas ISM del espacio de licencia gratuita: 915 MHz en EEUU, 868 MHz en Europa y 433 MHz en Asia. Tiene una velocidad de transferencia de datos de 0,3 a 5 kbps, y un tamaño de 0 a 250 bytes por paquete.

La topología en la que se basa este protocolo es la red de estrella.

Para evaluar el consumo energético que tiene LoRa, he tomado como referencia el componente CMWX1ZZABZ-078 [43]. Se alimenta con un voltaje típico de 3,3 V y puede transmitir a diferentes potencias: +20 dBm, +14 dBm y +10 dBm. El voltaje debería estar entre 2,4 V y 3,6 V. Cuando se transmiten +20 dBm el dispositivo requiere de 128 mA de corriente, para +14 dBm utiliza 47 mA y para +10 dBm utiliza 36 mA. Luego el consumo medio será de 155,1 mW.

2.4.4 Sigfox

Sigfox es un protocolo LPWAN que funciona en banda estrecha, desarrollado en 2009 en Toulouse, Francia. Esta es otra tecnología LPWAN que usa las bandas sin licencia ISM para un protocolo privado, requiriendo una suscripción de pago para su uso. Sigfox tiene una serie de restricciones que disminuyen su utilidad:

- Límite de hasta 140 mensajes de subida y 4 de bajada, diarios por dispositivo.
- Un tamaño de paquete de 12 bytes por mensaje de subida y ocho bytes de bajada.
- Una velocidad de 100 bps en subida y 600 bps de bajada.

Estas son las características que tiene la suscripción de 16,13 € anuales por dispositivo. Existe la posibilidad de reducir estas restricciones mediante un plan de suscripciones disponibles para compañías que necesiten por encima de 1000 dispositivos.

Originalmente, Sigfox era unidireccional. Pretendía ser una red compuesta por únicamente sensores subiendo información a la red, lo que implicaba soporte para subida y no bajada. Con el tiempo han decidido habilitar el canal de bajada.

Sigfox es una tecnología patentada y cerrada. Mientras que su hardware es abierto, la red no lo es. Esto es debido a que la empresa Sigfox, junto a Cellnex Telecom, ha instalado diversas

estaciones base por todo el mundo para dar cobertura a todos sus dispositivos. Cada una de estas estaciones base puede estar formada por hasta un millón de nodos, creando así una red en estrella y siguiendo la topología mostrada en la siguiente imagen:

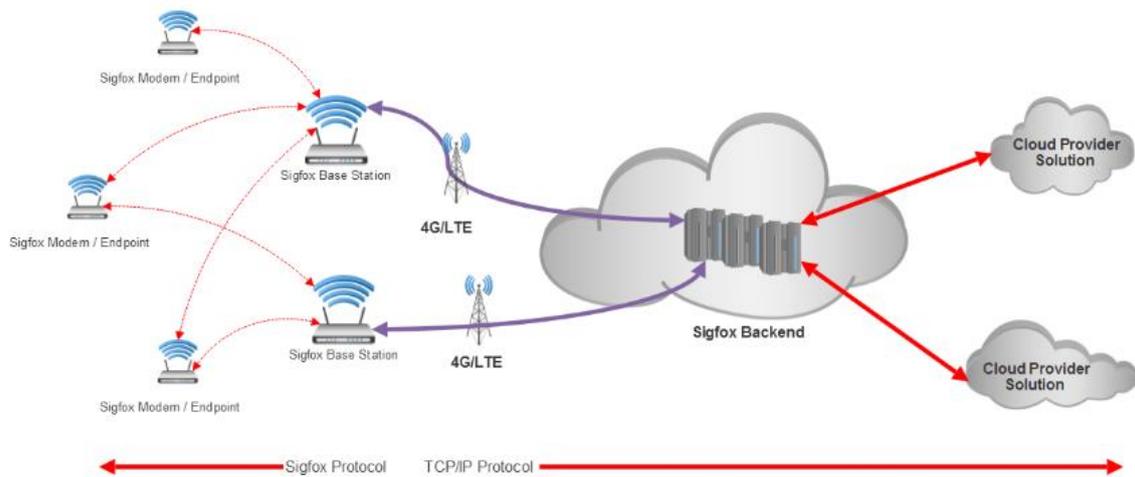


Figura 6- Topología de Sigfox [44]

Por lo tanto, debemos suscribir nuestros dispositivos a la red Sigfox para poder tener acceso a esas estaciones base.

Como hemos visto previamente, Sigfox usa UNB (Ultra Narrow Band) en las frecuencias: 868 MHz para Europa y 902 MHz para Norte América. Para la banda de subida (100 Hz de ancho) se usa OSSS (Orthogonal Sequence Spread Spectrum) y para la banda de bajada (600 Hz de ancho) se usa GFSK. Sigfox enviará un pequeño paquete a un canal aleatorio con un tiempo de retraso aleatorio (500 a 525 ms), para codificar los envíos. Este tipo de codificación es llamada RFTDMA (Random Frequency and Time Division Multiple Access) [44].

Para evaluar el consumo energético que tiene Sigfox, he tomado como referencia el componente SiPy [45]. Este componente se alimenta con un voltaje entre 3,5 V y 5,5 V, se alimentará con 5 V. Y necesita una corriente de 12 mA. Luego la potencia que necesitará el dispositivo es de 60 mW.

2.5 Resumen protocolos

A continuación, se hará una recopilación de la información más relevante de cada uno de los protocolos citados anteriormente:

Protocolo	Alcance	Velocidad	Consumo	Topología	N° dispositivos
Bluetooth 4.1	0,5 m	24 Mbps	16,2 mW	Piconet	8
				Scatternet	NA
Zigbee	10-75 m	250 kbps	144 mW	Estrella, árbol y malla	240
Z-Wave	100 m	100 kbps	149,82 mW	Malla	232
LiFi	5-10 m	10 Gbps	NA	Broadcast	NA
6LoWPAN	NA	250 kbps	630 mW	Malla	>50 mil millones
Thread	NA	250 kbps	760 mW	Malla	250
802.11b	50 m (interior) 150 m (exterior)	1-11 Mbps	612 mW	NA	NA
LTE/4G	NA	100-300 Mbps	1,8 W	Celular	NA
WiMAX	70 km	100 Mbps (móvil) 1 Gbps (estacionario)	280,5 mW	Punto a punto y malla	NA
LoRaWAN	2-5 km	0,3-5 kbps	155,1 mW	Estrella	NA
Sigfox	10 km (urbano) 40 km (rural)	100 bps (subida) 600 bps (bajada)	60 mW	Estrella	>1000

Tabla 4-Resumen características protocolos

2.6 Análisis protocolos

En este apartado se observarán las principales características vistas previamente de cada uno de los protocolos bajo análisis y se determinarán las mejores en cada uno de los aspectos. Buscaremos el equilibrio entre alcance de la comunicación, consumo, velocidad y número de dispositivos.

En primer lugar, Bluetooth es un protocolo muy útil para usarlo en dispositivos que se encuentren bastante cerca el uno del otro, ya que como se puede observar, su alcance es de 0,5 m aproximadamente. Sin embargo, tiene una buena velocidad de transferencia, de hasta 24 Mbps, y su consumo energético es el más bajo de todos los protocolos analizados en este TFG. En este caso descartaremos el uso de este protocolo debido al escaso rango de alcance.

En cuanto a Zigbee, se puede observar que el alcance es bastante mayor al del Bluetooth, pero no es de los mayores rangos que podemos encontrar en comparación con los otros protocolos. Este aumento de rango se ve repercutido en la velocidad de transmisión que se queda en 250 kbps y el consumo aumenta sin llegar a ser muy elevado. De nuevo, descartaremos este protocolo debido a que buscaremos un alcance mayor a los 10-75 m de este protocolo, para que pueda cubrir un área más extensa.

En Z-Wave se encuentra un equilibrio aceptable ya que el rango de alcance de 100 m es intermedio, la velocidad de transferencia no es muy alta, pero gracias a eso el consumo es comedido, y presenta una cantidad de dispositivos manejables de 232. Este protocolo podría ser una buena opción, pero por la misma razón que en Zigbee buscaremos algo más de alcance que 100 m.

Uno de los protocolos más fácil de descartar para este TFG (Trabajo de Fin de Grado) ha sido LiFi, debido a que las condiciones en las que funciona, es complicado realizar un enlace de luz visible más que en un entorno cerrado y sin interferencias.

Los protocolos que funcionan bajo IP como: 6LoWPAN, Thread y 802.11. Se descartarán debido a su elevado consumo energético, comparado con el resto de los protocolos vistos, y también a los inconvenientes que tiene trabajar sobre IP comentados previamente, entre otros el tráfico de datos indeseado.

La conectividad móvil es una opción muy buena ya que debido a la cantidad de usuarios que hacen uso de ella está muy avanzada en cuanto a cobertura y velocidad de transmisión. Sin embargo, está pensada para funcionar en dispositivos de telefonía móvil que cuenten con una gran capacidad de batería como podemos deducir de su elevado consumo energético. Por lo tanto, descartaremos esta opción para nuestro TFG.

Para las conexiones que necesiten mucho alcance, como por ejemplo zonas rurales, WiMAX es el protocolo que cuenta con el mayor alcance y una elevada tasa de transmisión de datos. Estas prestaciones suponen un consumo elevado de energía y debido a eso no será el protocolo adecuado para nosotros.

Sin lugar a duda, la opción que buscamos se encuentra entre LoRaWAN y Sigfox. Ambos protocolos tienen un amplio alcance. Los aspectos que determinarán la elección serán la velocidad de transmisión y el consumo. Por un lado, LoRaWAN cuenta con una velocidad de transmisión mayor a la de Sigfox, pero ese factor implica casi el doble de consumo energético.

Luego en una visión general de todos estos protocolos, el que más se asemeja a nuestras peticiones es Sigfox y es el que se comenzará a desarrollar en los siguientes apartados de esta memoria.

3 Diseño

Al comenzar el planteamiento del diseño, la primera opción fue el uso del protocolo Sigfox para comunicar el servidor web con los dispositivos SiPy. Dado que la finalidad del protocolo Sigfox es la subida de datos a la web, el esquema que en un principio se pretendió seguir es el siguiente.

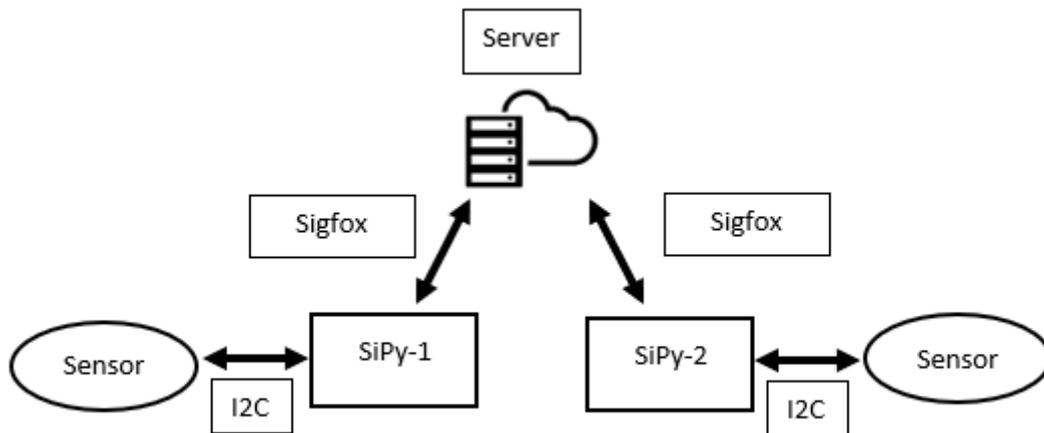


Figura 7-Esquema del diseño inicial

Sin embargo, en el apartado de desarrollo veremos que Sigfox no es el método adecuado para realizar nuestra prueba de concepto y por ese motivo se ha elegido utilizar la conexión del dispositivo SiPy-1 al servidor web a través de UART (Universal Asynchronous Receiver-Transmitter). Por lo tanto, se le dio un enfoque de un sistema de radio al dispositivo SiPy-1 a través del cual recibirá la comunicación del servidor y se la traspasará al SiPy-2, que estará dotado de ambos sensores. La comunicación entre ambos dispositivos se realizará mediante la modulación FSK (Frequency Shift Keying).

El esquema final de diseño que seguiremos es el que se muestra en la siguiente imagen.

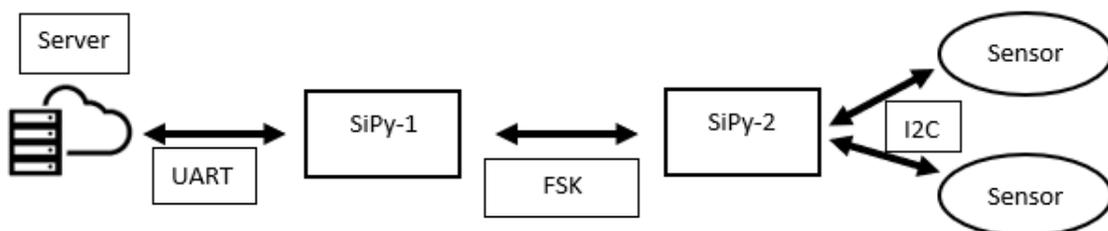


Figura 8-Esquema del diseño final

A continuación, se expondrá una breve explicación de todos los elementos involucrados en el apartado de diseño, los cuales se desarrollarán en detalle en la parte del desarrollo.

3.1 Comunicación

Como hemos visto, la elección del protocolo ha sido Sigfox luego los dispositivos que dispondremos para realizar la comunicación deben funcionar con este protocolo.

3.1.1 SiPy

SiPy es el dispositivo, con soporte para la tecnología Sigfox, que se va a usar. El principal motivo de la elección ha sido el soporte de Sigfox y otros métodos de comunicación como WiFi que serán útiles a la hora de su programación. Por otra parte, el fabricante facilita una guía al usuario muy completa que será de gran ayuda para la familiarización con el dispositivo.

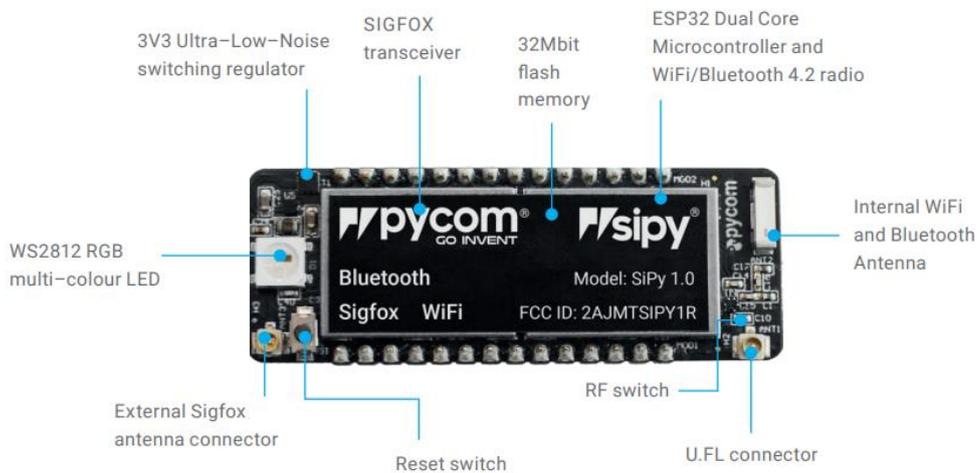


Figura 9-Dispositivo SiPy [46]

Como se puede observar en la Figura 9, existe un conector externo de antena para Sigfox. Es necesario conectar una antena si se quiere hacer uso del protocolo Sigfox, de lo contrario se podría dañar el dispositivo, según las advertencias del fabricante. Por lo tanto, es necesaria una antena externa para la utilización de Sigfox. Tanto SiPy como la antena comparten el mismo fabricante, Pycom.



Figura 10-Antena Sigfox

3.1.1 Pantalla OLED 0,91” y HTU21D

La pantalla OLED de 0,91” es el primer periférico que conectaremos a la SiPy-2. El HTU21 es el sensor digital de temperatura y humedad que también irá conectado a la SiPy-2.

Esta pantalla tiene una resolución de 128x32 píxeles. Se utilizará para imprimir los mensajes que mande el usuario desde la página web. La conexión entre SiPy-2 y el periférico se hará mediante el bus I^2C .

A través de la página web, el usuario podrá tener acceso a la temperatura y la humedad del lugar donde se encuentre el dispositivo SiPy-2. Este sensor también irá conectado a la SiPy-2 mediante el bus I^2C .

3.1.2 FSK Modulation

El dispositivo SiPy ofrece la posibilidad de intercambiar información con otro dispositivo SiPy utilizando la modulación FSK. La comunicación mediante FSK utiliza un enlace directo entre las antenas Sigfox de cada dispositivo.

La modulación por desplazamiento de frecuencia consiste en una señal binaria de entrada y en un sub-sistema que se encarga de elegir entre dos señales las cuales comparten amplitud, pero utilizan diferentes frecuencias. Si la señal digital de entrada representa un “1” lógico, el sub-sistema selector asignará la señal de frecuencia mayor en ese tramo a la señal modulada. Si la señal digital de entrada representa un “0” lógico, la señal de menor frecuencia será asignada a ese tramo. Dando como resultado una señal modulada en frecuencia [49].

3.1.3 UART (Universal Asynchronous Receiver-Transmitter)

Para poder introducir el término UART, primero debemos hacer lo propio con el término de puerto serie. En el puerto serie se tienen dos líneas, transmisora (Tx) y receptora (Rx). Estas líneas se deben cruzar para comunicar dos dispositivos, es decir, el Tx del dispositivo 1 debe conectarse al Rx del dispositivo 2 y viceversa. Además, ambos deben compartir una masa común [50].

El hardware que se usa para enviar datos a través de las líneas del puerto serie es el llamado UART.

El dispositivo SiPy conectado al ordenador, utilizara la transmisión UART para el intercambio de información con el servidor web que se encuentra alojado en el ordenador.

3.1.4 I^2C (Inter-Integrated Circuit)

El protocolo I^2C es un protocolo síncrono que usa cuatro líneas para funcionar, una para el reloj (SCL), otra para los datos (SDA) y por último se necesitan líneas de alimentación (VCC) y una línea de referencia de masa (GND).

Los dispositivos conectados al bus de comunicaciones en serie, I^2C , deben tener una dirección única para cada uno. A su vez, pueden ser maestros o esclavos. El dispositivo maestro transmite los datos y genera la señal de reloj. La capacidad de multimaestro radica principalmente en que los dispositivos son capaces de detectar si el bus I^2C está siendo utilizado para no interrumpir esa comunicación. El control del bus se realiza de manera

arbitraria, es decir que si dos elementos comienzan una comunicación a la vez uno de ellos será elegido mientras que el otro tendrá que cancelar cualquier escritura sobre el bus I^2C . Por estas razones al bus I^2C se le denomina bus multimaestro [51] [52].

En nuestro diseño, se usará el bus I^2C para comunicar el dispositivo SiPy-2 con el sensor HTU21D y con la pantalla OLED 0,91”.

3.2 Servidor Web

La página web se encargará de recoger la información que requiera el usuario y las acciones que desee realizar y enviarlas a los dispositivos SiPy. Estará montada en el sistema operativo Ubuntu, sobre una distribución de Apache llamada LAMPP. El nombre es en realidad un acrónimo: L (Linux), Apache, MariaDB/MySQL, PHP, Perl.

3.2.1 HTTP (Hypertext Transfer Protocol) Apache

El servidor que se ha elegido para alojar la página web del proyecto es Apache. Apache es un servidor web HTTP de código abierto, válido para la mayoría de las plataformas incluida la nuestra, Linux. Estas características, junto a la popularidad y sencillez de funcionamiento, han sido cruciales para la elección de esta opción.

3.2.2 PHP (PHP: Hypertext Preprocessor)

PHP es el código que se utiliza para el desarrollo web de contenido dinámico. Además de ser el más popular y más utilizado mundialmente. Este lenguaje tiene una gran documentación oficial dispuesta al usuario con todas las funciones disponibles en este lenguaje y ejemplos sobre su funcionamiento. Toda esta información convierte a este lenguaje en uno de los más útiles y sencillos para su implementación.

3.3 Lenguajes de programación

Tanto como el dispositivo SiPy como la página web tienen sus lenguajes de programación. Para SiPy se utiliza el lenguaje micropython y para la página web se necesitarán PHP, HTML (Hyper Text Markup Language) y CSS (Cascading Style Sheets).

3.3.1 Micropython

Siguiendo la definición que se indica en la página oficial [53], micropython es un pequeño pero eficiente intérprete del lenguaje de programación Python 3, que incluye un pequeño subconjunto de librerías y que además está optimizado para ejecutarse en microcontroladores. Este es el lenguaje que usaremos para programar nuestros dispositivos SiPy.

3.3.2 HTML y CSS

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. No es lo mismo que PHP, ya que en HTML no contamos con funciones aritméticas o variables [54] [55].

La función de HTML es, principalmente, añadir etiquetas o marcas con información adicional acerca de un texto o su presentación.

CSS es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación. Es decir, modela la parte estética de la página web [56].

4 Desarrollo

En este capítulo explicaré el desarrollo que se ha seguido tras la elección del protocolo Sigfox, detallando los inconvenientes que se han ido encontrando y explicando las decisiones que se han ido tomando.

4.1 Envío de mensajes Sigfox

Al principio del desarrollo, se enfocó a la utilización del protocolo Sigfox para el envío de mensajes. El funcionamiento de dicho protocolo se basa en hacer uso del servidor web de Sigfox llamado backend. El servidor backend se encarga de recolectar todos los mensajes que se envían mediante el protocolo Sigfox ya que estos mensajes, llegan primero a una estación base y desde ahí se envían al backend a través de una conectividad IP. Desde la página web del backend, se pueden ver los dispositivos Sigfox que tenemos a nuestra disposición y editar sus configuraciones, pudiendo desde renovar la suscripción de cada dispositivo hasta ver los mensajes que se han enviado desde los mismos.

Lo primero que se debe hacer es actualizar y activar los dispositivos Sigfox adquiridos en la página web oficial, explicado en el Anexo A Manual de instalación. Una vez tenemos activados nuestros dispositivos, según el tipo de licencia que tengamos, tendremos una limitación de mensajes de subida y de bajada mencionada en el análisis del protocolo. De esta manera, cuando se realiza el envío de un mensaje desde un dispositivo, este envío queda registrado en la página web y podemos ver la información relacionada a este envío como: el contenido del mensaje, la calidad de señal con la que se ha enviado, la fecha y la hora a la que se ha realizado, etc.

En nuestro caso, queremos mostrar la información de estos mensajes en nuestra propia página web. Para ello, deberemos configurar un callback, el callback consiste en redireccionar la información de los envíos realizados por el dispositivo a una URL (Uniform Resource Locator) de acceso público que comunica con nuestro servidor web, donde se encuentra alojado nuestro código. Este código se encarga de recibir la información y almacenarla en una base de datos creada dentro del servidor web. La información desde el Callback se puede enviar mediante dos métodos: GET o POST. El método GET sirve principalmente para el envío de parámetros entre cliente y servidor incluyéndolos en la URL, esto implica que los parámetros serán visibles. El método POST también se utiliza para enviar parámetros de manera oculta para que no se puedan ver y en este caso los parámetros enviados por POST serán codificados en el formato JSON, que es un tipo de formato de texto. Eligiendo el método POST, se envían los datos: identificador del dispositivo que ha enviado el mensaje, la fecha y la hora en la que se ha enviado el mensaje, el código de estación que ha recibido el mensaje y el contenido del mensaje.

El código de estación está relacionado a la red privada de estaciones base y antenas instaladas por Cellnex Telecom junto a Sigfox, que sirven para aumentar el ecosistema IoT (Internet of Things) [57].

El contenido del mensaje se envía en hexadecimal, por lo tanto, en el lado del servidor se hará una conversión de hexadecimal a cadena de caracteres para poder insertar el contenido del mensaje correctamente en la base de datos. Una vez realizada la inserción de toda esta información en la base de datos, podremos tener acceso al dispositivo que envió el mensaje,

el código de la antena que recibió ese envío conociendo de esta manera el lugar aproximado desde el que se envió el mensaje, cuándo lo envió y qué envió.

Habiendo realizado con éxito la subida de mensajes desde los dispositivos hacia el servidor web, nos encontramos con el problema de usar la misma interacción en diferente sentido, desde el servidor web hacia el dispositivo. Realizar la comunicación hacia un dispositivo no es tan sencillo ya que no se puede realizar en cualquier momento. Cuando se envía un mensaje desde el dispositivo se abre una ventana de transmisión y 20 segundos después de la transmisión del primer mensaje se abre una ventana de recepción. Esta ventana de recepción se mantendrá abierta durante 25 segundos, permitiendo la recepción de mensajes cortos (4 bytes) desde una estación base [44].

En este momento del desarrollo se decidió usar el método UART (Universal Asynchronous Receiver-Transmitter) para la comunicación del dispositivo SiPy-1 con el servidor, en lugar de utilizar Sigfox, debido al funcionamiento que siguen los mensajes de bajada y a las limitaciones de la suscripción vistas en el análisis del protocolo Sigfox. Mediante la conexión por UART no tendremos esas limitaciones en cuanto a cantidad o tamaño de los mensajes y se asegura una comunicación bidireccional.

Por último, para comunicar ambos dispositivos entre sí se usará la modulación FSK (Frequency Shift Keying). Existen varias diferencias entre el protocolo Sigfox y la modulación FSK. La modulación FSK se podría asemejar al funcionamiento de una radio, ya que modula el mensaje a una frecuencia y los dispositivos que se encuentren escuchando a la misma frecuencia, recibirán el mensaje. El alcance de FSK es mucho menor que el de Sigfox, dicho alcance se ha medido en el apartado 5.1 Alcance de comunicación radiofrecuencia y ronda entre los 100 y 150 metros aproximadamente. En FSK no contamos con las limitaciones de Sigfox en cuanto al número de mensajes ni de subida ni de bajada ni al tamaño de estos mensajes.

A continuación, explicaré el código utilizado para implantar la comunicación mediante FSK.

4.1.1 Device2device.py

Este programa comprueba el funcionamiento del intercambio de mensajes entre los dispositivos SiPy-1 y SiPy-2, utilizando los módulos incluidos en ambos dispositivos. Se inicializa un puerto que utiliza FSK y se configura a la frecuencia de 868 MHz, para ambos dispositivos. Por el lado del emisor, se envía un mensaje a través de ese puerto, este mensaje se envía en una cadena de caracteres. En el lado del receptor, el puerto se mantiene a la escucha de manera permanente y cuando recibe el mensaje decodifica la cadena de caracteres y lo imprime por pantalla.

Otro modo de funcionamiento se produce al enviar un mensaje preguntando por la temperatura o la humedad. Si este mensaje se da, el emisor queda a la espera de la respuesta del receptor, que recogerá los datos del sensor y los enviará de vuelta siguiendo el mismo procedimiento de envío.

4.2 RGB Led

Como se puede observar en la Figura 9, el componente SiPy viene dotado de un led RGB. Utilizando un ejemplo del manual del fabricante, comprobaremos que funciona correctamente [58].

4.2.1 Rgb.py

Por defecto, el led de la placa parpadea en un tono azul al arrancar el dispositivo como indicativo de que todo funciona correctamente. Para hacer uso de este led debemos desactivar este parpadeo, con los módulos incluidos en el dispositivo. Una vez desactivado basta con facilitar un color en formato hexadecimal para que el led se ilumine en ese color.

4.3 Pantalla OLED 0,91”

Para poder controlar esta pantalla específicamente se ha hecho uso la librería “sipy_lcd.py” obtenida de un repositorio de GitHub [59]. Por lo tanto, cabe agradecer al autor de dicha librería, dmitryelj, la integración de este periférico al proyecto.

4.3.1 Screen.py

Para hacer uso de la pantalla lo primero que debemos hacer es inicializar los pines que corresponden al bus I^2C : “P9” (SDA) y “P10” (SCL) [60], utilizando los módulos incluidos en el dispositivo SiPy. Escaneamos el bus I^2C para ver si se detecta el dispositivo y cuando se haya detectado, inicializamos la pantalla. Una vez se reciba el mensaje que se desea imprimir, borraremos el buffer de la pantalla e imprimiremos la cadena de caracteres que se ha recibido.

4.4 HTU21D

Al igual que con la pantalla, para el control de este sensor la librería se ha extraído de una modificación que ha realizado un usuario del foro de Pycom modificando el código base proporcionado por el MIT (Massachusetts Institute of Technology) [61]. De nuevo se agradece el funcionamiento del sensor a este usuario, dnear1.

4.4.1 Sensor.py

En este fichero se añade la librería del sensor, en la que se configura el bus I^2C , de nuevo usando los mismos pines usados para la pantalla. Se inicializa el objeto del sensor y llamando a las funciones de la librería se obtiene la temperatura y la humedad y se imprimen por pantalla.

4.5 UART

En este apartado se verá el método de configuración para implementar UART. Como se puede observar en la Figura 25, los pines “P0” (Rx) y “P1” (Tx) son los necesarios para conectar el cable UART a USB. Para conectarlo de manera adecuada, la línea Tx del cable UART a USB debe ir conectada al pin “P0” (Rx) del dispositivo y la línea Rx del cable UART a USB debe ir conectada al pin “P1” (Tx) del dispositivo.

Durante el desarrollo de la conexión UART, encontré dificultades a la hora de implementarla en el sistema operativo de Windows. Tras intentar solventar este problema se llegó a la conclusión de que realizar la comunicación UART en el sistema operativo Linux es la mejor opción. En esta parte del desarrollo se migró toda la implementación a una máquina virtual de Linux.

4.5.1 Reception.py

Este fichero está configurado para que la SiPy-1 lea continuamente el puerto UART y recopile las peticiones que provienen del servidor. Cuando obtenga esa información manda la orden a la SiPy-2 y ésta ejecuta la orden. En el caso de pedir la temperatura o humedad,

la SiPy-2 enviará la información de vuelta a la SiPy-1 y ésta la transmitirá al servidor vía UART.

4.6 Servidor Web

El servidor web funciona dentro de una máquina virtual que usa Linux, esta máquina virtual está montada con Oracle Virtual Box. En este momento del desarrollo surgió un problema con la configuración de red de la máquina virtual, debido a que no se le asignaba la dirección IP pública del ordenador, sino que se le asignaba una dirección IP virtual. Tras configurar la dirección IP correctamente, el puerto de acceso de tráfico del ordenador bloqueaba el tráfico al puerto de la máquina virtual y no nos dejaba acceder al servidor web instalado en la máquina virtual. La solución a este problema fue redireccionar los puertos de la máquina virtual al puerto del ordenador, de esta manera las peticiones que llegaran al ordenador se replicarían en la máquina virtual permitiendo el correcto funcionamiento del tráfico.

En el Anexo A Manual de instalación apartado Máquina virtual, se explica detalladamente el proceso de instalación y configuración para solucionar el problema del acceso a la dirección IP pública del ordenador.

4.6.1 Index.php

Este será el archivo que contenga nuestra página principal y será visible al usuario. En ella se diseña un formulario para recoger las peticiones y enviarlas al archivo “serial.php”, el cual explicaremos en la siguiente sección.

La página web tiene este aspecto:

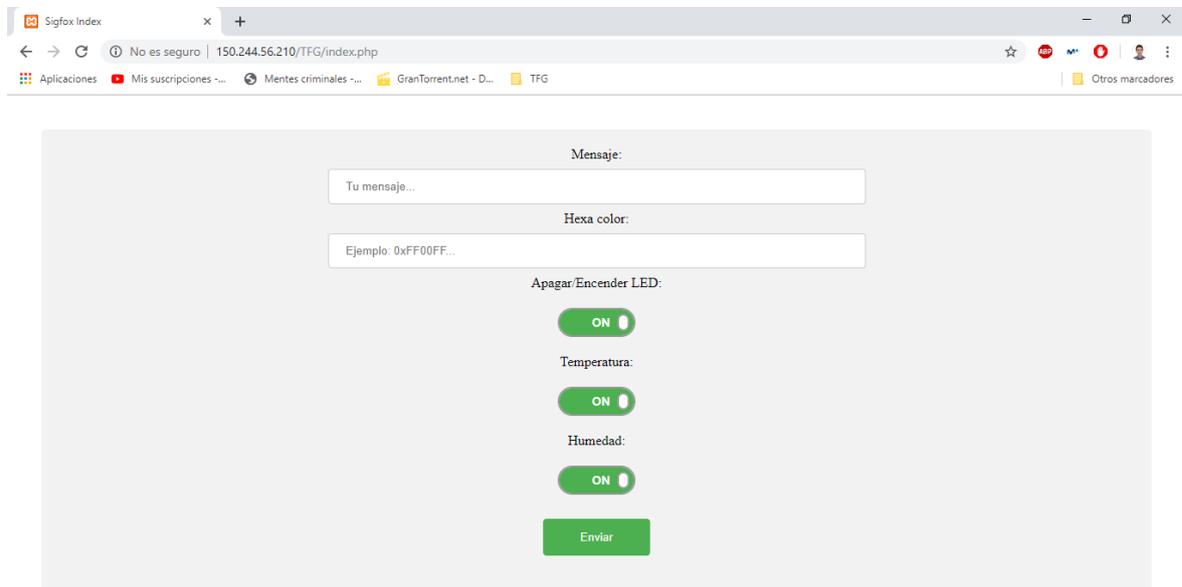


Figura 11-Página principal del servidor web

Como se puede observar en la Figura 13, el usuario puede: enviar un mensaje para imprimirlo por la pantalla oled, indicar un color en hexadecimal para encender el led RGB, decidir si apagar o encender dicho led y preguntar por la temperatura y humedad.

4.6.2 Serial.php

Una vez se envía el formulario que aparece en “index.php”, toda la información llega a este fichero y clasifica los datos y la forma de tratarlos. Se envía la información, mediante la conexión UART, a la SiPy-1. En caso de esperar respuesta, se mantiene a la espera hasta que recibe la información necesaria y la devuelve a la página principal para mostrarla al usuario. De lo contrario, vuelve de nuevo a la página principal para esperar nuevas peticiones.

4.7 SiPy-1

El código que se ejecutará al arrancar el dispositivo SiPy-1 consiste en inicializar la conexión mediante UART con el ordenador en el que se encuentra el servidor web y a la vez configurar el puerto con el que se comunicará con SiPy-2 mediante el uso de FSK. Una vez se haya recogido la información proporcionada por la página web, llegarán los mensajes a SiPy-1 mediante UART y dependiendo del mensaje realizará una acción u otra determinada. Si se requiere el valor de la temperatura, enviará un mensaje a SiPy-2 pidiendo el valor de temperatura y se pondrá a la espera de una respuesta, cuando esta respuesta llegue la trasladará a la página web a través de UART. En caso de recibir un color o un mensaje solo se enviará dicho color o mensaje a SiPy-2.

4.8 SiPy-2

En el lado del receptor irán conectados tanto la pantalla como el sensor de temperatura y humedad, luego debemos incluir las librerías para controlar ambos periféricos. Será necesario inicializar el puerto de comunicación con SiPy-1 mediante FSK a la misma frecuencia, 868 MHz. Como se hará uso del led RGB primero hay que deshabilitar el parpadeo por defecto. Se escaneará el bus I^2C para detectar ambos periféricos y una vez detectados se inicializarán como viene indicado en sus librerías. Este dispositivo se mantendrá a la escucha del puerto FSK y cuando reciba los mensajes de SiPy-1 actuará de una manera determinada dependiendo del mensaje. Si se recibe un mensaje preguntado por la temperatura se leerá dicho valor y se enviará una respuesta por el puerto FSK. En caso de que se reciba un color o un mensaje, ejecutará la acción directamente encendiendo el led o imprimiendo el mensaje.

5 Integración, pruebas y resultados

En este capítulo se comprobará la funcionalidad del sistema conjunto realizando diferentes pruebas como: medición del alcance de la comunicación por radiofrecuencia, medición de la tasa real de transferencia de la comunicación FSK (Frequency Shift Keying) y del bus I^2C (Inter-Integrated Circuit) y medición el tiempo que transcurrido entre los diferentes eventos de la comunicación.

5.1 Alcance comunicación radiofrecuencia

La medición de este aspecto de la comunicación se ha realizado mediante la obtención de coordenadas GPS (Global Positioning System), manteniendo como punto de referencia el laboratorio en el que se ha instalado el servidor junto a la SiPy-1. También se ha corroborado la medición mediante el instrumento de medida, metro. Cabe diferenciar dos rangos:

- Si no existen más obstáculos, a parte del propio edificio donde se encuentra instalado el dispositivo SiPy-1, la comunicación funciona correctamente hasta los 150 m aproximadamente.
- Cuando entre ambos dispositivos interfieren más obstáculos que el propio edificio donde se encuentra instalado el dispositivo SiPy-1, se pierde alcance dependiendo de la densidad de estos. Contando con los edificios principales de la EPS (Escuela Politécnica Superior), el rango se reduce a 100 m o menos dependiendo de los factores que se interpongan entre ambos puntos de la comunicación.

5.2 Tasa real de transferencia

Se han medido las tasas de transferencia de FSK y del bus I^2C ya que son los elementos principales de la comunicación. Para medir ambas tasas, se ha utilizado un osciloscopio con el que se verán los flancos de subida generados en unos pines determinados, los cuales se activarán cuando se efectúe el evento que se indique.

Para medir la tasa de FSK, se envía un mensaje de 3 bytes y se activa un flanco de reloj al envío del mensaje y otro al recibirlo. Se realiza un número de 15 veces para ver el tiempo medio entre el envío y la recepción. El resultado es de 4,2 ms de media, convirtiendo los bytes a bits, se obtiene una velocidad de transferencia de 5,7 kbps.

Para medir la velocidad de escritura del bus I^2C , se ha escrito un mensaje de 3 bytes en la dirección asignada por el bus I^2C al sensor. Se ha seguido el mismo proceso que se ha usado para medir la tasa de FSK, pero ahora los flancos de subida se activan al escribir el mensaje en el bus I^2C . El tiempo medio obtenido tras las pruebas ha sido de 717,417 μ s, convirtiendo los bytes a bits, se obtiene una tasa de transferencia de 33,453 kbps. Teniendo en cuenta que el bus I^2C se inicializa a una tasa de 100 kbps, podemos observar que la tasa real es más pequeña que la teórica, algo que resulta normal teniendo en cuenta las pérdidas que pueden suceder en diferentes puntos como, por ejemplo, en los cables.

5.3 Tiempos de acción

En este apartado se medirán los tiempos que transcurren entre los diferentes sucesos de la comunicación. En todos ellos se ha medido la diferencia entre los flancos de subida generados por un pin previamente configurado. Para poder concretar un valor fiable, cada prueba se ha repetido 15 veces antes de llegar al valor medio final.

5.3.1 Envío FSK SiPy-1 a SiPy-2

Para determinar el tiempo que tarda en llegar el mensaje del dispositivo SiPy-1 al SiPy-2 a través de FSK, se activa un flanco de subida tras enviar el mensaje y en cuanto se reciba en el SiPy-2 activamos otro flanco. La diferencia entre ambos flancos es el tiempo que ha tomado dicho envío. La media de esta prueba ha sido un tiempo de 4,2 ms.

5.3.2 Actuación sensor I^2C

Para medir el tiempo que tardan los elementos I^2C en responder, se activa un flanco de subida cuando llega la orden de leer datos del sensor I^2C y se vuelve a activar cuando se recoge la información requerida. Tras esta prueba se ha llegado a un tiempo medio de 44,026 ms.

5.3.3 Respuesta FSK SiPy-2 a SiPy-1

Una vez se obtiene la información solicitada por la SiPy-1, se activa un flanco de subida al enviar la respuesta desde la SiPy-2 y se vuelve a activar al recibirla en la SiPy-1. El tiempo medio de este proceso resulta en 1,457 ms.

5.3.4 Tiempo total SiPy-1 a SiPy-2

El tiempo total que transcurre desde que se envía la orden desde SiPy-1 hasta que regresa la información proporcionada por la SiPy-2, es de 156,998 ms de media.

5.3.5 Tiempo respuesta usuario

El tiempo de respuesta que aprecia el usuario al realizar cualquier consulta, es medido desde la página web del cliente. Obteniendo la marca de tiempo del sistema desde cuando se envía la petición y restándosela a la marca de tiempo obtenida al recibir toda la información. Dependiendo de la cantidad de información que se requiera, cada consulta tardará más o menos tiempo. Por ejemplo, la media de tiempo que transcurre al realizar todas las consultas posibles de 1,4 s, un tiempo elevado debido a que se debe esperar a la respuesta del sensor. Si enviamos solamente un mensaje el tiempo de ejecución de la página se reduce a 200 ms.

En este apartado se incluyen el tiempo de respuesta del servidor y otros aspectos que están fuera de nuestro control.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Partiendo desde el objetivo principal del trabajo, se ha realizado un análisis de las diferentes opciones que existen para realizar sistemas de comunicaciones para su uso en dispositivos IoT. A parte, se ha conseguido implantar un sistema de comunicaciones inalámbricas, aunque experimentando ciertas complicaciones en algunas fases del desarrollo que ha supuesto un cambio de planteamiento en la forma de funcionar del sistema.

Podemos concluir con este trabajo que, la opción de utilizar el protocolo Sigfox para realizar comunicaciones entre varios dispositivos, no es lo suficientemente efectiva. Debido a las limitaciones que posee y al enfoque que tiene, más orientado a la recolección de datos dirigidos a un servidor web en lugar de a otro dispositivo.

6.2 Trabajo futuro

En cuanto al trabajo futuro que se puede realizar se podría considerar el implementar diferentes modos de conectar los sensores a SiPy, aparte de usar el bus I^2C , como UART o SPI (Serial Peripheral Interface).

También cabe la posibilidad de seguir analizando nuevos protocolos como Neul. O seguir analizando más en detalle las posibilidades del recién llegado 5G, ya que la información que se tiene acerca del 5G aún no es suficiente como para implantarlo en dispositivos IoT. En torno al 2020, un análisis más exhaustivo acerca del 5G será más efectivo.

Finalmente, si se decide seguir utilizando Sigfox, habría que implementar otro modo de funcionamiento en el que sea útil la comunicación Sigfox a través del backend, que controla el tráfico de mensajes y configuración de los dispositivos conectados a una cuenta de usuario. Para ello, sería de gran utilidad contar con una suscripción sin tantas limitaciones en cuanto a cantidad de mensajes disponibles al día.

Referencias

- [1] R. Shuler, «Web Standford,» Promeroy IT Solutions, 2002. [En línea]. Available:<https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm>.
- [2] Statista, «Statista,» 2019. [En línea]. Available: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.
- [3] P. Lea, *Internet of Things for Architects*, Packt, 2018.
- [4] Universitat De Valencia, «Sistemas Industriales Distribuidos,» [En línea]. Available: https://www.uv.es/rosado/courses/sid/Capitulo2_rev0.pdf.
- [5] ElyArielMagnoRaul, «Topologia de malla,» 22 Marzo 2009. [En línea]. Available: <http://topologiademalla.blogspot.com/>.
- [6] J. Rábago, «EcuRed,» [En línea]. Available: https://www.ecured.cu/Red_en_malla.
- [7] EcuRed, «Redes punto a punto,» [En línea]. Available: https://www.ecured.cu/Redes_punto_a_punto.
- [8] M. Juncosa, «Aprende De Redes,» [En línea]. Available: <https://aprendederedes.com/redes/introduccion/tipos-redes-informaticas/>.
- [9] J. Estrella, «TOPOLOGIA EN ARBOL,» [En línea]. Available: <http://jorge-star.galeon.com/ARBOL.html>.
- [10] P. Lea, «Non-IP Based WPAN,» de *Internet of Things for Architects*, Packt, 2018.
- [11] P. Lea, «Bluetooth history,» de *Internet of Things for Architects*, Packt, 2018.
- [12] Mundo Altavoces , «Bluetooth clases y versiones, ¿ En qué se diferencian?,» [En línea]. Available:<https://mundoaltavoces.com/bluetooth-todas-las-clases-y-versi-ones-en-que-se-diferencian/>.
- [13] Cultura Sonora, «Tipos de bluetooth ¿cuáles son las diferencias?,» 6 Noviembre 2017. [En línea]. Available: <https://www.culturasonora.es/blog/tipos-de-bluetooth-y-diferencias/>.
- [14] BBC Mundo, «5 ventajas de Bluetooth 5, la última versión de la popular tecnología inalámbrica,» *BBC*, 13 Junio 2016.
- [15] J. Monsalve, «Solución WiSeMCU Bluetooth 5 de modo dual,» *Comunicaciones inalámbricas hoy*, 25 Septiembre 2018.
- [16] «RF Wireless World,» 2012. [En línea]. Available:<http://www.rfwireless-world.com/Terminology/difference-between-piconet-and-scatternet-in-bluetooth.html>.
- [17] P. Lea, «Bluetooth 5 stack,» de *Internet of Things for Architects*, Packt, 2018.
- [18] Atmel. [En línea]. Available:https://www.mouser.es/datasheet/2/268/Atmel-42498-SAMB11-MR210CA-MR510_Datasheet-1143047.pdf.
- [19] D. d. S. R. L. E. Ñ. S. Carlos Alberto Ortega Huembes, «Monografias,» 28 Julio 2008. [En línea]. Available:<https://www.monografias.com/trabajos61/zigbee-estandar-domotico-inmotica/zigbee-estandar-domotico-inmotica2.shtml>.
- [20] IEEE, «802.15.4,» 5 Septiembre 2011. [En línea].

- Available: <https://ieeexplore.ieee.org/document/6012487>.
- [21] Digi. [En línea]. Available: https://www.mouser.es/datasheet/2/111/ds_xbee-3-zigbee-3-1288823.pdf.
- [22] IEEE, «Z-Wave Protocol,» 27 Abril 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8463281>.
- [23] U. I. d. Telecomunicaciones, 13 Enero 2015. [En línea]. Available: <https://www.itu.int/rec/T-REC-G.9959-201501-I/es>.
- [24] Z-Wave, «Z-Wave,» [En línea]. Available: <https://www.z-wave.com/faq>.
- [25] S. Labs. [En línea]. Available: <https://www.silabs.com/documents/login/data-sheets/DSH12625-9.pdf>.
- [26] A. M. d. Frutos, «¿Qué es LiFi y cómo funciona?,» 19 Diciembre 2015. [En línea]. Available: <https://computerhoy.com/noticias/internet/que-es-lifi-como-funciona-38167>.
- [27] IEEE, «802.15.7,» 23 Abril 2019. [En línea]. Available: <https://ieeexplore.ieee.org/document/8751172>.
- [28] IEEE, «RFC4944,» 29 Octubre 2018. [En línea]. Available: <https://ieeexplore.ieee.org/document/8581742>.
- [29] «RFC6282,» September 2011. [En línea]. Available: <https://tools.ietf.org/html/rfc6282>.
- [30] P. Lea, «6LoWPAN topology,» de *Internet of Things for Architects*, Packt, 2018.
- [31] P. Lea, «6LoWPAN topology,» de *Internet of Things for Architects*, Packt, 2018.
- [32] I. D. Technology, 29 Noviembre 2018. [En línea]. Available: <https://www.idt.com/document/dst/zwir4512-datasheet>.
- [33] Thread Group, «What is Thread,» [En línea]. Available: <https://www.threadgroup.org/What-is-Thread>.
- [34] S. Labs. [En línea]. Available: <https://www.silabs.com/documents/public/data-sheets/efr32mg1-datasheet.pdf>.
- [35] IEEE Standards Association, «802.11,» 14 Diciembre 2016. [En línea]. Available: https://standards.ieee.org/standard/802_11-2016.html.
- [36] K. Shaw, «Network World,» 22 Octubre 2018. [En línea]. Available: <https://www.networkworld.com/article/3239677/the-osi-model-explained-how-to-understand-and-remember-the-7-layer-network-model.html>.
- [37] P. Lea, «IEEE 802.11 suite of protocols and comparison,» de *Internet of Things for Architects*, Packt, 2018.
- [38] E. Systems, Noviembre 2018. [En línea]. Available: https://www.mouser.es/datasheet/2/891/0a-esp8266ex_datasheet_en-1223899.pdf.
- [39] R. Muñoz, «Vodafone estrenará el 5G en España el 15 de junio con la red de Huawei,» *El País*, 10 Junio 2019.
- [40] P. Lea, «Governance models and standards,» de *Internet of Things for Architects*, Packt, 2018.
- [41] SIMCom, 01 Agosto 2013. [En línea]. Available: https://cdn-shop.adafruit.com/datasheets/sim800h_hardware_design_v1.00.pdf.

- [42] «Bibing,» [En línea].
Available:
<http://bibing.us.es/proyectos/abreproy/11435/fichero/Documentaci%C3%B3n+%252F07+-+Capitulo+3+-+La+Norma+IEEE+802.pdf>.
- [43] A. Technologies. [En línea].
Available:
https://www.digchip.com/datasheets/download_datasheet.php?id=1113762&part-number=MGA-22103-BLKG&type=prod.
- [44] Murata, «Sub-G Module Data Sheet,» [En línea].
Available: https://wireless.murata.com/pub/RFM/data/type_abz.pdf.
- [45] P. Lea, «Sigfox topology,» de *Internet of Things for Architects*, Packt, 2018.
- [46] P. Lea, «Sigfox physical layer,» de *Internet of Things for Architects*, Packt, 2018.
- [47] Pycom, «SiPy Datasheet,» [En línea].
Available:https://docs.pycom.io/.gitbook/assets/specsheets/Pycom_002_Specsheets_SiPy_v2.pdf.
- [48] Pycom, «SiPy Datasheet,» [En línea].
Available:
https://docs.pycom.io/.gitbook/assets/specsheets/Pycom_002_Specsheets_SiPy_v2.pdf.
- [49] J. E. S. Rodriguez, «Telecomundo,» 14 Octubre 2016. [En línea].
Available: <https://telecomundo.wordpress.com/2016/10/14/modulaciones-digitales-ask-fsk-y-psk/>.
- [50] E. Gómez, «Rincón Ingenieril,» 31 Octubre 2017. [En línea].
Available: <https://www.rinconingenieril.es/funciona-puerto-serie-la-uart/>.
- [51] jecrespom, «I2C,» 9 Julio 2017. [En línea].
Available: <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>.
- [52] I2C-BUS, «MultiMaster,» [En línea]. Available: <https://www.i2c-bus.org/multimaster/>.
- [53] MicroPython, [En línea]. Available: <https://micropython.org/>.
- [54] J. P. P. y. A. Gardey, «Definición.DE,» 2008. [En línea].
Available: <https://definicion.de/html/>.
- [55] G. C. C. y. A. M. D. Sturm Jack, «EcuRed,» [En línea].
Available: https://www.ecured.cu/Lenguaje_de_marcado.
- [56] «masadelante,» [En línea]. Available: <https://www.masadelante.com/faqs/css>.
- [57] Cellnex, Sigfox, «Cellnex Telecom y SIGFOX impulsan el ecosistema,» 14 Julio 2015. [En línea].
Available:https://www.cellnextelecom.com/content/uploads/2015/07/150714_Jornada-IOT_def.pdf.
- [58] Pycom, «RGB LED,» [En línea].
Available: <https://docs.pycom.io/tutorials/all/rgbled.html>.
- [59] dmitryelj, «GitHub,» 29 Enero 2017. [En línea].
Available: https://github.com/dmitryelj/LoPyI2CLCD/blob/master/lopy_lcd.py.
- [60] Pycom, «I2C,» [En línea].
Available: <https://docs.pycom.io/firmwareapi/pycom/machine/i2c.html>.
- [61] dnear1, 27 Enero 2019. [En línea].
Available:https://forum.pycom.io/topic/4282/attributeerror-i2c-object-has-no-attribute-__exit__/2.

- [62] Pycom, «Updating Firmware,» [En línea].
Available: <https://docs.pycom.io/gettingstarted/installation/firmwaretool.html>.
- [63] Sigfox, «Login,» [En línea]. Available: <https://id.sigfox.com/login>.
- [64] Sigfox, «DevKit's indentification,» [En línea].
Available: <https://buy.sigfox.com/activate/devkit/ES>.
- [65] C. HOFFMAN, «How-To Geek,» 19 Febrero 2019. [En línea].
Available: <https://www.howtogeek.com/122641/how-to-forward-ports-to-a-virtual-machine-and-use-it-as-a-server/>.
- [66] Apachefriends, «XAMPP,» [En línea].
Available: <https://www.apachefriends.org/es/index.html>.
- [67] Pycom, «Pycom Network Sigfox,» [En línea].
Available: <https://docs.pycom.io/firmwareapi/pycom/network/sigfox.html>.
- [68] P. Lea, «Cellular connectivity,» de *Internet of Things for Architects*, Packt, 2018.

Glosario

IP	Internet Protocol
WPAN	Wireless Personal Area Network
LE	Low Energy
GPL	General Public License
FHSS	Frequency Hopping Spread Spectrum
BPSK	Binary Phase Shift Keying
O-QPSK	Override Quadrature Phase Shift Keying
ASK	Amplitude Shift Keying
GFSK	Gaussian Frequency Shift Keying
FSK	Frequency Shift Keying
WLAN	Wireless Local Area Network
PLC	Power Line Controller
AES	Advanced Encryption Standard
OSI	Open System Interconnection
AP	Access Point
BSS	Infrastructure Basic Set Service
IBSS	Independent Basic Set Service
WAN	Wide Area Network
LoRa	Long Range Radio
ITU	International Telecommunication Union
TBD	To Be Determined
AMPS	Advanced Mobile Phone System
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
GSM	Global System for Mobile communications
GPRS	General Packet Radio Service
EDGE	Enhanced Data rates for GSM technology
1xRTT	1 times Radio Transmission Technology
FDMA	Frequency Division Multiple Access
WCDMA	Wideband Code Division Multiple Access
PSTN	Public Switched Telephone Network
LTE	Long Term Evolution
WiMAX	Worldwide interoperability for Microwave Access
LPWAN	Low Power Wide Area Network
3GPP	3 rd Generation Partnership Project
MAC	Medium Access Control
OFSS	Orthogonal Sequence Spread Spectrum
GFSK	Gaussian Frequency Shift Keying
RFTDMA	Random Frequency and Time Division Multiple Access
UNB	Ultra Narrow Band
UART	Universal Asynchronous Receiver-Transmitter
I ² C	Inter-Integrated Circuit
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
SPI	Serial Peripheral Interface
UTF-8	8-bit Unicode Transformation Format

SDA	Serial Data
SCL	Serial Clock
NAT	Network Address Translation
GPS	Global Positioning System
EPS	Escuela Politécnica Superior

Anexos

A Manual de instalación

Al recibir el dispositivo SiPy se deben seguir una serie de acciones para activar el dispositivo, y poder utilizar Sigfox. Todos los contenidos presentados en este manual están basados en el manual oficial proporcionado por el fabricante [62].

Actualizar dispositivo

Cuando se disponga del dispositivo el primer paso que se debe tomar es el de actualizar el software. Para ello, debemos descargar una herramienta facilitada por el fabricante.

Debemos conectar el dispositivo al ordenador mediante el cable UART a USB y a la vez, encender el dispositivo en modo actualización. Para encender el dispositivo en modo actualización debemos conectar el pin “P2” con “GND” y reiniciar el dispositivo.

Si todo ha ido bien, veremos la siguiente ventana en la que podremos seleccionar el puerto en el que se ha reconocido el dispositivo:

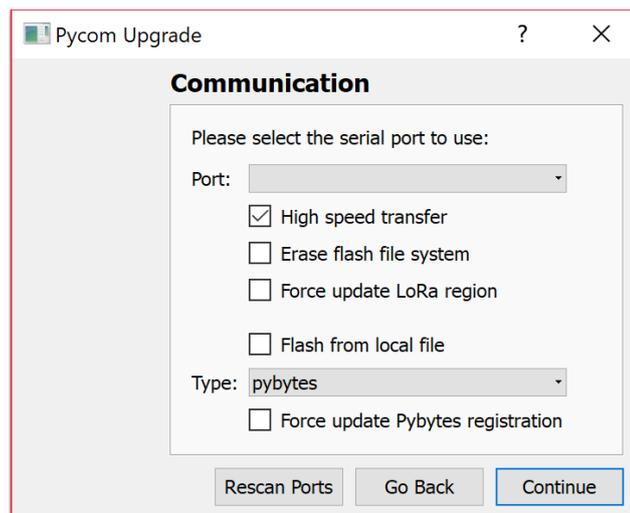


Figura 12-Actualización SiPy

Continuamos con los pasos del programa y se finalizará la actualización en la que nos mostrarán los siguientes parámetros que se necesitarán para la activación del dispositivo en la página oficial de Sigfox: Sigfox ID y Sigfox PAC.

Activación en Sigfox

Una vez se dispone de los parámetros Sigfox ID y Sigfox PAC, deberemos acceder a la página oficial de Sigfox y crear un usuario para registrar los dispositivos bajo el nombre de ese usuario.

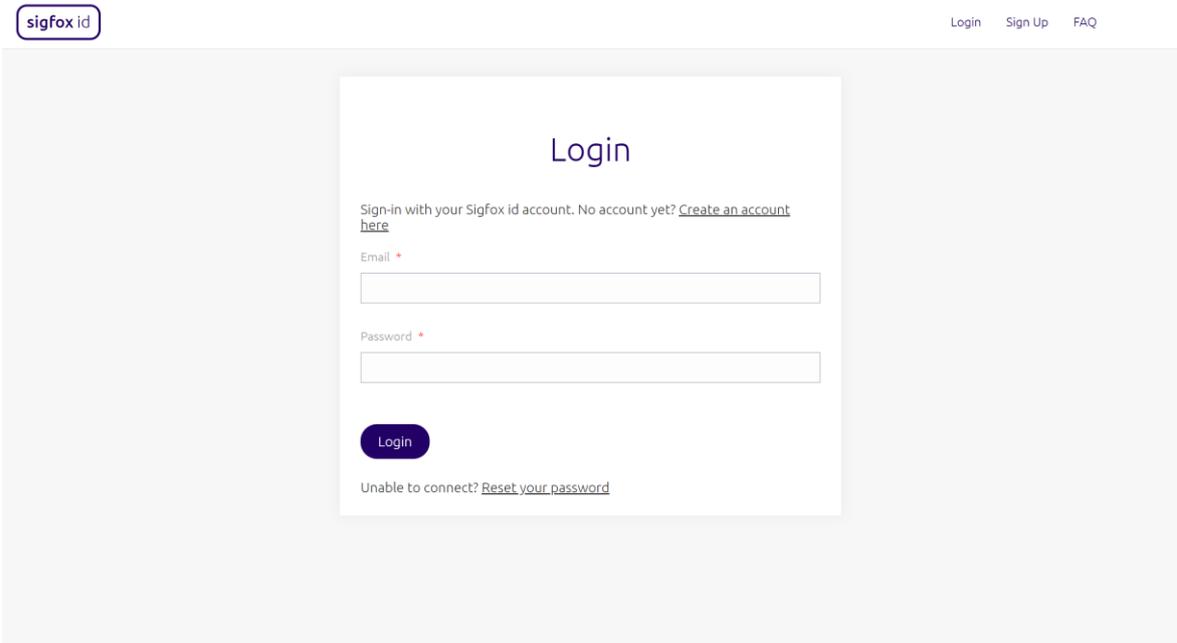


Figura 13-Portal registro de usuario Sigfox [63]

Una vez creado el usuario, habrá que registrar los dispositivos en el apartado de compras. Se nos pedirá el ID del dispositivo y el PAC, obtenidos previamente.

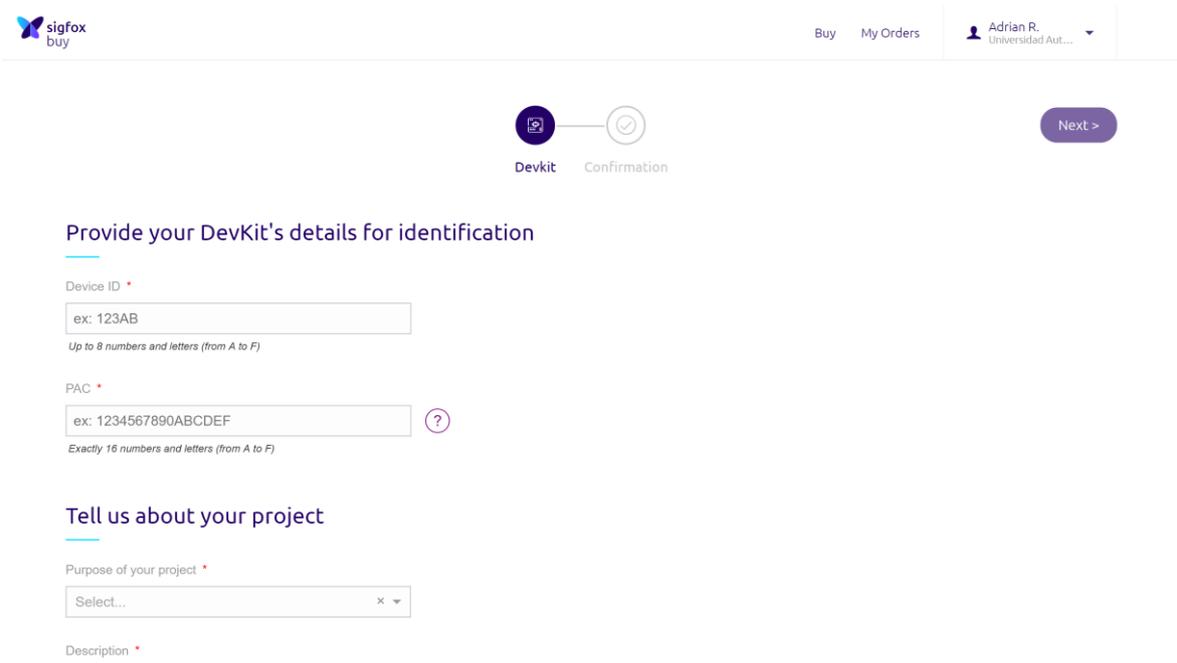


Figura 14-Portal registro dispositivo Sigfox [64]

De esta manera ya quedarían registrados los dispositivos bajo el usuario acreditado.

Entorno de desarrollo

Para poder programar los dispositivos haremos uso de un editor de texto llamado Atom. Dentro de este editor, podremos instalar el paquete que proporciona el fabricante para programar las placas, el paquete se llama Pymakr.

Para ello solo hace falta buscar el paquete dentro de Atom e instalarlo.

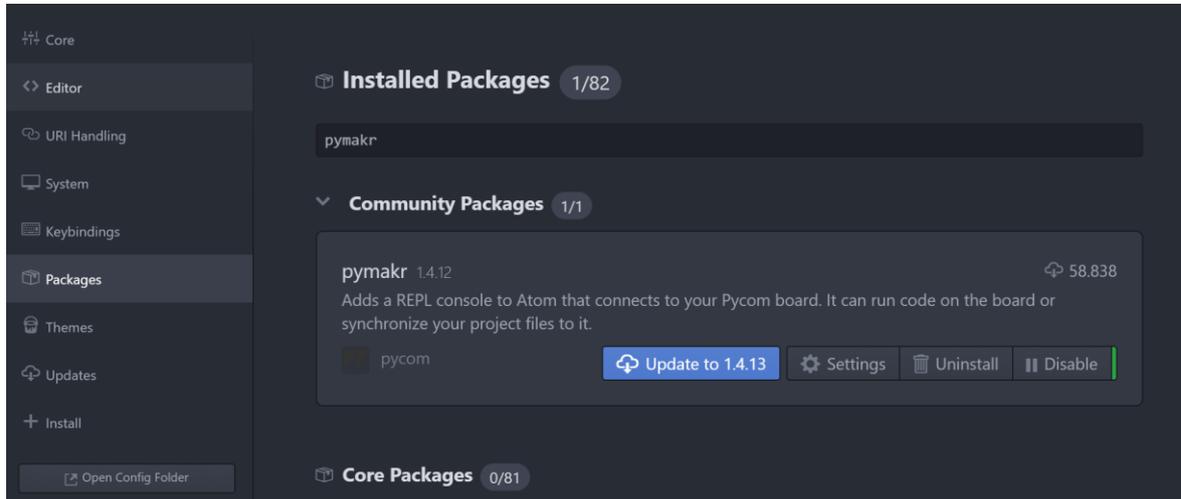


Figura 15-Paquete a instalar en Atom

Gracias a esta extensión podremos conectarnos a la placa y ejecutar el código directamente desde el terminal que proporciona.

PuTTY

Otro método para programar el dispositivo es mediante PuTTY. Una vez hemos instalado PuTTY en nuestro ordenador, al abrir el programa nos encontramos un cuadro en el que tendremos que configurar los campos de la conexión que se va a realizar. En este caso habrá que seleccionar:

- Connection type: Serial.
- Serial line: COM (número de puerto asignado por el ordenador en Windows) o /dev/tty[USB0] (puerto asignado en linux).
- Speed: 115200.

Quedaría de esta manera:

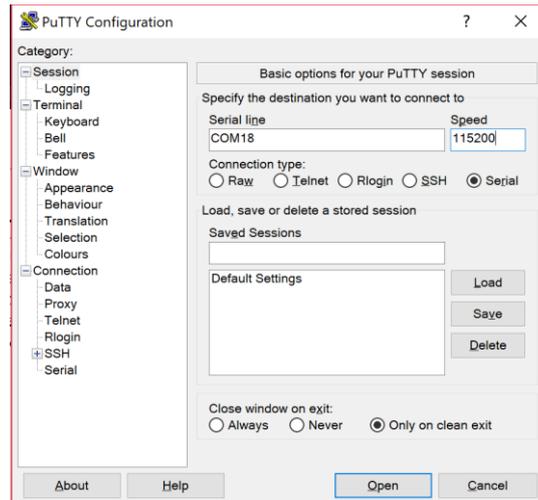


Figura 16-Configuración conexión PuTTY Windows

Cuando abramos la conexión se nos abrirá un terminal. Reiniciamos el dispositivo dando al botón de reset y a continuación, si todo ha salido bien, veremos unos mensajes por consola indicando que se ha reiniciado el dispositivo.

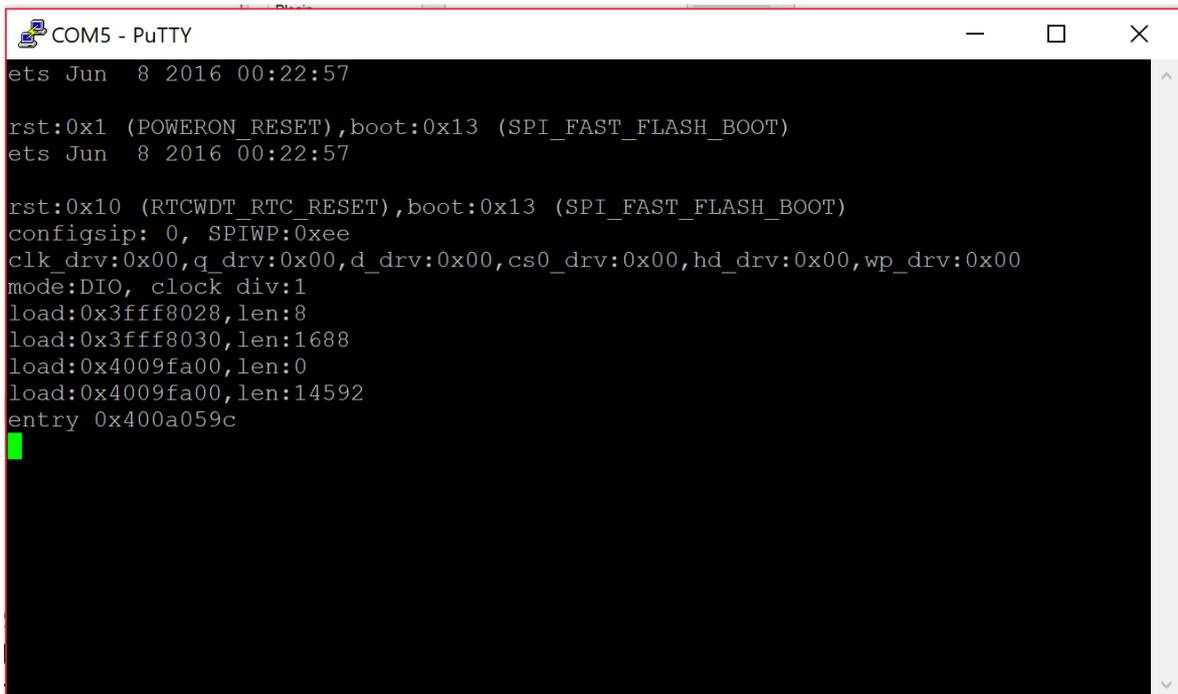


Figura 17-Salida del terminal PuTTY

Conexión WiFi

El dispositivo dispone de conexión WiFi, la cual nos será útil para conectar el dispositivo al paquete Pymaker de Atom y para acceder a los archivos que contiene el dispositivo mediante FileZilla Client.

Para establecer esta conexión, buscaremos la red que genera el dispositivo llamada sipy-wlan-XXXX cuya contraseña es www.pycom.io y de esta manera se conectará el ordenador con el dispositivo.

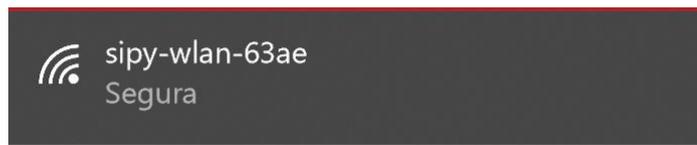


Figura 18-Red WiFi del dispositivo SiPy

Conexión FTP

Para implementar la conexión FTP (File Transfer Protocol) se usará el programa FileZilla que se obtiene de la página oficial.

Previamente se debe conectar el dispositivo SiPy por WiFi al ordenador, para que este protocolo funcione correctamente.

Para configurar la conexión FTP, dentro del programa FileZilla se debe buscar el gestor de sitios. El icono que se encuentra arriba a la izquierda y al clicar encima se abrirá la ventana de configuración del nuevo sitio que se rellenará con los siguientes datos:

- Protocolo: FTP – Protocolo de Transferencia de Archivos
- Servidor: 192.168.4.1
- Cifrado: Usar sólo FTP plano (inseguro)
- Modo de acceso: Normal
- Usuario: micro
- Contraseña: python

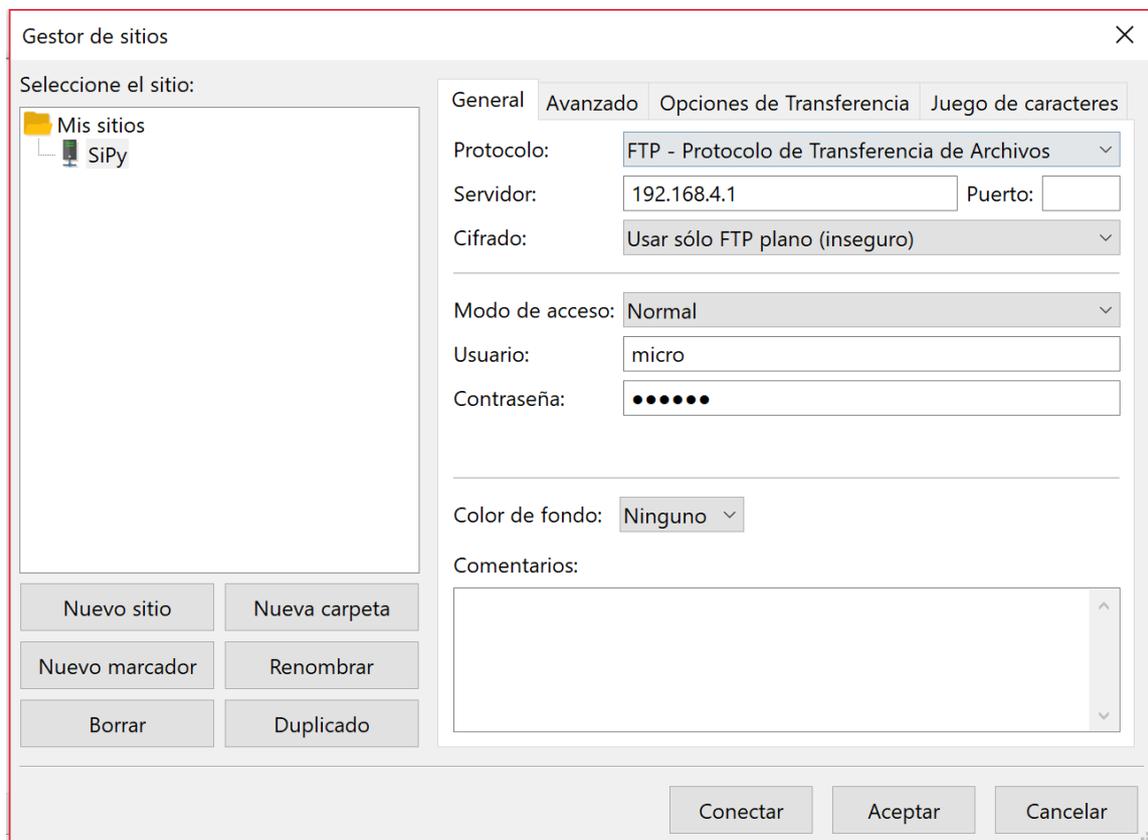


Figura 19-Configuración de sitio en FileZilla General

Cambiamos a la pestaña de Opciones de Transferencia y seleccionamos:

- Modo de transferencia: Pasivo
- Marcamos Limitar el número de conexiones simultáneas
- Número máximo de conexiones 1

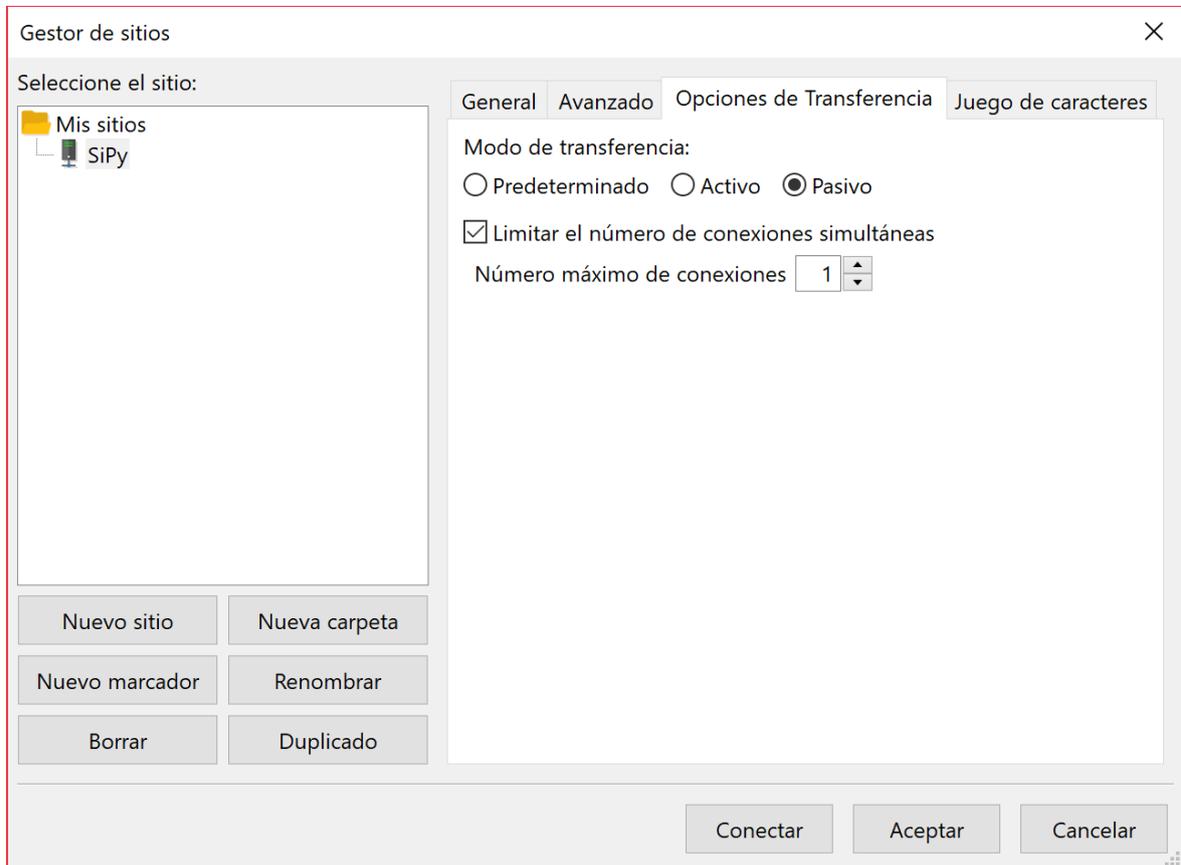


Figura 20-Configuración de sitio en FileZilla Opciones de Transferencia

Una vez finalizados los ajustes al conectarse se deberían ver las carpetas contenidas dentro del dispositivo.

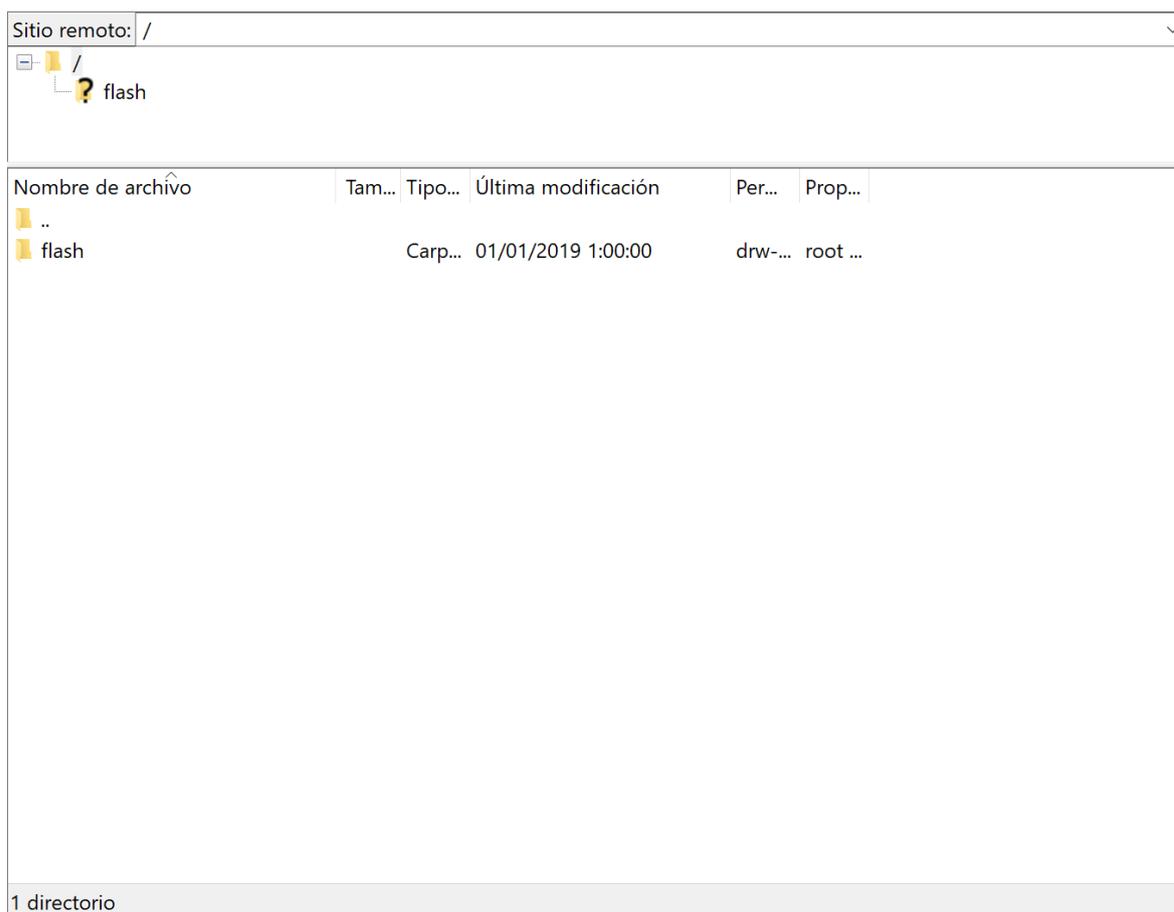


Figura 21-Directorios contenidos en el dispositivo SiPy

Máquina virtual

La máquina utilizada es Oracle VirtualBox, que se podrá descargar desde su página oficial.

Para configurar la red de manera adecuada debemos configurar el reenvío de puertos [65]. Para ello se accede a la configuración de la máquina y en el apartado Red debemos elegir una conexión de tipo NAT. Desplegando el apartado de configuraciones avanzadas dentro de Red, configuraremos el reenvío de puertos de la siguiente manera:

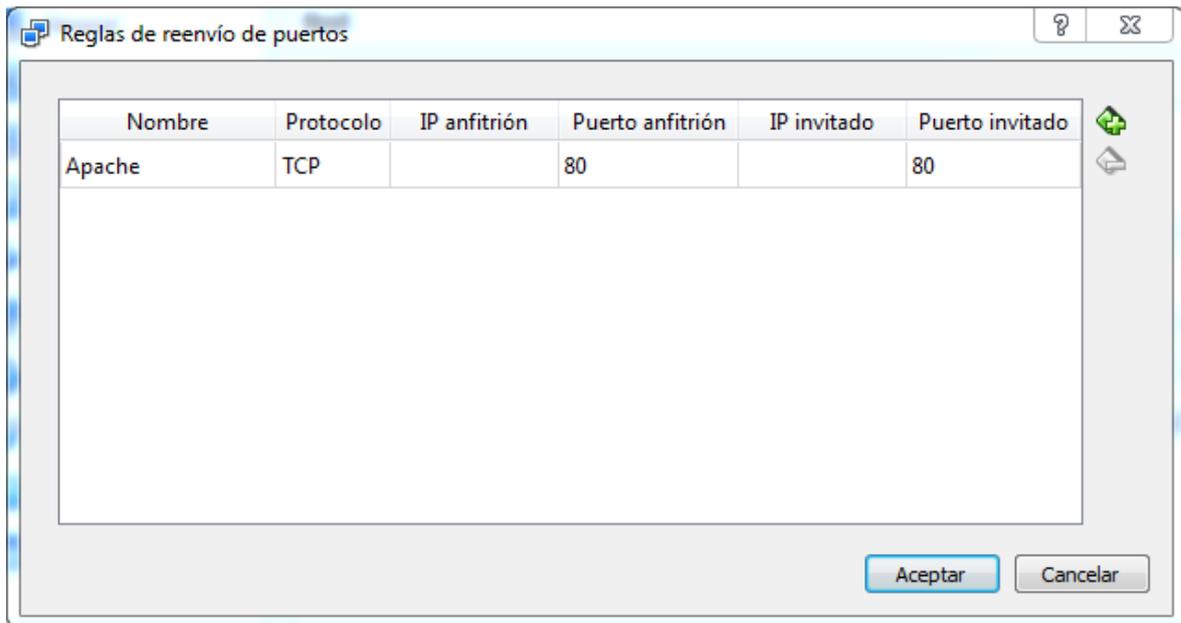


Figura 22-Configuración reenvío de puertos

Servidor Web

Para instalar LAMPP se seguirán los siguientes pasos:

1. Acceder a la página oficial y descargar el archivo correspondiente a Linux. [66]
2. Una vez tengamos el archivo descargado, introduciremos en el terminal la ruta en la que se encuentra y ejecutaremos los siguientes comandos:

```
>> sudo su
>> chmod + xampp-linux-x64-7.3.6-0-installer.run
>> ./xampp-linux-x64-7.3.6-0-installer.run
```

Cuando se completen estos pasos, se abrirá el instalador y daremos a continuar hasta que acabe la instalación.

Para iniciar, parar o reiniciar el servidor se utilizarán los siguientes comandos:

- Iniciar: `sudo /opt/lampp/lampp start`
- Parar: `sudo /opt/lampp/lampp stop`
- Reiniciar: `sudo /opt/lampp/lampp restart`

A la hora de manipular los archivos y el puerto serie desde el servidor, necesitaremos conceder permisos al usuario actual del equipo. Para ello deberemos utilizar estos comandos:

- Permisos LAMPP: `sudo chmod -R 777 /opt/lampp`
- Permisos para acceder a los puertos serie del sistema: `sudo adduser [YOURUSER] dialout`
- Permisos para que el servidor acceda al puerto que se está manipulando: `sudo chmod 666 /dev/ttyUSB0`

B Montaje

En este anexo se mostrarán todas las conexiones que han sido necesarias durante el desarrollo del proyecto. A lo largo de todo el proceso se tomará como referencia el pinout de la placa SiPy:

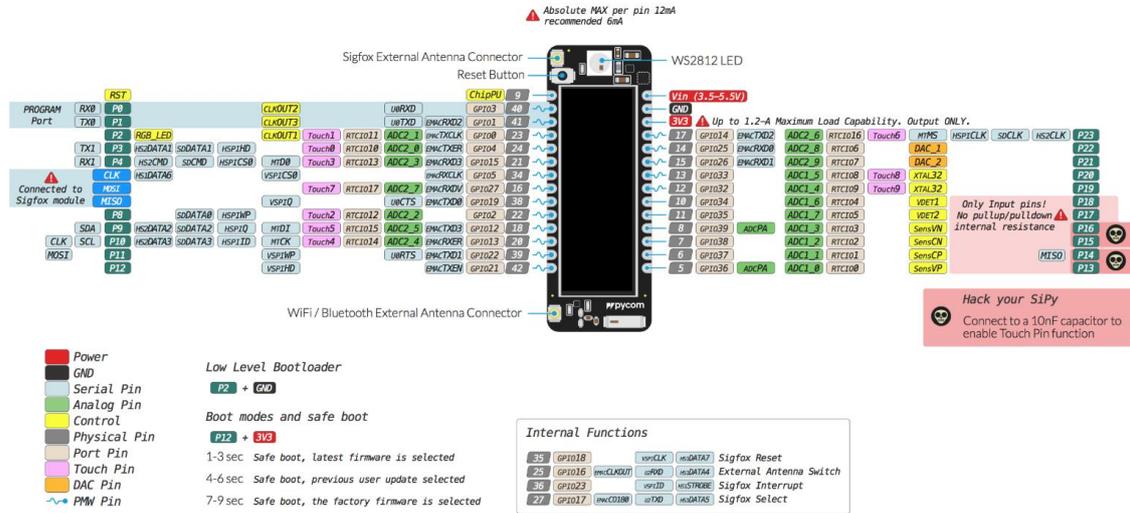


Figura 23-SiPy Pinout [46]

Pantalla OLED 0,19"

Como hemos visto en el apartado de diseño, la pantalla se conectará por medio de I²C. Observando los pines de la placa y los de la pantalla, las conexiones quedarían de esta manera:

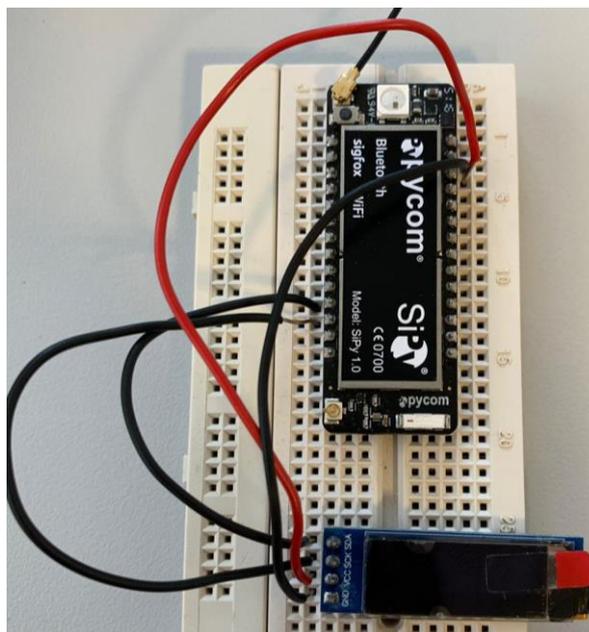


Figura 24-Conexiones pantalla OLED 0.91"

HTU21D

Este sensor comparte los mismos buses I²C que la pantalla, pero se le asignará una dirección de bus diferente. El montaje de este sensor quedaría así:

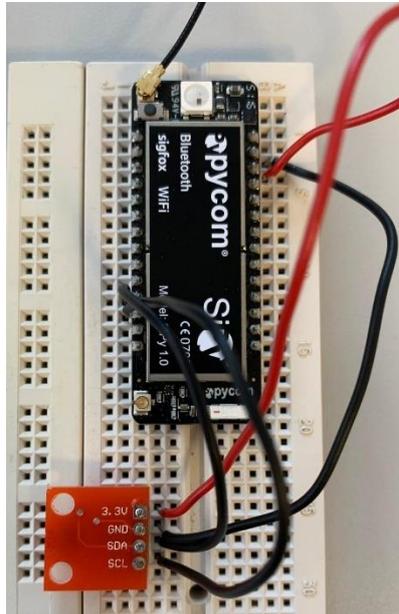


Figura 25-Conexiones sensor HTU21D

UART

Las conexiones tendrían este aspecto:

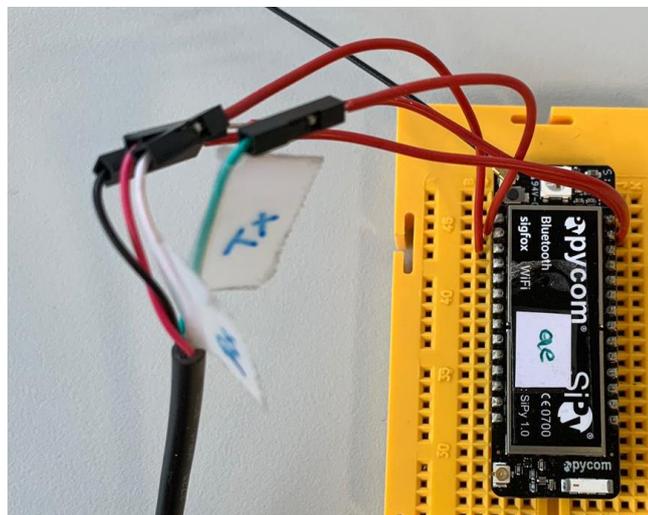


Figura 26-Conexión UART

