

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN



**“DESARROLLO DE UN SISTEMA DE
CONTROL DE EQUIPOS MIDI A TRAVÉS DE
WI-FI SOBRE EL PROTOCOLO OPEN
SOUND CONTROL”**

David Abramian Petrosian

MADRID, 2014



**PROYECTO FIN
DE GRADO**

Autor: David Abramian Petrosian

Tutor: Francisco Javier Casajús Quirós

Título:

DESARROLLO DE UN SISTEMA DE CONTROL DE EQUIPOS MIDI A TRAVÉS DE WI-FI
SOBRE EL PROTOCOLO OPEN SOUND CONTROL

Tribunal calificador:

Presidente: José Fermín Parera Bermúdez

Vocal: José Manuel Menéndez García

Secretario: Francisco Javier Casajús Quirós

Suplente: Mariano García Otero

Fecha de entrega:

Resumen del trabajo

La finalidad de este proyecto es el desarrollo de un dispositivo que permita realizar de manera inalámbrica el control de toda clase de equipos MIDI, tales como sintetizadores o mesas de mezclas. Las aplicaciones prácticas que puede tener un dispositivo de estas características son múltiples, tales como simplificar el conexionado en un estudio de grabación, controlar a distancia instrumentos musicales, y facilitar el uso de dispositivos portables y de amplio acceso tales como teléfonos móviles y tablets para controlar diversos equipos de audio.

El desarrollo del proyecto consistirá en primer lugar en la búsqueda y adquisición de los componentes hardware y software necesarios para el diseño del dispositivo, y en segundo lugar del diseño, programación y puesta en funcionamiento de un prototipo del mismo.

Palabras clave: MIDI, Open Sound Control, OSC, Wi-Fi, inalámbrico

The aim of this project is the development of a device with which to be able to wirelessly control any kind of MIDI equipment, such as synthesizers and audio mixers. The practical applications of such a device are numerous, among which we can count simplifying the wiring in a recording studio, remotely controlling musical instruments, and facilitating the use of portable and easily accessible devices, like mobile phones and tablets, to control audio equipment.

The development of this project will require first the search and acquisition of the various hardware and software tools needed for the design of the device, and second the design, programming and setting up a prototype of the device.

Agradecimientos

Quisiera agradecer los buenos consejos recibidos de Virginia Yagüe Jiménez y José Luis Blanco Murillo

Índice

Acrónimos y glosario	5
1. Introducción	6
1.1. Introducción y objetivos	6
1.2. Estado de la tecnología	8
2. Desarrollo	10
2.1. Preliminares	10
2.1.1. OSC	10
2.1.2. TouchOSC	10
2.1.3. MIDI	11
2.2. Desarrollo	12
2.2.1. Búsqueda de software para transmisión de mensajes OSC	12
2.2.2. Adquisición y pruebas de TouchOSC	13
2.2.3. Diseño de un interfaz para TouchOSC	13
2.2.4. Hardware	14
2.2.5. Pruebas iniciales sobre el MCU y el módulo Wi-Fi	15
2.2.6. Programación del MCU	17
2.2.7. Diseño del interfaz de salida	19
2.3. Finalización del diseño	22
3. Resultados y conclusiones	24
3.1. Resultados	24
3.2. Conclusiones	26
3.3. Mejoras y extensiones	26
4. Bibliografía	28
4.1. Documentos	28
4.2. Páginas web	28
4.3. Procedencia de las imágenes, por orden de aparición	29
Anexo A: Manual de usuario	30
Alimentación	30
Conexión a un equipo MIDI receptor	30
Conexión a una red Wi-Fi	31
Configuración de TouchOSC	31

Anexo B: Hardware y software empleado	32
Hardware	32
Software	33
Anexo C: Composición de mensajes OSC y MIDI	35
Anexo D: Componentes software del sistema y pseudocódigo del programa..	36

Acrónimos y glosario

CNMAT: *Center for New Music and Audio Technology, UC Berkeley*

ETSIT: Escuela Técnica Superior de Ingenieros de Telecomunicación

DAW: (*Digital Audio Workstation*) Estación de trabajo de audio digital

GAPS: Grupo de Aplicaciones de Procesado de Señales

MCU: (*Microcontroller unit*) Microcontrolador

MIDI: Music Instrument Digital Interface

OSC: Open Sound Control

PC: (*Personal Computer*) Ordenador Personal

RSSI: (*Received Signal Strength Indicator*) Indicador de fuerza de la señal recibida

TI: Texas Instruments

UART: (*Universal Asynchronous Receiver-Transmitter*) Transmisor Receptor
asíncrono universal

URL: (*Uniform Resource Locator*) Localizador de recursos uniforme

1. Introducción

1.1 Introducción y objetivos

El estándar MIDI ha cambiado el mundo del audio y la música desde su publicación en el año 1983. Introduciendo para diversos instrumentos electrónicos la capacidad de comunicarse entre sí y con ordenadores, el estándar permitía controlar varios equipos diferentes desde un único instrumento, facilitando mucho la labor del músico, del productor y del ingeniero. La simplicidad del estándar de comunicaciones y del hardware de MIDI, junto con su eficiencia, lo convirtieron en un estándar imprescindible, empleado universalmente durante más de 30 años en el mundo de la música a todos los niveles, desde el músico amateur hasta los grandes estudios y empresas de producción de contenido multimedia.

La contribución de MIDI al mundo de la música no radica únicamente en las facilidades que concede desde el punto de vista técnico, sino que también influyó la creatividad de la escena musical mediante las nuevas posibilidades que otorgaba a los músicos. Junto con los sintetizadores digitales, el estándar permitió reproducir diferentes sonidos pregrabados mediante la pulsación de un botón, lo que posibilitó la inclusión en las representaciones en directo de sonidos que nunca antes podrían haberse empleado. La combinación del estándar MIDI y los sintetizadores y secuenciadores digitales favoreció la creación de nuevos tipos de sonidos y de música, reduciendo los requerimientos necesarios para poder crearla, y permitiendo de este modo un nuevo flujo de creatividad procedente de músicos individuales o grupos de reducido tamaño.

A pesar de todas sus ventajas y de los beneficios que supuso para la escena musical, el estándar MIDI no carece de imperfecciones. Sus principales defectos tienen que ver con los tiempos de propagación de la señal:

- En la transmisión de señales MIDI, existe una pequeña variación en los tiempos de llegada de la información al dispositivo receptor (*jitter*). Esta variación, aun siendo muy pequeña, puede llegar a ser perceptible.
- Más notablemente, la latencia que se produce entre el momento en que se activa una función en MIDI y el momento en que esta se lleva a cabo puede ser del orden de varios milisegundos, y generalmente es dependiente del número de dispositivos conectados, además de los dispositivos concretos que se conecten y de la potencia del hardware que se encarga de manejarlos.

Los problemas que presenta el estándar MIDI no le han impedido convertirse en una de las principales herramientas en el mundo de la música y el audio durante los últimos 30 años. Sin embargo, los desarrollos tecnológicos que se han producido subsiguientemente hacen que hoy en día se nos presenten

alternativas. Entre las más importantes se encuentra el protocolo Open Sound Control (OSC), desarrollado en la Universidad de California, Berkeley, por Adrian Freed y Matt Wright. Este protocolo está diseñado para permitir la comunicación entre diversos equipos multimedia, al igual que MIDI, pero está optimizado para su uso en redes, permitiendo una mayor interoperabilidad, precisión y flexibilidad.

Algunas de las principales características del protocolo OSC son su sistema de nomenclatura de mensajes similar a URL, la capacidad de transmitir datos simbólicos o numéricos de alta precisión, y el uso de etiquetas temporales de gran resolución. Estas características le permiten transmitir una gran variedad de mensajes, incluyendo mensajes MIDI encapsulados en OSC. Además, puesto que se emplea en sistemas con un ancho de banda muy superior al de las conexiones MIDI, es capaz de resolver los problemas de temporización mencionados anteriormente. La transmisión de etiquetas temporales permite además enviar la información necesaria a gran velocidad para ser reproducida cuando haga falta, en lugar de basarse en las comunicaciones en tiempo real necesarias en MIDI.

El protocolo OSC resuelve muchos de los problemas del estándar MIDI, y tal vez en el futuro sea un posible candidato para tomarle el relevo como el estándar universal de comunicaciones entre instrumentos. Aun así, a día de hoy el estándar MIDI sigue siendo el más empleado, y no se puede cuestionar su idoneidad para muchas de los requerimientos del entorno de la música y el audio. A pesar de que algunos sistemas están incorporando compatibilidad con OSC, la cantidad de equipos que emplean MIDI es enormemente superior, por lo que una transición de un estándar a otro, si se llevara a cabo, llevaría un tiempo considerable.

La capacidad de transmitir mensajes por la red supone sin duda una gran ventaja del protocolo OSC sobre MIDI. Sin embargo, puesto que el protocolo OSC puede transmitir mensajes MIDI, esta capacidad podría ser aprovechada por los sistemas basados en MIDI, siempre y cuando existiera una manera de convertir los mensajes OSC a MIDI antes de entregarlos al dispositivo receptor. Las ventajas que supondría tener la capacidad de enviar mensajes MIDI de manera inalámbrica son numerosas, con aplicaciones tales como:

- Simplificación del cableado en un entorno de estudio. En los estudios de grabación es necesario utilizar una gran cantidad de cables y conexiones, lo cual complica el proceso de grabación y puede dar lugar a resultados no deseados. Mediante la transmisión inalámbrica, cada dispositivo únicamente necesitaría un pequeño receptor conectado a su entrada MIDI, pudiéndose modificar las conexiones meramente variando la dirección de destino de los distintos mensajes que se envían.
- Favorecimiento del desarrollo de nuevas aplicaciones con capacidad de control de equipos MIDI. Debido a la alta disponibilidad que existe hoy en día de teléfonos móviles y de tablets, la sustitución del interfaz físico MIDI

por una conexión Wi-Fi concedería a una gran cantidad de dispositivos y de personas la capacidad de controlar equipos MIDI a distancia. Puesto que estos dispositivos se manejan mediante pantallas táctiles, permiten un control preciso similar al que permitiría un equipo físico. De esta manera se favorecería el desarrollo de aplicaciones específicas para estas plataformas, y surgiría una nueva manera de utilizar equipos MIDI y de crear música con ellos. Más aun, un enlace bidireccional permitiría utilizar estos dispositivos como mezcladores simples, o incluso como estaciones de trabajo de audio digital.

En este proyecto se pretende, pues, crear el prototipo de un dispositivo que se valga de la transmisión por red Wi-Fi de mensajes OSC para permitir el control inalámbrico de equipos MIDI. Los principales objetivos del proyecto son los siguientes:

- Desarrollo del prototipo de un dispositivo que permita recibir por una red Wi-Fi mensajes OSC que encapsulen información MIDI, y transmitirla a un equipo controlable por MIDI.
- El dispositivo debe ser autónomo, evitándose en la medida de lo posible la necesidad de intervención física de un operario a la hora de utilizarlo.
- El dispositivo debe ser capaz de controlar en tiempo real un equipo MIDI. El retardo en la comunicación debe ser el mínimo posible.
- El dispositivo debe ser lo bastante compacto como para ser fácilmente conectable a cualquier tipo de equipo MIDI.

En el resto de este documento, se documenta el procedimiento seguido para el desarrollo de un dispositivo de estas características, así como los materiales empleados y los resultados obtenidos.

1.2 Estado de la tecnología

En la actualidad existen varios dispositivos comerciales que permiten comunicaciones MIDI inalámbricas. A continuación se detallarán las características de los principales dispositivos, proporcionadas por el fabricante:

- MIDIJet Pro: Permite comunicaciones bidireccionales en la banda de 2,4 GHz, con un alcance de 150 m y con un máximo de 29 pares de transmisores operando simultáneamente. El precio de una pareja de transmisores es de 450 \$ (aproximadamente 360 €).
- CME WIDI-X8: Permite comunicaciones bidireccionales en la banda de 2,4 GHz, con un alcance de 80 m, y un máximo de 64 canales de transmisión radio. El precio de una pareja de transmisores es de 250 \$ (aproximadamente 200 €).

- Kenton MIDISStream: Se comunica unidireccionalmente en la banda UHF de 869,85 MHz en el modelo europeo y 914,5 MHz en el americano, con un alcance de 80 m. No ofrece soporte para utilizar varias unidades simultáneamente. El precio de una pareja transmisor/receptor es de 300 £ (aproximadamente 375 €).
- Starr Labs Air Power 2: Permite comunicaciones bidireccionales en la banda de 2,4 GHz con un alcance de 75m, y un máximo de 16 canales de transmisión radio. El precio de una pareja de transmisores es de 300 \$ (aproximadamente 240 €).
- Panda Audio midiBeam: Permite comunicaciones unidireccionales en la banda de 2,4 GHz con un alcance de 250 m, y un máximo de 72 parejas transmisor/receptor operando simultáneamente. El precio de una pareja transmisor/receptor es de 125 €.

A pesar de la amplia disponibilidad de productos, todos ellos están caracterizados por un precio elevado, que limita la viabilidad del sistema, y reduce su mercado potencial a grandes empresas y organizadores de conciertos. Por lo tanto, un producto con un precio más asequible podría favorecer la implementación de sistemas MIDI inalámbricos en un mayor número de ámbitos.

2. Desarrollo

A continuación se describe el procedimiento que se siguió para el desarrollo del proyecto. Cabe señalar que el trabajo se realizó en el laboratorio del Grupo de Aplicaciones de Procesado de Señales (GAPS) de la ETSIT. En el Anexo B se puede encontrar información más detallada sobre el hardware y el software empleado.

2.1 Preliminares

En esta sección se introducirán las diversas tecnologías y estándares en los que estará basado el proyecto y de los que depende. En el anexo C se puede ver un ejemplo del funcionamiento de los mensajes de OSC y MIDI.

2.1.1 OSC

OSC es un protocolo de comunicación entre ordenadores, sintetizadores y otros dispositivos multimedia que está optimizado para la transmisión por red. Los mensajes con los que se comunica emplean un sistema de nombrado simbólico similar a URL, y ofrecen una alta versatilidad tanto en los argumentos transmitidos como en la selección de destinatario. Los mensajes incluyen etiquetas temporales de alta resolución, además de ofrecerse ciertas soluciones para la agrupación de mensajes cuyos efectos deben ser simultáneos. En general, supone una infraestructura versátil para la transmisión de mensajes musicales por la red, y soporta el encapsulado de mensajes MIDI, lo cual lo hace idóneo para su uso en este proyecto.

OSC fue desarrollado en la Universidad de California, Berkeley, en el Center for New Music and Audio Technology (CNMAT), por Adrian Freed y Matt Wright. La primera especificación fue definida en marzo de 2002, y fue seguida por otra en 2009. Tanto el software como la documentación se distribuyen de manera libre, y ha visto ya numerosas implementaciones en diversas áreas tales como de procesado de sonido en tiempo real, sintetizadores software y herramientas de interactividad web, entre otras.

2.1.2 TouchOSC

TouchOSC es una aplicación desarrollada por Hexler y disponible para plataformas iOS y Android que permite utilizar diversos interfaces gráficos para transmitir mensajes musicales de tipo OSC o MIDI por la red. Los interfaces están compuestos por una variedad de botones y sliders, que al ser accionados transmiten por la red a la que está conectado el dispositivo móvil un mensaje asociado a cada uno de ellos. Estos mensajes pueden ser recibidos e interpretados posteriormente por un dispositivo receptor que se encuentre conectado a la misma red, y pueden utilizarse, entre otras cosas, para controlar instrumentos virtuales o DAWs. Así mismo, la aplicación soporta la comunicación bidireccional, por lo que puede emplearse otro

dispositivo con capacidad de transmisión de mensajes OSC para producir cambios en la interfaz de TouchOSC. Sin embargo, esta forma de uso es menos significativa, ya que la aplicación no tiene capacidad de producir más que cambios visuales sin ningún efecto aparte.

Los mensajes OSC y MIDI asociados a cada elemento del interfaz pueden transmitirse a destinatarios diferentes, especificados por su dirección IP y puerto de entrada. Los mensajes MIDI se transmiten encapsulados en OSC, por lo que ambos tipos de mensaje son OSC, pero difieren en su contenido. En adelante se hará referencia a los primeros como mensajes OSC y a los segundos como mensajes OSC/MIDI.

Además de la aplicación en sí, se dispone de un editor de que permite diseñar interfaces personalizados, eligiendo el mensaje concreto que se envía con cada elemento. Esta herramienta resulta indispensable para poder realizar pruebas diversas sobre el dispositivo que se diseñe y comprobar su funcionalidad.

2.1.3 MIDI

El estándar MIDI fue introducido en 1983, y desde entonces ha sido el estándar por defecto para la transmisión de información musical, tanto para el control de equipos como en instrumentos musicales o en herramientas de composición. La definición del estándar describe tanto el protocolo de comunicación como el interfaz digital y los conectores.

La comunicación especificada en el protocolo se basa en la transmisión de mensajes de evento, que especifican los parámetros de notas musicales, y mensajes de control, que controlan otros parámetros relacionados, como el volumen y el panorama. El protocolo permite manejar simultáneamente 16 canales con parámetros independientes. Estos se pueden utilizar para transmitir información a varios instrumentos simultáneamente, para simular múltiples instrumentos, o con otros fines

En lugar de especificar la duración de las notas musicales en los mensajes, o de enviar un stream constante de mensajes mientras se mantenga una nota, se envía un único mensaje que señala el comienzo de una nota, y otro mensaje que señala el final. Uno de los fenómenos más molestos relacionados con el uso de MIDI es el que se produce cuando mensaje de fin de nota no se recibe correctamente y la nota no cesa de sonar hasta que se envía otra vez el mensaje de fin de nota o un mensaje especial de “all notes off”. Incidentes de este tipo no son tolerables en un ambiente profesional de estudio o en directo, por lo que las conexiones entre los equipos deben ser muy fiables.

Los mensajes MIDI se componen, con ciertas excepciones, de un byte de estado, seguido de uno o dos bytes de datos. El byte de estado indica el tipo de mensaje que se envía y el canal al que va dirigido, mientras que los bytes de datos pueden considerarse los argumentos, que pueden incluir, según el tipo de mensaje, información de nota, velocidad, programa (instrumento), presión, valor de control, etc.

En cuanto al interfaz digital, los mensajes se transmiten en serie a 31,25 kbaudios, de manera asíncrona, en formato LSB primero, con un bit de comienzo (0 lógico, corriente activada), 8 bits de datos, y un bit de parada (1 lógico, corriente desactivada).

Los conectores empleados por los equipos MIDI son DIN de 5 pines, y los puertos pueden ser de tres tipos: de entrada (MIDI IN), de salida (MIDI OUT), y de copia de la entrada en la salida (MIDI THRU). Estos últimos pueden emplearse para situaciones en las que se controlen múltiples equipos con soporte para MIDI desde un único dispositivo maestro, enlazándolos en cascada uniendo el puerto MIDI THRU de un equipo con el MIDI IN del siguiente. Este sistema de conexión se puede ver limitado por la degradación de la señal al atravesar múltiples equipos, por lo que la cantidad de aparatos conectados de esta manera no puede ser elevada.

2.2 Desarrollo

2.2.1 Búsqueda de software para transmisión de mensajes OSC

El desarrollo del software necesario para transmitir mensajes MIDI encapsulados en OSC no entra dentro de los objetivos de este proyecto, que se centra únicamente en el dispositivo receptor. Por lo tanto, el primer paso a llevar a cabo consiste en la búsqueda de software existente que cumpla este cometido. Sin embargo, la flexibilidad del protocolo OSC permite que haya más de una manera de enviar información MIDI, pudiéndose variar el orden de los bytes o la cantidad de argumentos del mensaje. Por lo tanto, en este proyecto hemos decidido elegir un único software y compatibilizar el dispositivo con los mensajes que éste envía, en lugar de intentar hacer un dispositivo compatible con todas las posibles opciones.

El primer paso en el desarrollo del proyecto consistió, por lo tanto, en la prospección tecnológica con el fin de encontrar software adecuado para generar mensajes OSC que encapsulen mensajes MIDI. El software seleccionado para este fin fue “TouchOSC” de Hexler, por su versatilidad y adecuación a las necesidades de este proyecto. Las capacidades y características principales de esta aplicación han sido definidas en el apartado 2.1.2.

2.2.2 Adquisición y pruebas de TouchOSC

Tras adquirir el software en su versión para Android e instalarlo en un teléfono móvil con este sistema operativo, se procedió a realizar una serie de pruebas entre el teléfono móvil y un PC para comprobar el funcionamiento y la capacidad de esta aplicación. Estas pruebas tenían como objetivo examinar el formato de la información

transmitida por la aplicación, así como comprobar que la transmisión se realizaba sin problema.

En primer lugar, se utilizó un programa básico de prueba proporcionado por Hexler para comprobar que se establece la comunicación entre el dispositivo emisor y el receptor. Este programa estaba codificado en el lenguaje de programación visual Pure Data, y consistía en un simple volcado de la información recibida por el PC procedente del puerto asignado en TouchOSC para la transmisión de los mensajes OSC (por defecto puerto 8000). Así mismo, permitía comprobar la comunicación en el sentido contrario, mediante la transmisión de un mensaje desde el PC a TouchOSC por el puerto destinado a la recepción de mensajes OSC (por defecto puerto 9000), lo cual debería tener como resultado la alteración del valor de uno de los sliders de la interfaz.

La segunda prueba realizada tenía como finalidad comprobar la capacidad de TouchOSC para transmitir mensajes MIDI. Para ello se utilizó el software “SendSX” de Bome, un sencillo analizador que muestra el tráfico MIDI que entra en el PC por un puerto determinado. Para crear un puerto de entrada compatible con los mensajes OSC/MIDI enviados por TouchOSC se instaló el programa “TouchOSC MIDI Bridge”, destinado específicamente a convertir el tráfico MIDI encapsulado en OSC generado por TouchOSC en señales MIDI, y creando un puerto llamado “TouchOSC Bridge” en el PC, por el que se envían las señales MIDI. Para conseguir que TouchOSC enviara mensajes destinados a TouchOSC Bridge, se configuró el campo “MIDI Bridge” de TouchOSC con la dirección IP del PC.

En ambos casos el resultado de las pruebas ha sido favorable, mostrando que el PC recibía el tráfico procedente de TouchOSC. Sabiendo esto, se realizó una prueba final, que consistió en redirigir el tráfico procedente de TouchOSC a un sintetizador General MIDI (Yamaha TG3000), que respondió a los mensajes y podía ser controlado remotamente por TouchOSC. Por lo tanto se ha determinado con estas pruebas que TouchOSC es un software adecuado para generar los mensajes que posteriormente deberán ser decodificados por el dispositivo que se pretende diseñar.

2.2.3 Diseño de un interfaz para TouchOSC

Para realizar las pruebas relacionadas con los mensajes OSC/MIDI de TouchOSC, ha sido necesario diseñar un interfaz que envíe este tipo de mensajes además de los mensajes OSC normales. Este interfaz es el que se utilizó para todas las pruebas subsiguientes realizadas para comprobar el funcionamiento del proyecto. Para el diseño del interfaz de TouchOSC se ha utilizado el editor proporcionado para este fin por Hexler.

Los interfaces en TouchOSC se organizan en páginas, para compensar por el pequeño tamaño de pantalla de que disponen los dispositivos en los que se usan. En la Figura 1 se muestran las distintas páginas que componen el interfaz creado:

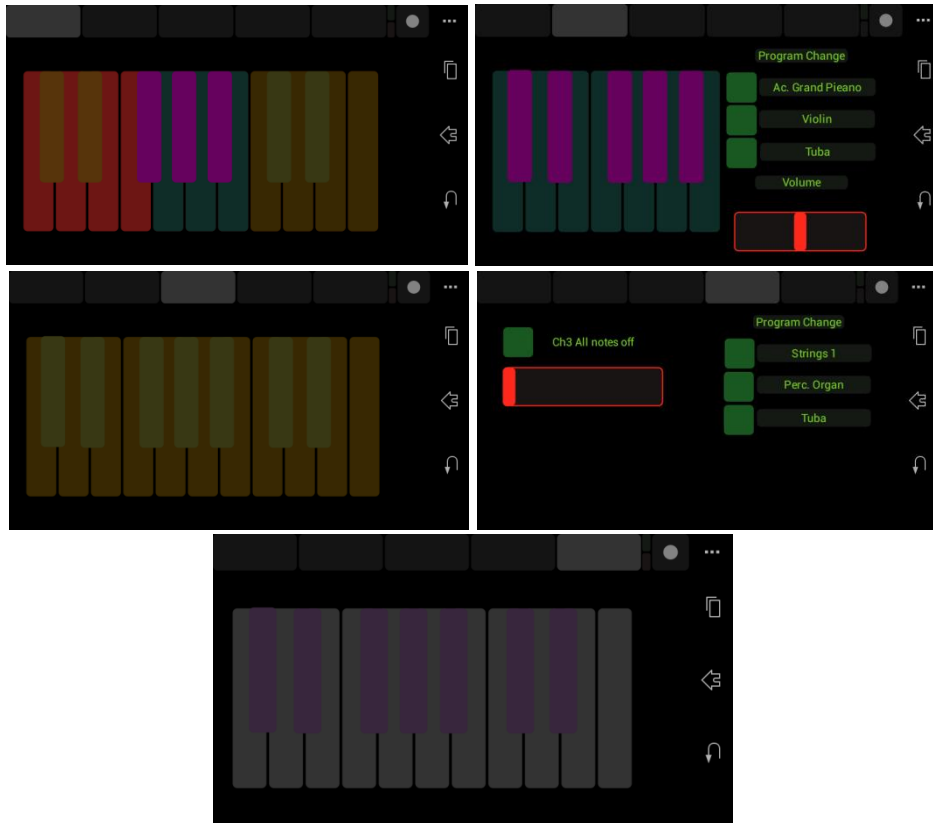


Figura 1 - Páginas del interfaz TouchOSC creado

- La primera página contiene un teclado con teclas pertenecientes a los canales MIDI 1, 2 y 3.
- La segunda página contiene un teclado en el canal 2, con botones para cambiar el programa (variar el instrumento), y un slider que controla el volumen del canal.
- La tercera página contiene un teclado en el canal 3.
- La cuarta página contiene controles de programa y volumen para el canal 3, además de un botón que envía el mensaje de control *all notes off*, que desactiva todas las notas que estén sonando por el canal 3.
- La quinta página contiene un teclado en el canal 10. En el estándar General MIDI, este canal está reservado para los instrumentos de percusión.

2.2.4 Hardware

En esencia, el dispositivo que se pretende diseñar debe ser un sustituto del PC en la última prueba realizada, es decir, debe ser capaz de recibir los mensajes, decodificarlos extrayendo la información MIDI, y transmitirlos al dispositivo al que esté conectado. Para realizar esta labor son necesarios dos componentes principales: una tarjeta de red que permita recibir los mensajes transmitidos por Wi-Fi, y un microcontrolador que se encargue de recibirlos, desempaquetarlos, y transmitirlos mediante una UART a un dispositivo con una interfaz MIDI.

Se ha decidido examinar el catálogo de Texas Instruments (TI) para buscar estos componentes, debido a que todos sus productos están extensamente documentados y poseen programas de prueba que pueden utilizarse para facilitar la familiarización con los productos.

Tras examinar el catálogo, comprobamos que los productos de TI que mejor se ajustan a las necesidades del proyecto son el módulo de Wi-Fi CC3000 y la serie de microcontroladores MSP430. Estos dispositivos se pueden adquirir en forma de placa de desarrollo interconectable, lo cual facilita la labor de establecer la comunicación entre ellos. Finalmente se opta por un kit de evaluación de Wi-Fi, compuesto por una placa de desarrollo MSP-EXP430FR5379, dos módulos Wi-Fi basados en CC3000 producidos por Murata y TiWi, y un router Wi-Fi TL-WR740L.

El diseño completo, por lo tanto, se ha realizado siguiendo el siguiente esquema:

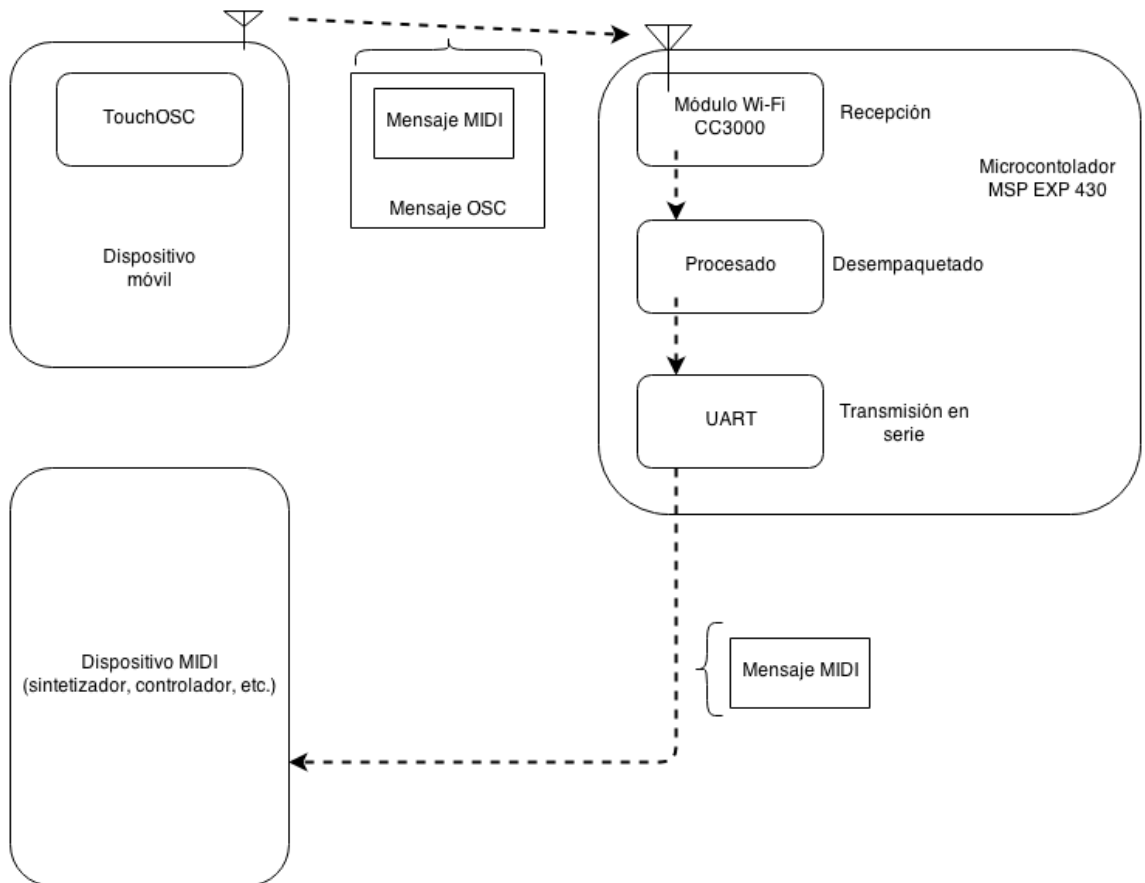


Figura 2 - Esquema general del sistema a diseñar

2.2.5 Pruebas iniciales sobre el MCU y el módulo Wi-Fi

Una vez fue recibido el kit adquirido, se realizaron varias pruebas básicas para familiarizarse con el uso de los dispositivos, además de adquirir el software necesario para su uso. Puesto que el MCU se conecta al PC a través de un interfaz USB Mini-B,

se puede cargar y depurar los programas desde el PC. Para ello es necesario instalar el software Code Composer Studio 5, un entorno de desarrollo de Texas Instruments basado en Eclipse que facilita la integración con los microcontroladores y dispositivos de TI.

El programa de prueba del MCU viene cargado directamente de fábrica, y se ejecuta correctamente según las especificaciones. Para probar el módulo Wi-Fi es necesario conectarlo al MCU mediante los sockets que dispone para ese fin, y cargar el programa de prueba correspondiente. El programa debería responder a los comandos que se le envían desde el ordenador, permitiendo, entre otras cosas, conectarse a una red Wi-Fi, abrir y cerrar sockets UDP, asignar un puerto de escucha determinado a un socket, y transmitir y recibir paquetes.

Para comunicarse con el MCU y enviarle los comandos correspondientes es necesario instalar un hiperterminal, que se comunica en serie con la UART del MCU a través de un puerto COM virtual creado en el PC. En este caso se optó por el software “Putty”, siguiendo las recomendaciones del fabricante. Durante el desarrollo de la mayor parte del proyecto se empleó este hiperterminal para comunicarse con el MCU, pero debido a que con frecuencia perdía la capacidad de comunicarse con el puerto COM virtual, se buscó un software sustitutivo, que en este caso fue “Advanced Serial Port Monitor”. Este nuevo software resolvió los problemas de conectividad que se experimentaron con Putty.

La manera que emplea el módulo Wi-Fi de conectarse a una red consiste en recibir la información necesaria de otro dispositivo que ya esté conectado a esa misma red (PC, teléfono móvil, etc.) y que se lo transmita a través de una aplicación proporcionada por TI, mediante la cual se emiten los datos de la red. Una vez recibidos los datos de conexión, el módulo intenta conectarse a la red, tras lo cual se pueden utilizar el resto de sus funciones. Este procedimiento tiene el nombre de “SmartConfig”. También es posible configurarlo manualmente, pero puesto que el objetivo final del proyecto es crear un dispositivo autónomo, la función SmartConfig es idónea para este propósito.

La primera prueba que se realizó sobre el módulo Wi-Fi consistió en conectarlo a la red que se utiliza en el laboratorio del GAPS, donde se estaba realizando el proyecto. El módulo no fue capaz de conectarse a la red. Aunque la causa no se ha podido determinar con precisión, se debe probablemente a que la red tiene contraseña de seguridad. Para probar el módulo en una red sin contraseña, se instaló el router Wi-Fi incluido en el kit y se configuró una red sin contraseña y dedicada únicamente a los dispositivos empleados en este proyecto. Al emplear esta nueva red, el dispositivo fue capaz de conectarse. Por lo tanto, en adelante ésta fue la red que se empleó para trabajar con el módulo y realizar todas las pruebas.

Tras conseguir que el módulo se conectara a una red, se hizo una prueba más para comprobar que recibía tráfico. Para ello se abrió un socket UDP, se asoció al puerto 8000, correspondiente al puerto al que dirige TouchOSC sus mensajes de salida

OSC, y se comenzó a escuchar el puerto a la espera de mensajes. Al pulsar una tecla en la interfaz de TouchOSC el mensaje fue recibido por el módulo Wi-Fi, por lo que se determinó que el módulo Wi-Fi funcionaba adecuadamente.

Aunque en TouchOSC se puede configurar el puerto al que se dirigen los mensajes OSC, no sucede lo mismo con los mensajes OSC/MIDI, y el puerto por el que se envían no aparece indicado explícitamente en la aplicación, ya que está pensada para funcionar con el software TouchOSC MIDI Bridge. Sin embargo, puesto que para recibir correctamente los mensajes en el MCU es necesario asociar un puerto al socket UDP, ha hecho falta descubrir cuál es el puerto por el que se transmiten los mensajes OSC/MIDI. Para ello se utilizó el software de captura de tráfico de red “Wireshark”, con el que se capturaron los paquetes IP que contenían los mensajes OSC/MIDI, y se comprobó que el puerto al que estaban dirigidos era el 12101.

2.2.6 Programación del MCU

El módulo MSP-EXP430FR5739 es una placa de desarrollo programable por USB que incluye un microcontrolador MSP-430FR5739. Este microcontrolador, junto con el módulo CC3000, posee una gran cantidad de documentación, además de una serie de bibliotecas de funciones y código a las que se puede acceder directamente desde el entorno de desarrollo Code Composer Studio. El estudio de la documentación relacionada con estos módulos, así como del código disponible para ellos, supuso una de las labores que más tiempo consumió en el desarrollo del proyecto.

Como base para desarrollar el programa necesario, se partió de la aplicación de prueba del módulo CC3000 que se ha descrito anteriormente. Puesto que ya contenía el código necesario para realizar las funciones de la conexión a una red mediante SmartConfig y la recepción de datos, se reduciría la cantidad de funciones que sería necesario implementar.

El proceso de programación se realizó en dos fases: la primera se centró en conseguir recibir un mensaje OSC/MIDI procedente de TouchOSC y enviar por una salida digital su contenido MIDI en el formato necesario para poder ser interpretado, y en la segunda se realizaron las modificaciones necesarias para que el proceso fuese automático, sin necesidad de intervención del usuario desde un PC.

Omitiendo los procesos de ensayo y error y las pruebas realizadas, se puede resumir el trabajo realizado en la primera fase en los siguientes pasos:

- Se estableció un tamaño de buffer de lectura de 16 bytes, ya que este es el tamaño de los mensajes que se espera recibir. Esto no significa, sin embargo, que si se recibe un mensaje mientras no se ha leído el anterior, se pierda, ya que los mensajes se acumulan en un buffer de entrada diferente. Únicamente implica que cada vez que se llame al comando responsable de leer los mensajes se tomen 16 bytes de este buffer. De esta manera, si se juntan

varios mensajes a la vez y se llama al comando de lectura, se tomará únicamente un mensaje, sin cortar ni perder los otros.

- Se activó la segunda UART del MCU (la primera se utiliza para comunicarse con el PC a través del puerto COM virtual), y se configuró con los parámetros necesarios para transmitir mensajes MIDI, es decir, 32.5 kbaudios, con un bit de arranque y uno de parada. También se configuró una de las salidas digitales para que funcionara como salida de la segunda UART.
- Se crearon las funciones necesarias para enviar un mensaje a la segunda UART y que fuera transmitido.
- Se crearon las funciones necesarias para extraer la información MIDI de los paquetes OSC/MIDI recibidos. La información MIDI se encuentra en los dos o tres últimos bytes del paquete OSC/MIDI, y en orden inverso. Se crearon funciones separadas para los mensajes MIDI de 2 bytes y los de 3. Estas funciones toman los últimos bytes del buffer de lectura, los invierten, y los introducen en un buffer con información MIDI. Además, puesto que la información en el buffer de entrada es de tipo “char”, es necesario convertirla a los valores que representan los caracteres antes de enviarla.
- Se modificó la función de respuesta al comando de lectura de los datos de entrada, añadiendo en ella las funciones antes mencionadas. Además de imprimir en el terminal del PC la información contenida en el paquete leído, en caso de que este paquete contenga información OSC/MIDI se extrae esta información y se envía a ambas UARTs, de manera que se imprime por pantalla y también se transmite a la salida digital.

Una vez finalizada esta parte, el MCU es capaz de recibir mensajes de TouchOSC y transmitirlos a un equipo MIDI. Sin embargo, este proceso requiere que el usuario envíe una serie de comandos a través del hiperterminal: en primer lugar, en caso de no estar el MCU ya conectado a una red, es necesario iniciar el proceso de SmartConfig. Tras esto, hace falta abrir un socket UDP, enlazarlo al puerto adecuado, y escuchar la llegada de mensajes. En total, esto supone 3 comandos, más un comando adicional cada vez que se quiera recibir un paquete. La segunda fase del proceso de programación consistió en automatizar este proceso:

- Puesto que el puerto al que debe conectarse el MCU no varía, se modificó el comando de creación de un socket para que lo asocie automáticamente al puerto adecuado (12101).
- Se modificó la rutina *main* del programa para que se produzca la apertura del socket tras inicializar la configuración básica del MCU.
- Se modificó la rutina *main* del programa para que su única labor tras realizar la configuración inicial y la apertura del socket UDP fuera ejecutar la función de lectura de los datos de entrada.

Tras realizar esta segunda serie de modificaciones, el programa recibe y transmite los mensajes de manera autónoma después de ser iniciado. Sin embargo,

puesto que ya no escucha los comandos recibidos desde el hiperterminal del PC, no permite realizar la función SmartConfig, la cual resulta esencial para configurar el dispositivo cuando se desconecta de una red o se necesita cambiar la red a la que está conectado. Para resolver este problema, se modificó el código de inicialización del dispositivo de manera que si se mantiene presionado el botón 1 de la placa de desarrollo mientras se inicia, el dispositivo comienza el proceso de SmartConfig. De esta manera, para conectarlo a una red únicamente es necesario resetear el dispositivo mientras se mantiene presionado el botón 1, y después transmitir desde otro dispositivo los datos de la red a la que se desea conectar el dispositivo.

2.2.7 Diseño del interfaz de salida

Tras programar el MCU ha sido necesario encontrar la manera de conectar la placa de desarrollo a cualquier equipo con el que se desee probar. Para poder comprender las decisiones que se tomaron a la hora de realizar la conexión es necesario examinar el funcionamiento de una conexión física MIDI, que ya se introdujo en la sección 2.1.3.

De acuerdo al estándar MIDI, las conexiones MIDI se realizan mediante un conector DIN de 5 pines. De los 5 pines se emplean 3 en el caso de conectores de salida, y 2 en los de entrada. El circuito que se emplea en una conexión MIDI de entrada se muestra en la Figura 3 (se omite la parte del circuito correspondiente al MIDI-Thru, por ser opcional y no relevante para este proyecto):

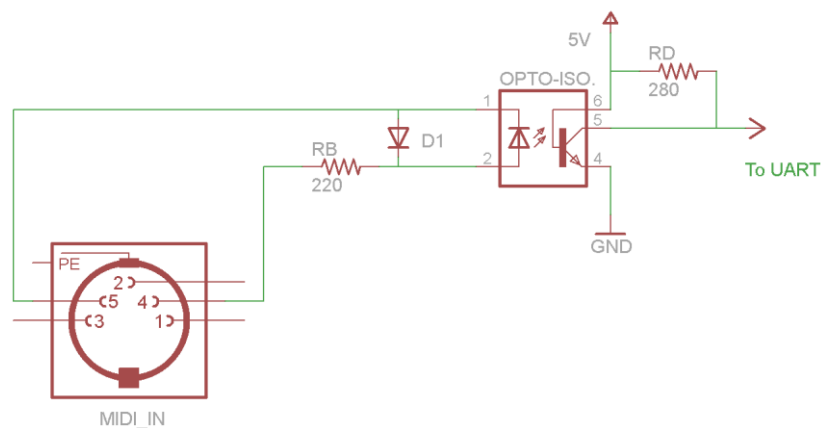


Figura 3 - Circuito de una conexión MIDI de entrada

El circuito que debe tener un conector de salida se muestra en la Figura 4:

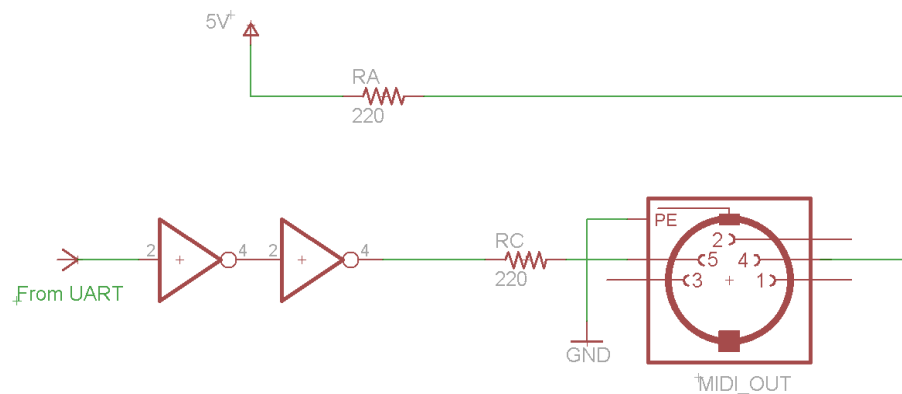


Figura 4 - Circuito de una conexión MIDI de salida

Tras conectar ambos circuitos con un cable DIN de 5 pines, el funcionamiento del circuito resultante es el siguiente:

- El pin 4 está alimentado a 5 voltios de manera constante, mientras que el pin 5 recibe la salida de la UART del emisor.
- Cuando de la UART del emisor sale un 1 lógico (5 V), ambos terminales de los diodos tienen la misma tensión, por lo que éstos se polarizan inversamente. El optoacoplador tiene su salida a nivel alto para una corriente de entrada nula, de manera que la UART del dispositivo receptor recibe un 1 lógico.
- Cuando de la UART del emisor sale un 0 lógico (0 V), el diodo del optoacoplador se polariza directamente. El optoacoplador tiene su salida a nivel bajo cuando la corriente de entrada es la nominal, por lo que la UART del receptor recibe un 0 lógico.
- Los dos negadores concatenados que se muestran en el circuito de salida forman un buffer de corriente, destinado a generar la corriente necesaria para hacer funcionar el circuito en caso de que la UART no sea capaz de generala por sí sola.

Este tipo de circuito se denomina lazo de corriente, ya que la transmisión de información se basa en la corriente en lugar de en la tensión. De acuerdo al estándar MIDI, el circuito funciona con una corriente nominal de 5 mA.

Para conseguir una comunicación satisfactoria entre la placa de desarrollo y un dispositivo receptor con un interfaz MIDI de entrada, es necesario crear un circuito similar al descrito en el interfaz de salida MIDI y conectarlo a la salida de la placa de desarrollo. Sin embargo, a pesar de que la placa posee pines de salida correspondientes a VCC, a tierra, y a la salida de la UART, los valores de tensión tanto de VCC como el 1 lógico de la UART son de 3,8 V, en lugar de los 5 V requeridos.

Se contemplaron varias alternativas para resolver este problema:

- Los 5 V de alimentación se pueden tomar del circuito que alimenta la placa de desarrollo. Aparte de eso, se puede crear un circuito simple para amplificar la señal de salida de la UART hasta que la señal alcance los 5 V requeridos por el estándar.
- Puesto que el circuito es un lazo de corriente y no depende de las tensiones concretas de las señales, se puede generar la corriente necesaria modificando los valores de las resistencias especificadas en el estándar.

La segunda solución fue la primera que se intentó, ya que resulta más sencilla y elimina la necesidad de emplear circuitos activos. El primer paso para implementarla consistió en averiguar si la placa de desarrollo es capaz de generar la corriente necesaria para hacer funcionar el circuito. Tras consultar las especificaciones de la placa, se ha visto que la única restricción respecto a la corriente de salida es que en total, entre todas sus salidas, no supere los 48 mA. Por lo tanto, la corriente que es capaz de suministrar es más que suficiente, lo cual elimina también la necesidad de utilizar el buffer de corriente sugerido por la especificación MIDI.

Antes de cambiar los valores de las resistencias definidas en el estándar, se realiza un cálculo de la corriente que generaría con las resistencias actuales:

$$I = \frac{V_{cc}}{\sum R_i} = \frac{3.8 V}{3 \cdot 220 \Omega} = 5.75 mA$$

A pesar de que este cálculo no incluye la caída de tensión producida en el diodo del optoacoplador, muestra que el circuito podría funcionar sin necesidad de cambiar los valores de las resistencias definidos en el estándar. De esta manera, únicamente sería necesario conectar una resistencia de 220 Ω a los pines de alimentación y de salida de la UART para conseguir un interfaz de salida válido.

Para poner a prueba el resultado, se unieron las resistencias a un conector hembra y se conectaron a la placa de desarrollo mediante cables. Una vez hecho esto, se empleó un interfaz MIDI para poder hacer llegar la salida de la placa de desarrollo al PC y comprobar que la conexión entre la placa y la interfaz es eficaz. Se conectó con un cable DIN la salida de la placa de desarrollo con el interfaz MIDI, y éste al PC mediante un cable USB. Tras instalar los drivers adecuados, se pudo comprobar que el PC recibía el tráfico MIDI enviado por la placa de desarrollo, lo cual mostró que tanto el programa como el interfaz de salida funcionaban correctamente. Como prueba adicional, se conectó la placa de desarrollo al sintetizador General MIDI, y se comprobó que recibía los mensajes correctamente.

Tras realizar las pruebas anteriores, se optó por construir un cable DIN macho que se conecte directamente a los dispositivos receptores. Para ello se utilizó un cable de 2 hilos + tierra y un conector DIN macho. Debido a la cantidad de espacio libre en el conector DIN, fue posible introducir las dos resistencias en su interior y soldarlas

directamente a los pines de salida, de manera que la conexión resultara más compacta, robusta y fiable. El cable resultante puede verse en la Figura 5:

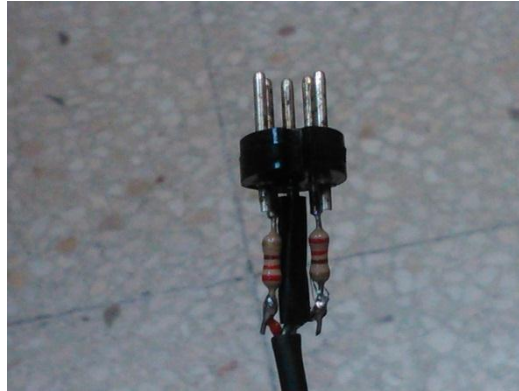


Figura 5 - Detalle de colocación de resistencias en el cable fabricado

Finalmente, se puso a prueba el funcionamiento del cable, conectándolo a los pines de VCC, salida de la UART y tierra de la placa de desarrollo por un extremo, y al sintetizador General MIDI con el conector DIN. Se comprobó que el sintetizador respondía a los mensajes enviados desde TouchOSC, por lo que se dio por finalizado el interfaz de salida.

2.3 Finalización del diseño

Una vez se comprobó que el dispositivo diseñado es capaz de recibir mensajes y comunicarse tal como se necesita, se procedió a realizar varias modificaciones menores para finalizar el proyecto.

En primer lugar, se eliminó todo el código sobrante del programa del dispositivo. Este paso se había dejado para el final, ya que se consideró que el código incluido inicialmente en el programa podría ser necesario posteriormente en caso de que el dispositivo no funcionara como es debido.

Por otra parte, la placa de desarrollo se alimenta a través del cable USB que la conecta al ordenador. Puesto que el objetivo es crear un dispositivo lo más autónomo posible, se decidió sustituir la alimentación desde el PC por un sistema con pilas AAA. Éste se conectó a la entrada para alimentación externa de 5 V de que dispone la placa de desarrollo. Puesto que el rango de tensiones permitido por la placa es relativamente amplio, se optó por emplear 3 pilas AAA, que producen una tensión conjunta de 4,5 V, la cual se comprobó que es suficiente para hacer funcionar la placa.

Tras comprobar el funcionamiento de la placa con las pilas, se añadió un interruptor simple entre las pilas y la placa, y se cableó todo el diseño. El circuito de alimentación final aparece representado en la Figura 6:

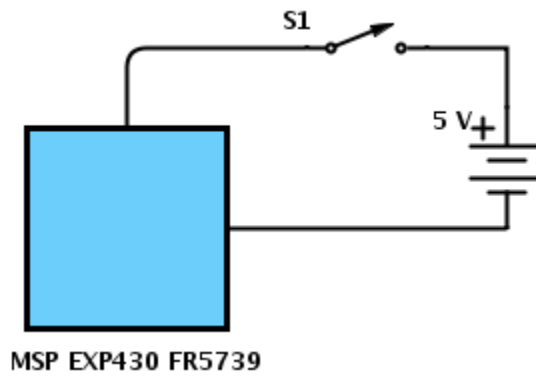


Figura 6 - Esquema de alimentación de la placa de desarrollo

Por último, para garantizar la mejor comunicación posible entre los dispositivos conectados a la red, se ha realizado un análisis del estado de ocupación de los canales de transmisión de las redes Wi-Fi circundantes, que permitió elegir el canal menos ocupado para la red Wi-Fi dedicada que se estaba empleando para el proyecto. Este análisis se realizó mediante el software “inSSIDer Office”.

Tras realizar estas modificaciones, el dispositivo se encuentra en su estado óptimo de funcionamiento (para los propósitos de este proyecto), y se dio por terminada la parte del proyecto relacionada con el desarrollo del dispositivo.

3. Resultados y conclusiones

3.1 Resultados

Una vez finalizado el diseño del dispositivo, se han realizado pruebas conectándolo directamente al sintetizador General MIDI y enviando diversos mensajes desde TouchOSC. La red en la que se hicieron las pruebas contenía únicamente el teléfono móvil sobre el que se ejecutaba TouchOSC, y el dispositivo de recepción, por lo que el router no tenía que manejar tráfico procedente de otras fuentes.

Los resultados de las pruebas muestran que la comunicación entre TouchOSC y el dispositivo se establece adecuadamente, al igual que la comunicación entre el dispositivo y el sintetizador, y este responde correctamente a los mensajes recibidos. Se ha conseguido reproducir notas musicales en varios canales, modificar el programa de un canal, modificar el volumen de un canal, y utilizar la función *all notes off*. En general, el dispositivo está programado para transmitir todos los mensajes MIDI de 3 bytes, además de mensajes de 2 bytes de cambio de programa y de cambio de presión del canal.

El principal problema del dispositivo es la latencia entre la activación de un elemento en TouchOSC y la activación del sintetizador en respuesta. Esta latencia puede llegar a ser de 1 ó 2 segundos, lo cual vuelve el uso del dispositivo completamente inefectivo. En condiciones adecuadas, sin embargo, la latencia es de únicamente unos milisegundos, volviéndose imperceptible.

Antes de utilizar el software inSSIDer Office para verificar el estado de ocupación de las frecuencias Wi-Fi, la latencia se producía en la mayoría de los mensajes. Tras cambiar el canal al menos ocupado, sin embargo, la latencia desapareció prácticamente por completo. Estos hechos indican que con toda probabilidad la latencia se debe a la saturación de las frecuencias de Wi-Fi, lo cual dificulta y retarda la llegada de los paquetes al router y al dispositivo receptor. Parte de la latencia puede tener que ver con el tiempo que tarda TouchOSC en transmitir los mensajes desde que se pulsa un elemento de su interfaz.

Las pruebas iniciales se realizaron en un entorno con muchas redes Wi-Fi funcionando simultáneamente y con relaciones señal/ruido (S/N) próximas a 0 ó negativas, por lo que se ha realizado una segunda serie de pruebas en otro laboratorio con condiciones más benignas.

El objetivo de la segunda tanda de pruebas era medir la latencia percibida en distintas condiciones de relación S/N (se entiende como ruido N a la potencia de otras redes Wi-Fi que transmiten en el mismo canal que el sistema). El emisor y el receptor se encontraban próximos el uno al otro, y la variación de la relación S/N se conseguía alejando ambos del punto de acceso. El receptor se encontraba conectado al interfaz USB, que transmitía los mensajes a un PC para poder visualizarlos. En la Figura 7 se muestra un diagrama del montaje empleado para las pruebas:

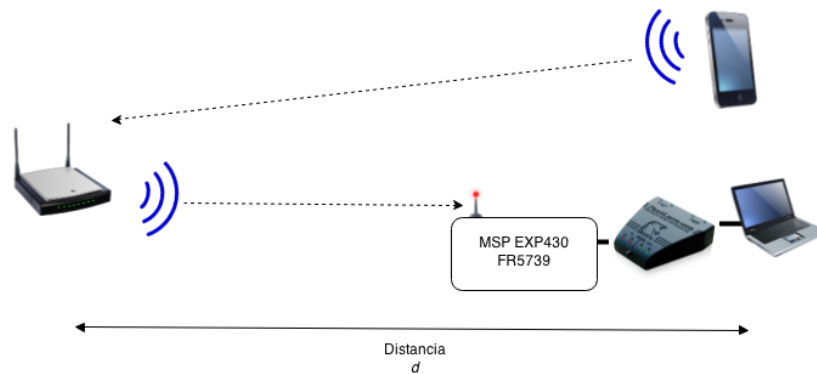


Figura 7 - Esquema del montaje empleado para las pruebas

La prueba se realizó en un laboratorio con línea de visión directa entre el transmisor, el router y el receptor. Las medidas de potencia de señal emplean la métrica RSSI, que no tiene una relación directa con potencias en mw ó dBm. Esta es la métrica más utilizada en medidas de potencia en redes Wi-Fi. Se realizaron medidas con distancias de entre 1 y 30 m, sin llegar a experimentar ninguna latencia en la transmisión ni pérdida alguna de mensajes. Las potencias de las diversas redes se midió en el PC mediante el software Acrylic Wi-Fi. En la gráfica de la Figura 8 puede verse la relación entre el RSSI de la señal de nuestro router en comparación con el resto de redes del mismo canal.

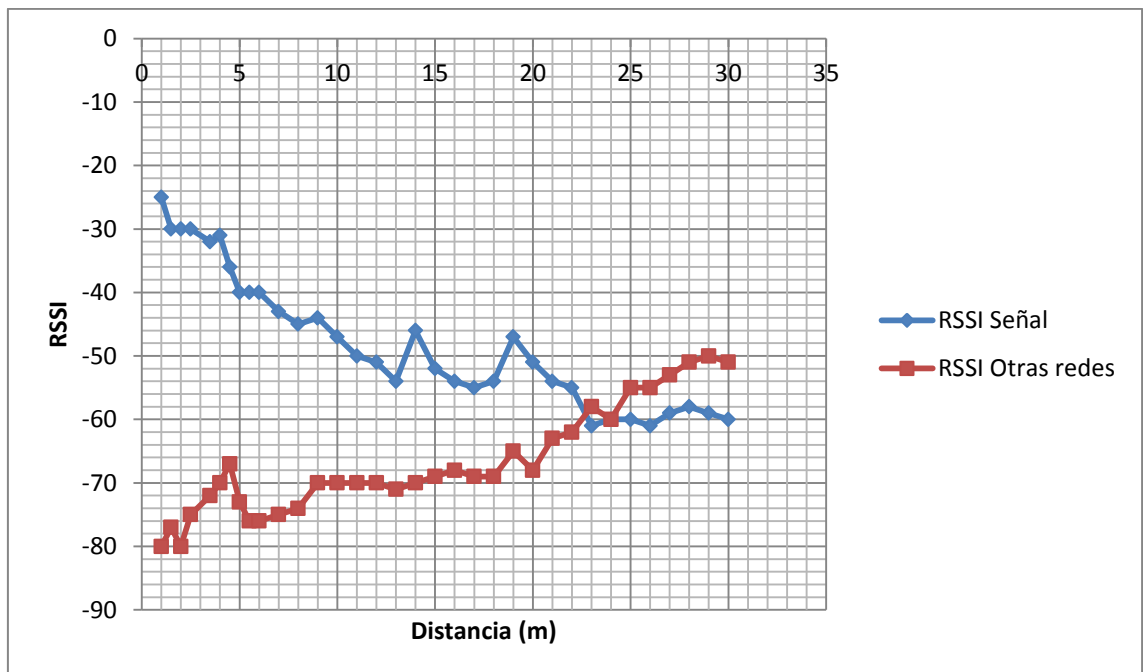


Figura 8 - Resultado de las pruebas

La ausencia de latencias muestra un comportamiento adecuado del dispositivo, aunque no se puede conocer la carga que las redes ajenas experimentaron durante esta

prueba, por lo que este comportamiento podría no ser el característico para estas condiciones. Sin embargo, demuestra que el dispositivo es plenamente utilizable en redes dedicadas y entornos con pocas interferencias de otras redes.

3.2 Conclusiones

En cuanto al cumplimiento de los objetivos los resultados son los siguientes:

- Se ha conseguido diseñado un dispositivo que reciba mensajes MIDI encapsulados en OSC a través de la red Wi-Fi y transmitirlos a un equipo receptor.
- El dispositivo es autónomo en gran medida, y una vez conectado a una red puede funcionar sin intervención de un operario. Además el proceso de conexión a una red se facilita gracias a que el usuario no sebe configurar manualmente los datos de la red, sino que se transfieren automáticamente al dispositivo mediante una aplicación móvil. Además, el dispositivo puede alimentarse con pilas, lo cual aumenta su autonomía.
Sin embargo, no hay ninguna manera simple de averiguar la dirección IP del dispositivo una vez se ha conectado a una red, más allá de comprobarla en la lista de hosts del router. Esto dificultaría el uso de varios de estos dispositivos simultáneamente
- La transmisión es fiable cuando se emplea el dispositivo en una red dedicada y en un entorno libre de interferencias con otras redes. Las pérdidas y latencias comienzan a producirse a medida que se incrementan las interferencias con redes ajenas.
- El dispositivo tiene una anchura de 50 mm, una longitud de 80 mm y una altura de unos 40 mm. El tamaño no es excesivamente compacto, pero el exceso se debe a que se ha utilizado una placa de desarrollo. En un modelo de producción reduciría en mucho su tamaño, al emplear placas impresas mucho más pequeñas.

En conclusión, se han satisfecho en general los objetivos del proyecto, y se ha conseguido fabricar un dispositivo que cumple su función.

3.3 Mejoras y extensiones

El simple prototipo diseñado para este proyecto podría verse mejorado de varias formas:

- Implementación de comunicaciones bidireccionales, que permitan al dispositivo funcionar como emisor y receptor simultáneamente.
- Diseño de un modelo en un circuito impreso, que emplee los chips directamente en lugar de una placa de desarrollo, reduciendo

considerablemente el tamaño del diseño. La laca podría entonces introducirse en una caja de metal o plástico para incrementar su robustez.

- Sustitución de la alimentación por pilas por una batería recargable interna, conectable a un soporte. Esto incrementaría la comodidad de uso, y supondría a largo plazo un ahorro en pilas.

4. Bibliografía

4.1 Documentos

The MIDI Manufacturers Association (1995): '*MIDI 1.0 Detailed Specification*', document version 4.2

Texas Instruments (2011), '*MSP430FR57xx Family: User's Guide*'

---- (2014), '*MSP430 Hardware Tools: User's Guide*'

---- (2014), '*MSP430FR573x Mixed-Signal Microcontrollers*'

---- (2012), '*MSP-EXP430FR5739 FRAM Experimenter Board: User's Guide*'

---- (2012), '*TI SimpleLink™ CC3000 Module - Wi-Fi 802.11b/g Network Processor*'

---- (2012), '*TI CC3000 BoosterPack Evaluation Board: User's Guide*'

4.2 Páginas web

OSC:

<http://opensoundcontrol.org/>

http://en.wikipedia.org/wiki/Open_Sound_Control

MIDI:

<http://www.midi.org/techspecs/midimessages.php>

<http://en.wikipedia.org/wiki/MIDI>

<http://www.musicradar.com/news/tech/30-years-of-midi-a-brief-history-568009/>

TouchOSC:

<http://hexler.net/software/touchosc>

Texas Instruments:

http://processors.wiki.ti.com/index.php/CC3000_Wi-Fi_MSP430_FRAM

[http://processors.wiki.ti.com/index.php/CC3000 First Time Getting Started Guide](http://processors.wiki.ti.com/index.php/CC3000_First_Time_Getting_Started_Guide)

[http://processors.wiki.ti.com/index.php/CC3000 Basic Wi-Fi example application for MSP430](http://processors.wiki.ti.com/index.php/CC3000_Basic_Wi-Fi_example_application_for_MSP430)

4.3 Procedencia de las imágenes, por orden de aparición:

TouchOSC: imágenes propias

Esquema de sistema: imagen propia

Hardware de interfaz MIDI: imágenes propias

Cable DIN: imagen propia

Esquema de alimentación: imagen propia

Esquema de prueba: imagen propia

Resultado de prueba: imagen propia

MSP-EXP430FR5739 (1):

<http://www.rapidonline.com/catalogueimages/Product/S73-4716P02WL.jpg>

MSP-EXP430FR5739 (2):

<http://contrans.pl/images/articles/MSP-EXP430FR5739.jpg>

Módulo CC3000:

[http://processors.wiki.ti.com/images/5/59/MuRata module.jpg](http://processors.wiki.ti.com/images/5/59/MuRata_module.jpg)

Interfaz MIDI Motu FastLane-USB:

<http://cdn-data.motu.com/site/images/fastlane/web1-5/fastlane-large-iso.jpg>

Sintetizador Yamaha TG-300:

<http://www.wikizic.org/1-044-044837-Yamaha-TG-300.jpg>

Anexo A: Manual de usuario

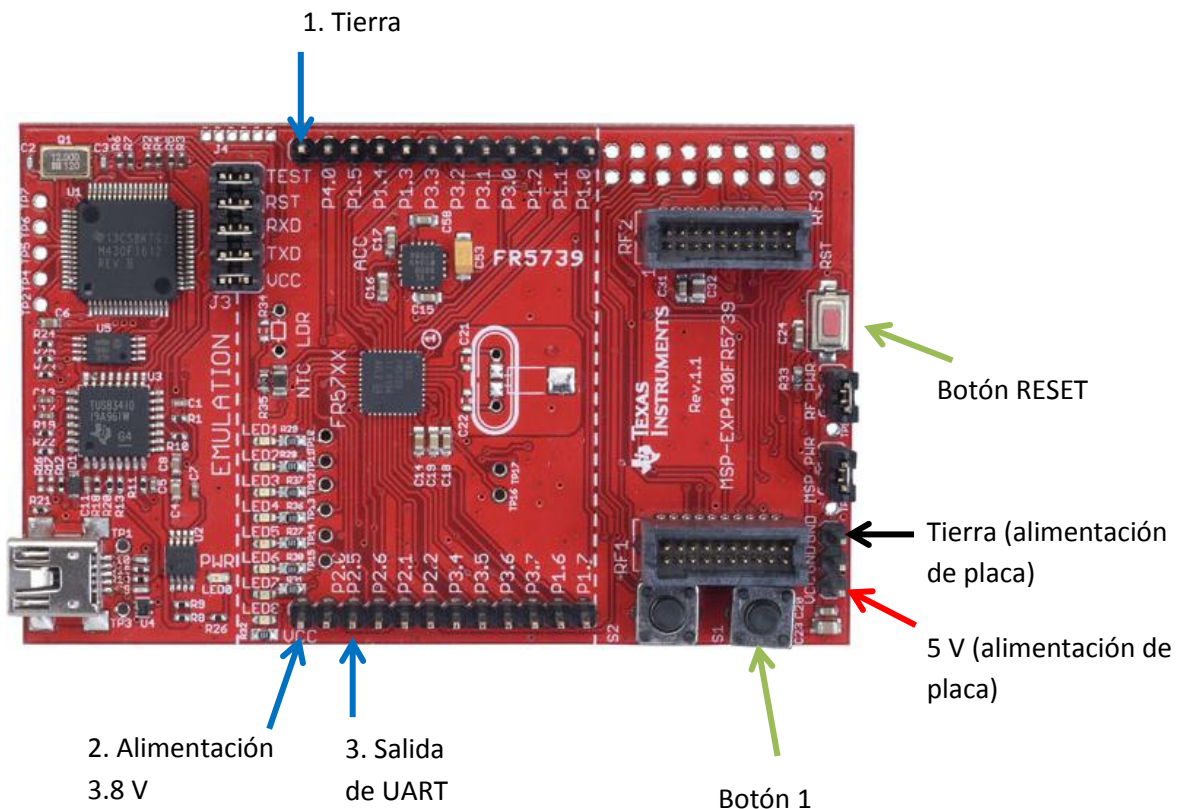
El dispositivo que se ha diseñado es sencillo de utilizar, y no permite una gran interacción con el usuario. A continuación se muestra la manera de alimentarlo, conectarlo a un dispositivo receptor, y conectarlo a una red Wi-Fi.

Alimentación

La alimentación de la placa se obtiene de 3 pilas AAA colocadas en un porta pilas. Para alimentarla es necesario conectar los cables de salida a la entrada de alimentación externa de la placa, consistente en tres pines macho situados en un extremo de ésta. Para facilitar la conexión, se han soldado los cables necesarios a tres pines hembra, por lo que únicamente es necesario conectarlos orientándolos correctamente. Una vez conectados, se puede alimentar la placa accionando el interruptor incluido en el circuito de alimentación.

Conexión a un equipo MIDI receptor

Para conectar el dispositivo a un equipo MIDI, se ha fabricado un cable que se conecta al puerto MIDI in del equipo por un lado, y por el otro a tres pines de salida de la placa de desarrollo. Debido a la separación de estos pines, se recomienda utilizar cables con pinzas cocodrilo para realizar la conexión.



Las conexiones al cable se realizan como sigue:

Cobertura del cable	Función	Conexión
Sin cobertura	Tierra	1
Blanco	Alimentación 3.8 V	2
Negro	Señal procedente de UART	3

Conexión a una red Wi-Fi

El dispositivo receptor únicamente puede conectarse a redes Wi-Fi sin clave de acceso.

Para conectar el dispositivo a una red Wi-Fi, es necesario primero activar el modo SmartConfig. Para ello es necesario mantener pulsado el botón 1 de la placa y pulsar el botón de RESET. Seguir manteniendo pulsado el botón 1 hasta que se encienda el LED5. El LED6 debería comenzar a parpadear, indicando que el dispositivo está esperando para recibir los datos de la red a la que se debe conectar.

Para pasarle los datos de la red, es necesario conectar un dispositivo móvil a la red a la que se desea conectar el dispositivo receptor y abrir la aplicación SmartConfig proporcionada por TI. Los datos de la red deberían aparecer automáticamente. Pulsar el botón START para comenzar a emitir los datos hacia el dispositivo receptor. Cuando los recibe, el LED6 deja de parpadear y se mantiene encendido, tras lo cual se encienden también el LED7 y el LED8. Esto indica que el dispositivo se ha conectado a la red y pasa al modo de escucha.

Si el dispositivo se desconecta y se vuelve a conectar, o si se pulsa el botón RESET, intentará conectarse automáticamente a la red a la que estaba conectado antes. Si no lo consigue, se encenderá únicamente el LED5, en cuyo caso será necesario repetir el procedimiento de SmartConfig.

Configuración de TouchOSC

Para poder comunicarse adecuadamente con el dispositivo receptor, es necesario introducir la dirección IP del receptor en la configuración de MIDIBridge en TouchOSC.

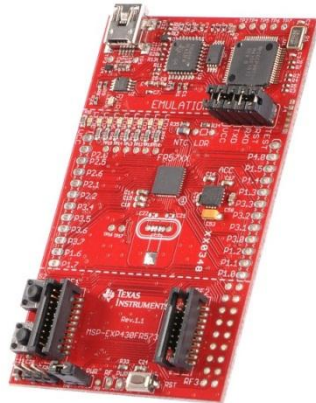
La manera más simple de averiguar la dirección IP del dispositivo receptor es examinar los host conectados al router Wi-Fi, y comprobar cuál de ellos coincide con la dirección MAC de la tarjeta de red CC3000. La dirección MAC de la tarjeta de red puede verse impresa en el chip CC3000 de la placa correspondiente.

Anexo B: Hardware y software empleado

En este anexo se pueden consultar las características más destacables del hardware y software empleado en lo relativo al desarrollo de este proyecto.

Hardware

Placa de desarrollo MSP-EXP430FR5739



- Microcontrolador MSP430 FR5739
 - o Reloj de 24 MHz
 - o Tensiones de alimentación de 2 V a 3.6 V
 - o Dos UARTs con múltiples modos de funcionamiento
- Debugging y alimentación por USB, comunicación entre UART y PC
- Dos botones y un interruptor de reset
- 8 LEDs
- Permite conectividad con tarjetas de radio CCxxxx

Módulo Wi-Fi CC3000



- Características de transmisión
 - o Potencia de TX de 18 dBm a 11Mbps
 - o Sensibilidad de RX de -88dBm a 11Mbps
- Permite conectividad con la placa de desarrollo MSP-EXP430FR5739

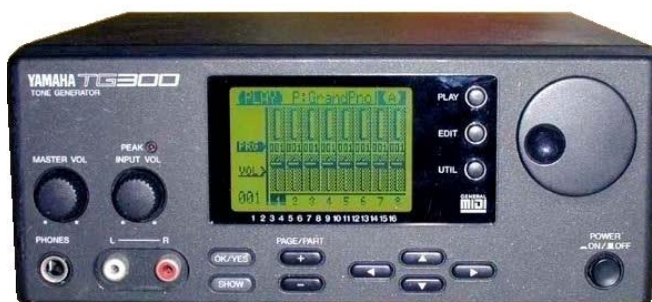
Interfaz MIDI Motu FastLane-USB



- Dos puertos MIDI de entrada
- Dos puertos MIDI de salida
- Conexión USB
- MIDI-Thru activable

Se ha utilizado para comprobar junto con el software Bome's SendSX para comprobar que se generan correctamente mensajes MIDI.

Sintetizador Yamaha TG-300



- Conformación al estándar General MIDI

Es el equipo sobre el que se ha probado el funcionamiento del dispositivo receptor diseñado.

Software

TouchOSC

Software empleado transmitir mensajes OSC a través de una red Wi-Fi desde interfaces gráficas personalizables. Tiene capacidad para transmitir mensajes OSC y MIDI encapsulado en OSC. Es el programa principal sobre el que se trabaja

Code Composer Studio 5

Entorno de desarrollo integrado de Texas Instruments. Se ha utilizado para programar el MCU.

Bome's SendSX

Permite capturar tráfico MIDI por un puerto de entrada, y enviar mensajes MIDI por un puerto de salida. Se ha empleado para comprobar el funcionamiento de las comunicaciones en diversas pruebas.

Wireshark

Analizador de protocolos de red que permite capturar los paquetes que entran en el PC. Se ha utilizado para examinar el tráfico generado por TouchOSC.

Putty

Hiperterminal que permite comunicarse con puertos serie reales o virtuales. Se ha utilizado para comunicarse con el MCU desde el PC. Durante el desarrollo del proyecto se sustituido por el software Advanced Serial Port Monitor.

Advanced Serial Port Monitor

Al igual que Putty, se ha utilizado para comunicarse con el MCU desde el PC.

inSSIDer Office

Software de análisis del estado de la red Wi-Fi. Se ha utilizado para buscar el canal Wi-Fi menos ocupado para mejorar las comunicaciones. Durante el desarrollo del proyecto se ha sustituido por el software Acrylic Wi-fi

Acrylic Wi-Fi

Software de análisis de red similar a inSSIDer Office, pero gratuito.

Anexo C: Composición de mensajes OSC y MIDI

Los mensajes OSC están estructurados de manera que su tamaño en bytes es siempre múltiplo de 4. Los mensajes OSC se componen de tres partes:

- Dirección OSC. Secuencia de caracteres que comienza con el carácter “/” y representa el destino del paquete OSC (por ejemplo, una función determinada). El equipo receptor es el encargado de hacer llegar los argumentos del mensaje OSC a su destino.
- String de tipos de argumento. Secuencia de caracteres que comienza con el carácter “;”, y va seguida de caracteres que representan el tipo de los argumentos que siguen, y en el mismo orden en que se envían los argumentos.
- Argumentos. Secuencia que contiene los argumentos del mensaje OSC, en el mismo orden en que aparecen en el string de tipos de argumento.

Cada uno de estos campos tiene una longitud múltiplo de 4 bytes, por lo que se puede rellenar con 0-3 ceros al final para cuadrar sus longitudes.

Un ejemplo del tipo de mensaje OSC con el que se trabajará es:

ASCII	/	m	i	d	i				,	m						
Hex	2F	6D	69	64	69	00	00	00	2C	6D	00	00	00	7F	48	90
	Dirección OSC								Tipos de argumento. m = MIDI			Mensaje MIDI en orden inverso				

En cuanto a los tipos de mensajes MIDI, el dispositivo receptor está diseñado para poder responder a los siguientes:

- 8n kk vv: Mensajes Note Off. Desactivan una nota.
- 9n kk vv: Mensajes Note On. Activan una nota.
- An kk vv: Mensajes de pulsación polifónica
- Bn cc vv: Mensajes de cambio de control. Permiten modificar la configuración de un canal.
- Cn pp: Mensajes de cambio de programa. Sirven para cambiar de instrumento en un sintetizador.
- Dn vv: Mensajes de presión de canal.
- En ll mm: Mensajes de pitch bend.

En las descripciones anteriores, n representa el canal, k la tecla (nota), v la velocidad, c el parámetro de control, p el programa, l los bits menos significativos del pitch bend, y m los bits más significativos.

Anexo D: Componentes software del sistema y pseudocódigo del programa

En este anexo se listarán los diversos componentes software que se requieren tanto para poder hacer funcionar el dispositivo como para continuar su desarrollo, así como dónde conseguirlos. Además de eso, se proporciona pseudocódigo de alto nivel correspondiente al programa que se ejecuta en la placa de desarrollo.

Componentes software

TouchOSC: disponible en Google Play Store para dispositivos Android, o en el App Store para dispositivos iOS

Code Composer Studio: disponible a través de la página web oficial de Texas Instruments:

<http://www.ti.com/tool/ccstudio>

Bome's SendSX: disponible a través de la página web de Bome:

<http://www.bome.com/products/sendsx>

Selección de pseudocódigo

- Método *main*:

```
Reiniciar Flags;
Desactivar Watchdog_timer;

inicializarDriver();

si(boton1 pulsado) {
    comenzarSmartConfig();
}

abrirSocket();

Inicializar buffer de Rx de UART;

escucharMIDI();
```

- Inicialización de driver:

```
inicializarIO();
inicializarSPI();
configurarUART();
inicializarWLAN();
```

```

encenderLED(5);
Enmascarar eventos innecesarios de CC3000;

Generar y enviar un string de versión por la UART1;
Reiniciar flag de smartConfig;

```

- Apertura de socket UDP:

```

(opcional) Seleccionar valor de timeout
direccionSocket;

mientras (flagCC3000DHCP ó flagCC3000Conectado) {
    esperar;
}

crearNuevoSocket(AF_INET[dominio], SOCK_DGRAM[tipo],
    UDP[protocolo]);

(opcional) Configurar timeout de socket;

Configurar tipo de socket;
Configurar puerto de socket; //puerto 0x2F45 = 12101

Aplicar configuración al socket;
Asociar puerto al socket;

```

- Escuchar MIDI

```

Tomar datos del socket;
si(no se han recibido datos) {
    enviarUART(stringNoDatos);
} si no {
    char2ASCII(bufSalida[ASCIIBuf], bufEntrada[RxBuf],
    tamañoBufEntrada);
    enviarUART(nuevaLinea);
    enviarUART(ASCIIBuf);
    enviarUART(nuevaLinea);
    si(mensaje MIDI){
        si(mensaje comienza con C ó D) { //mensaje de 2 bytes
            MIDI2(bufSalida[MIDIBuf2], bufEntrada[RxBuf],
            tamañoBufEntrada[16]);
            char2ASCII(bufSalida[ASCII_MIDIBuf],
            bufEntrada[MIDIBuf], tamañoBufEntrada[4]);
            enviarUART(ASCII_MIDIBuf);
            enviarUART_MIDI(MIDI_BUF);
            enviarUART(nuevaLinea);
        } si no { //mensaje de 3 bytes
            MIDI3(bufSalida[MIDIBuf3], bufEntrada[RxBuf],
            tamañoBufEntrada[16]);
            char2ASCII(bufSalida[ASCII_MIDIBuf],
            bufEntrada[MIDIBuf], tamañoBufEntrada[6]);
            enviarUART(ASCII_MIDIBuf);
            enviarUART_MIDI(MIDI_BUF);
            enviarUART(nuevaLinea);
        }
    }
}

```

```
}  
}
```