

## *Protocolo de medidas en cámara anecoica*

Para llevar a cabo las grabaciones en la cámara anecoica de forma óptima, se ha creado en MATLAB un conjunto de *scripts* en los cuales se mandan los parámetros ya fijados de la comunicación junto con los comandos de inicio y detención de la grabación. Además, el equipo de Teoría de la Señal nos ha proporcionado el correspondiente código para generar de forma automática los audios de la base de datos a grabar (con silencios, claqueta, etc..) y posteriormente poder desglosar los audios grabados separándolos cada vez que se detecte una claqueta.

PARTE 1: Generación de audio para reproducir → ***generar\_audio.m***

PARTE 2: Configuración de los parámetros de JLink para el envío alámbrico del audio desde el micro al ordenador → ***Interactuar\_RTT.m***

PARTE 3: Configuración de los parámetros del puerto serie UART para el envío de los comandos de inicio y finalización de las grabaciones → ***Interactuar\_Termite.m***

PARTE 4: Reproducción del audio generado para su grabación → ***reproducir\_audio.m***

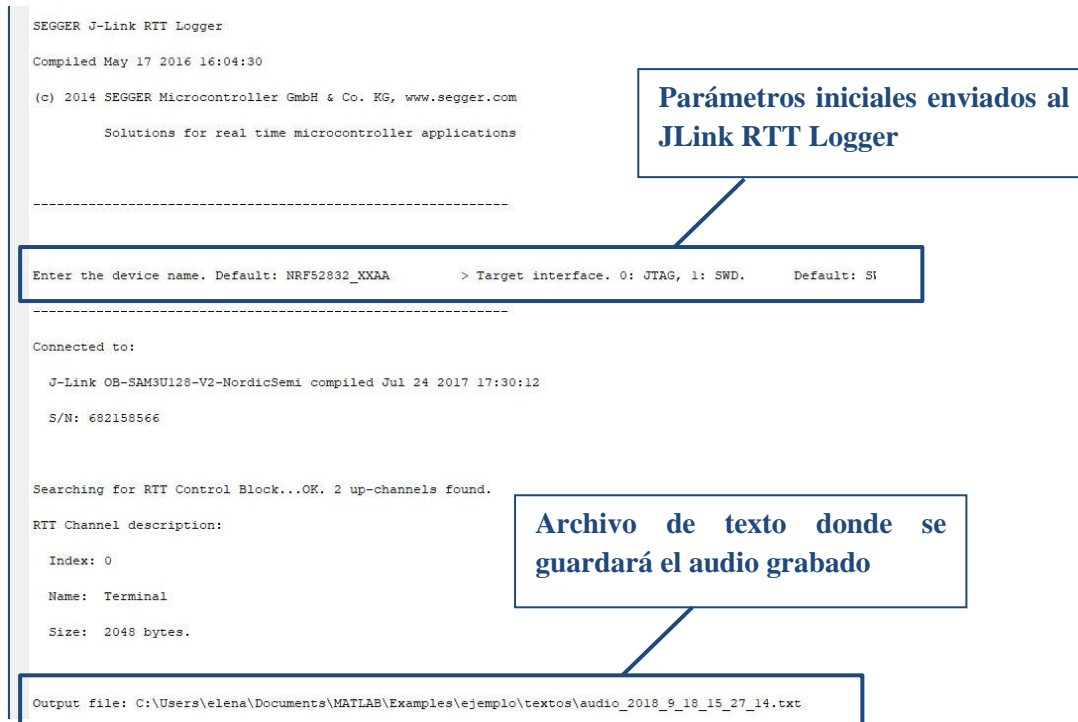
PARTE 5: Extraer los diferentes audios a partir del fichero de texto que genera el micro → ***dividir\_audio.m***

Los *scripts* de la PARTE 1 y de la PARTE 5 son los realizados por el equipo de Teoría de la Señal. La funcionalidad del resto (PARTE 2, PARTE 3 y PARTE 4) se explica a continuación.

Además, todas estas partes se hayan englobadas en un *script* general ***grabar\_audio.m*** desde el cual se van llamando a las diversas partes e incluye el correspondiente código en orden para realizar todo el proceso de grabación.

- **PARTE 2**

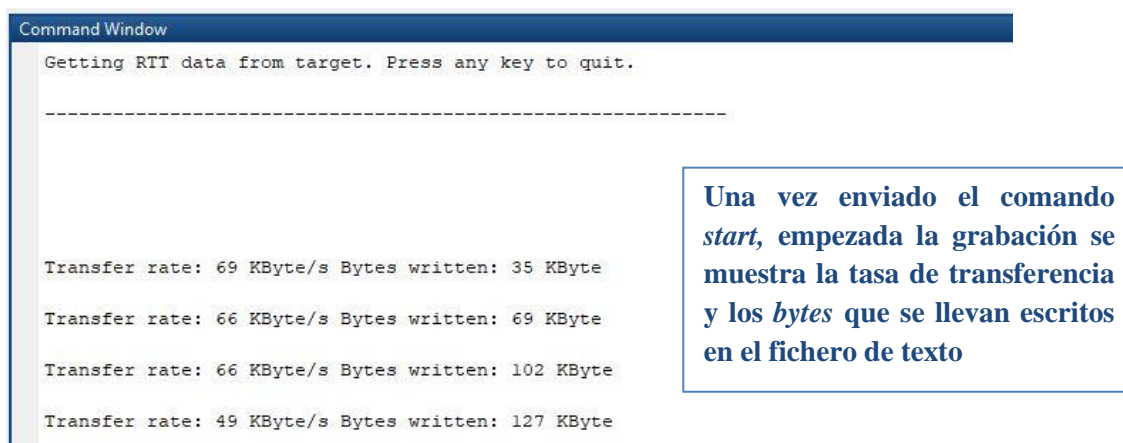
El *script* “Interactuar\_RTT.m” permite leer y mandar una cadena de texto de entrada a una aplicación de consola como es el ejecutable “JLinkRTTLogger.exe”, el cual se abre al ejecutar el código. En él se mandan tantas cadenas de texto como parámetros iniciales solicita JLink para configurar la comunicación entre el micro y el ordenador para el envío del audio. En él también se crea el archivo de texto donde se guardarán las muestras de audio que se graben, en la ruta: \ejemplo\textos\



Nota 1: es conveniente poder nombrar a los audios que se vayan grabando y almacenando en la ruta mencionada de forma diferente para no mezclarlos, de modo que se guardarán según la fecha y hora de su grabación, por ejemplo: audio\_2018\_9\_15\_11\_58\_21.txt (audio\_AA\_MM\_DD\_hh\_mm\_ss.txt).

Nota 2: para poder obtener la fecha y hora cada vez que se graba y usarla para nombrar a los sucesivos audios, se crea el *script* obtener\_fecha.m que está contenido en Interactuar\_RTT.m y que almacena en una variable el *string* de la fecha para luego pasarlo como uno de los parámetros que solicita el JLink inicialmente, concretamente, como el nombre y extensión del archivo donde se desea guardar el audio que se grabe.

Nota 3: cuando se envíen los comandos *start* y *stop*, JLink RTT Logger seguirá ejecutándose mostrando por el *command window* de MATLAB los *bytes* que se están enviando de audio, almacenándose en el archivo de texto, y la velocidad de la transferencia (*kilobytes/s*).



```
Command Window

Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte
Transfer rate: 0 KByte/s Bytes written: 1.349 MByte

SEGGER J-Link RTT Logger is shutting down...
```

Una vez enviado el comando *stop*, detenida la grabación la tasa de referencia pasa a 0 y se muestran los bytes escritos

- PARTE 3

El *script* “Interactuar\_Termite.m” hace posible el envío de datos a un puerto COM asignando un puerto serie del ordenador a un objeto declarado en MATLAB y de esta forma configura los parámetros de la comunicación tales como el *baudrate*, la paridad, etc... y envía automáticamente como cadena de texto el *start* y *stop* para iniciar y finalizar la grabación.

```
Command Window

Termite

'serial' 'Serial-COM4' 'COM4' [115200] 'none' [8] [1]

Command sent to Serial Port is:
start
```


Nota: el código correspondiente al envío del comando *stop* debe ir implementado posterior a la reproducción del audio que se desea grabar puesto que, si se enviase el *stop* antes de empezar a reproducir el audio, no se grabaría nada. Por ello, en el *script* que engloba todas las partes, *grabar\_audio.m*, se ha tenido que partir el código de *Interactuar\_Termite.m*, de modo que se ha implementado primero la parte de configurar los parámetros del puerto serie y el envío del *start*, entre medias el código correspondiente a *reproducir\_audio.m*, que se explica a continuación, y después el envío del comando *stop*.

```
Command Window

Command sent to Serial Port is:
stop
```

- PARTE 4

En el *script* “*reproducir\_audio.m*”, se selecciona el archivo de audio creado en la PARTE 1 (*generar\_audio.m*), que se guarda en la ruta `\ejemplo\audios\` y el cual cuenta con audios de la base de datos, tiempos de silencio al inicio y final y claqueta entre ellos; y se reproduce completo desde MATLAB.



```
Command Window

filepath =

    0×0 empty char array

name =

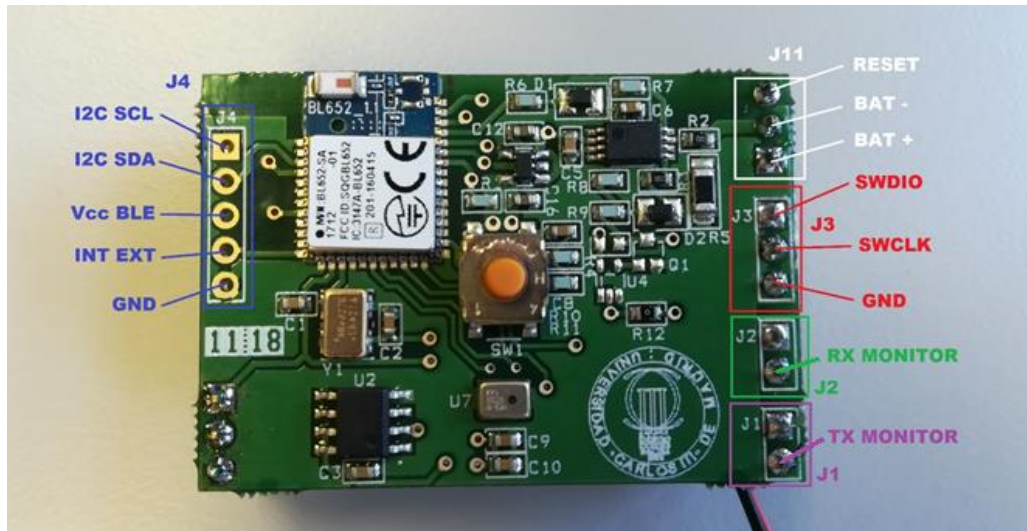
    'total_corto_nuevo'

ext =

    '.wav'
```

Nota: el código referido a esta parte debe implementarse en `grabar_audio.m` entre el envío del comando *start* y el envío del comando *stop* (`Interactuar_Termite.m`), para que la secuencia sea la siguiente: empezar a grabar-reproducción del audio a grabar-finalizar la grabación.

## Conectores del colgante



- Alimentación: si la batería del colgante tuviera la suficiente carga podría utilizarse accionando los dos interruptores que dispone y por tanto, no haría falta alimentarla con la placa de desarrollo. Por otra parte, si se alimenta con la placa de desarrollo, habría que conectar el pin Vcc BLE del jumper 4 (J4) del colgante con un pin Vdd de la placa.
- Programación y comunicación mediante interfaz de depuración: para poder cargar el respectivo programa al colgante a través de la placa de desarrollo, deben conectarse los pines del jumper 3 (J3): SWDIO, SWCLK y GND a sus respectivos de la placa de desarrollo. Debido a que la comunicación se realiza a través de dos vías, por una parte el envío de los comandos se realiza a través de la interfaz UART del micro, y por otra parte el envío del audio grabado se realiza a través de la interfaz de depuración (interfaz utilizada para programar el colgante); **es necesario realizar estas conexiones (SWDIO, SWCLK y GND) aun ya habiendo cargado el correspondiente programa a la placa del colgante.**
- Comunicación serie mediante puerto UART: para poder comunicarse a través del puerto serie UART, hay que conectar el pin RX del jumper 2 (J2) con el correspondiente pin RX (pin 08) de la placa y el pin TX del jumper 1 (J1) con el TX (pin 06) de la placa.

