

I2S (Medium?)

I2S: Driver para k64f

Esto en teoría está hecho, pero con cosas no tan felices. ¿Qué quiero decir con esto?

Para poder arrancar con esta parte lo que hice fue leer **Micros TPF Info/I2S/Audio Output kinetis.pdf** donde detalla cómo encarar audio con la FRDM K64F. Ahora, mientras escribo esto pienso que puede que sea un overkill para nuestro proyecto, pero en un momento sugiere utilizar DMA y Ping Pong Buffers y en esto me basé.

En **Micros TPF Info/I2S/i2s application.pdf** detalla cómo configurar DMA para: 1) hacer ping pong buffer 2) que los datos queden en el buffer como R, L, R, L ... (right, left ...) ya que ese es el formato de los datos que se transmiten por I2S, que con un cable llamado WORDSELECT/ LEFTRIGHTCLK, selecciona a qué canal le corresponden las muestras en ese momento.

Es decir, generar en un buffer algo de este estilo:

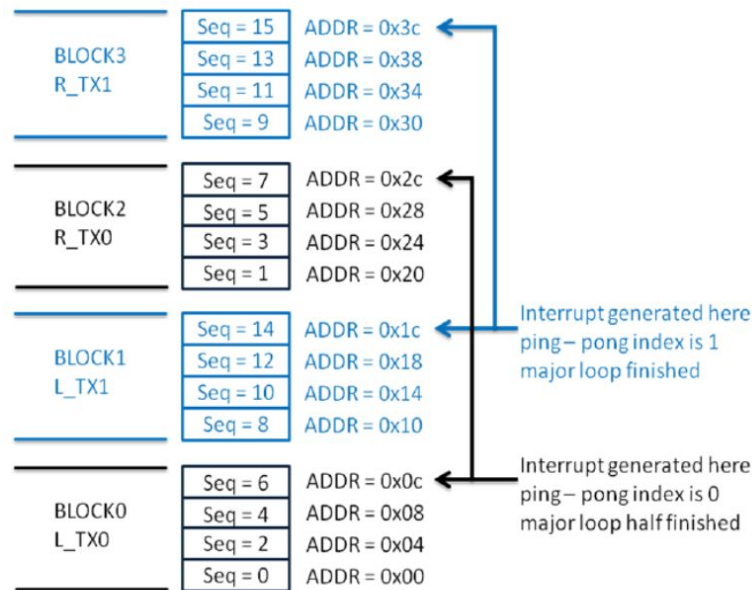
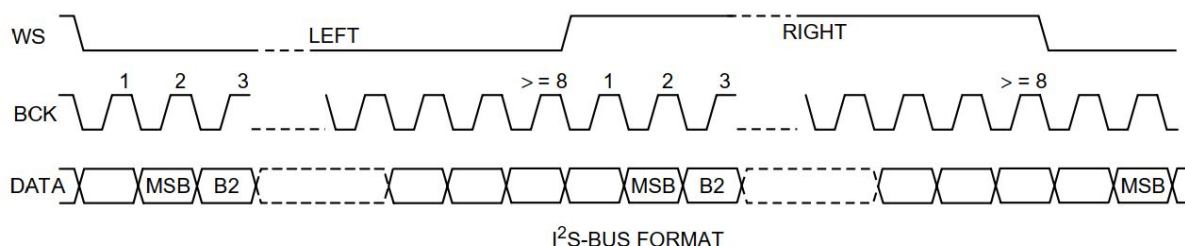


Figure 5. Interrupt when loop is finished and half finished

Esto es un ejemplo de las señales de I2S:



Al seguir prácticamente al pie de la letra la **i2s application.pdf** choque con que no andaba bien la configuración del DMA, es decir crasheaba, algo del estilo segmentation fault. Tiene

un juego de punteros interesante que cuando vi que (despues de reiterados intentos) no podia hacer andar dije: ok, no anda dma vamos a ver si puedo hacer un ping pong buffer pero con buffer circular es decir, cuando llene la primera mitad, proceso la primera mitad, cuando llene la segunda, proceso la segunda, etc. (sin tener en cuenta lo de L-R-L-R).

En mis recuerdos este último approach funcionaba (literal hay un commit que dice que anda), aunque no recuerdo como testearlo sinceramente.

Está **pendiente** revisarlo a fondo y en el peor de los casos, si no se entiende mucho hacerlo de nuevo pero esta vez con el SDK, lo cual debería salir rápido.

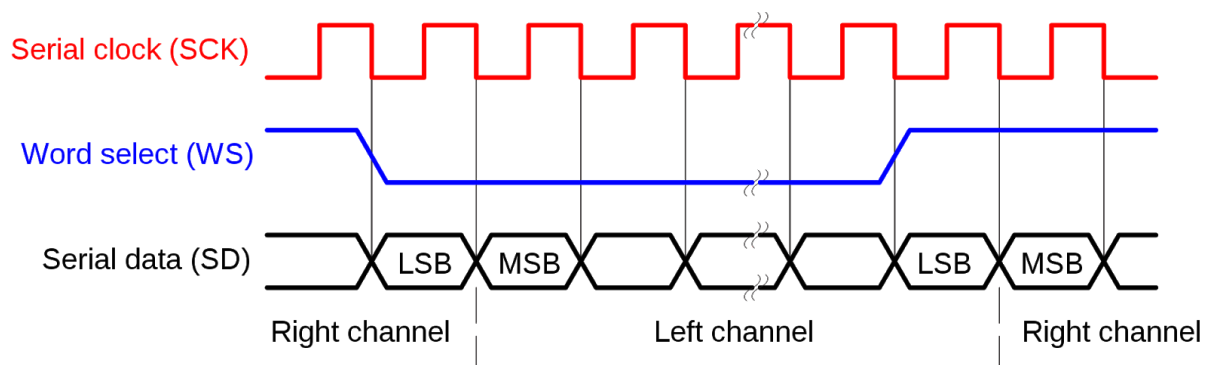
Pregunta del millón, necesito DMA? Si es asi, para que?

Esto debería estar respondido en Audio Output Kinetis.pdf

I2S: Conexionado con la placa UDA 1380:

La idea con I2S es que es un protocolo serial cuya salida es claramente digital.

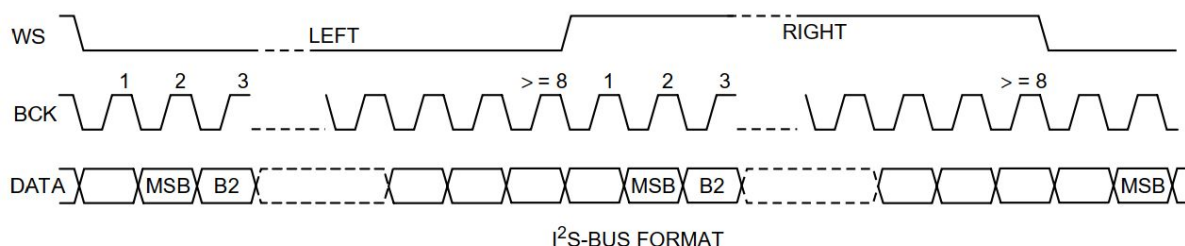
¿En qué consiste? En 3 cables:



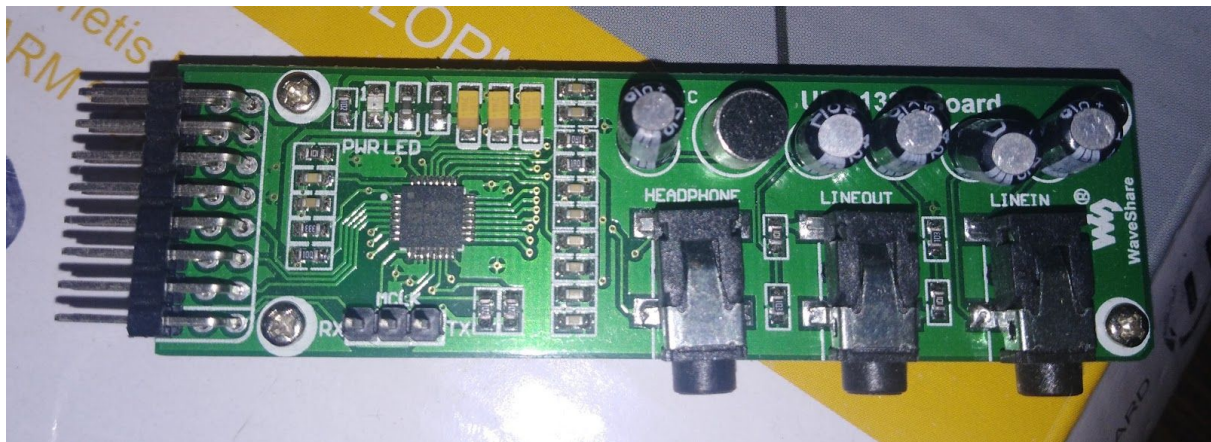
A veces incluye un MCLK, typically $256 \times \text{LRCLK}$. This is not part of the I2S standard,[4] but is commonly included for synchronizing the internal operation of the analog/digital converters.

(fuente: <https://en.wikipedia.org/wiki/I%C2%B2S>)

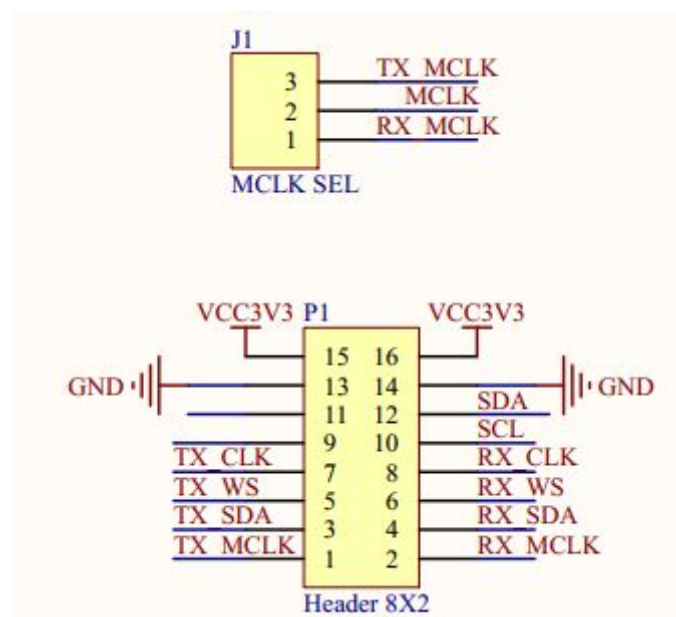
Veamos como es i2s para la datasheet de UDA 1380



Si nos fijamos en la placa:



En esta primera imagen solo se ven **RX**, **MCLK** y **TX** abajo a la izquierda.



En el esquemático aparece como un **MCLK SEL** (ver UDA 1380-Board-Schematic.pdf).

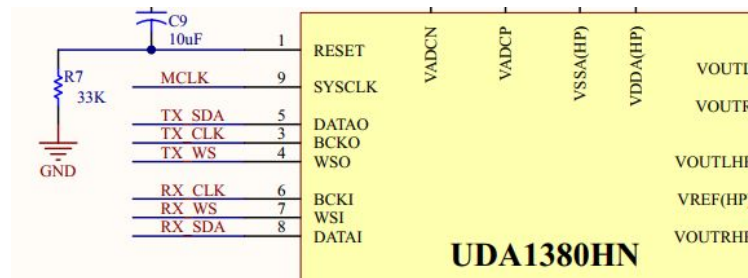
MCLK es la entrada **SYSCLK** que maneja el UDA 1380.

Con un jumper puedo elegir quien le da el clock al **MCLK**, si el **RX_MCLK** o el **TX_MCLK**.

O, puedo poner directamente el MCLK de la kinetis al MCLK de UDA, que creo que es lo más viable.

Al parecer soporta tener RX_MCLK y TX_MCLK distintos y con el jumper elegir cual es MCLK.

Lo que **nos interesa** de todas formas es la parte de **RX**, que es a la que le vamos a mandar el audio en **punto fijo (o eso creo)**.



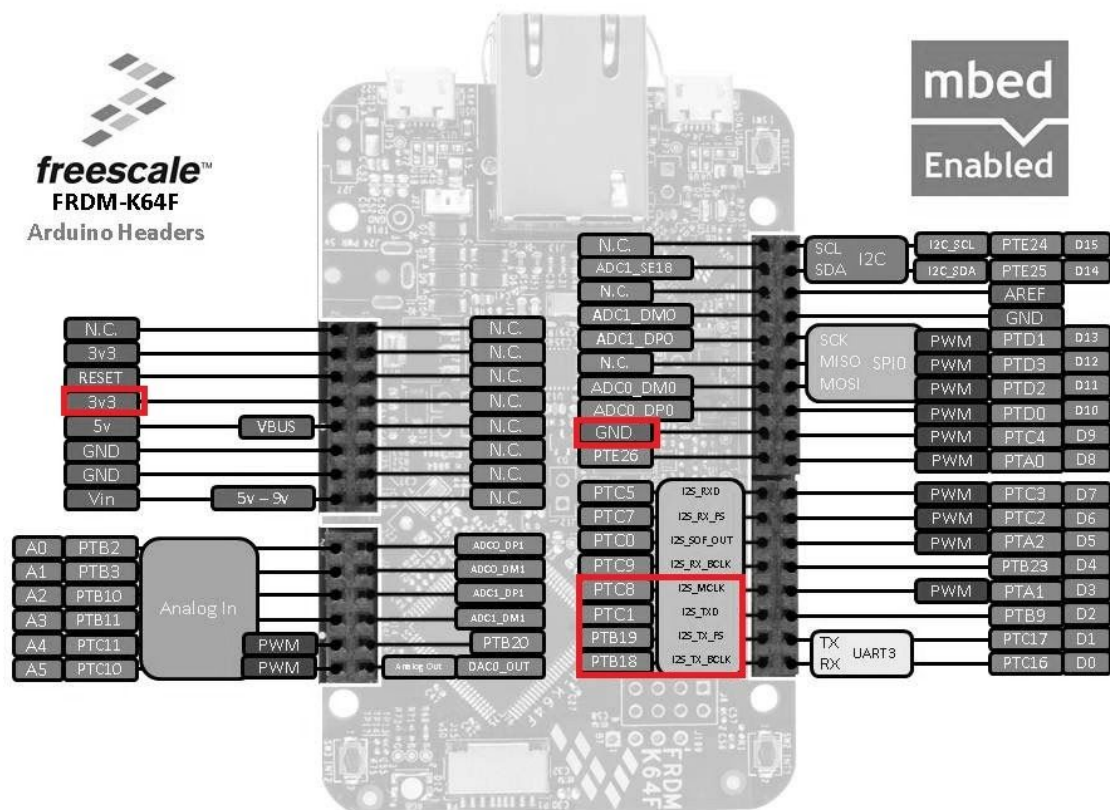
Sabemos (por el esquemático) que **RXCLK** debe ser el bit clock, marcado por **I2S_TX_BCLK** de la K64F.

Las conexiones finales de la FRDM K64F a la placa UDA1380 son:

FRDM 64F => UDA1380

- I2S_MCLK => MCLK
- I2S_TXD => RX_SDA
- I2S_TX_FS => RX_WS
- I2S_TX_BCLK => RX_CLK
- VCC y GND (tener cuidado, ver esquemático, el header es medio feo)

Utilizando los siguientes pines de la FRDM K64F:



FLASH (Medium) Se puede bypasear? Si, en el programa puedo guardar un arreglo con el archivo "sonic.mp3" todo en hexa. Así que en principio no sería tan vital. Aunque habría que pensar cómo conectar si funciona todo sin haberlo cargado en la Flash, porque si no estoy mal utiliza buffers para obtener el contenido de la flash.

DECODE MP3 (Medium/Hard)

¿Qué hace esto? Agarra un buffer de entrada y escribe en otro buffer de salida con el mp3 decodificado (pendiente ver en qué formato, ya que creo que i2s usa punto fijo, esto debería devolver en un buffer punto fijo).