

22.49 Laboratorio de DSP y FPGA

# Análisis de la interfase códec CS4218 y DSP56307

TRABAJO PRÁCTICO N° 4

**GRUPO N° 2**

DAVIDOV, Gonzalo Joaquín

FARALL, Facundo David

KAMMANN, Lucas Agustín

TROZZO, Rafael Nicolás

## Contenido

Objetivo .....	3
Alcance .....	4
Análisis.....	5
Descripción General .....	5
Diagrama de Bloques .....	5
Configuración de la interfaz del CS4218 .....	6
Software .....	8
Hardware.....	10
Filtro antialiasing .....	12
Puerto de comunicación para <b>control</b> .....	13
Puerto de comunicación para <b>muestras</b> .....	13
Configuración de la frecuencia de muestreo .....	13
Conexión de entradas <b>LIN1</b> y <b>RIN1</b> .....	13
Configuración <b>S.MODE</b> .....	14

## Objetivo

Analizar y documentar la interfaz entre el códec CS4218 y el procesador DSP56307 en la placa de desarrollo DSP56307EVM.

## Alcance

Se analiza la interfaz desde el punto de vista del hardware, es decir, los circuitos eléctricos y los protocolos de comunicación. Además, se estudia la interfaz desde el punto de vista del software, es decir, los drivers desarrollados en assembler para el DSP56307.

## Análisis

### Descripción General

El CS4218 es un códec de audio estéreo de 16 bits que permite utilizar hasta cuatro entradas de audio y dos salidas de audio. Para esto, posee dos conversores A/D y dos conversores D/A, empleando un esquema de modulación delta-sigma y 64X oversampling. Dado que es un códec de audio estéreo, las entradas y salidas pueden ser agrupadas de a pares (canales izquierdo y derecho), y la entrada posee un seleccionador (multiplexor) para seleccionar uno de los posibles pares.

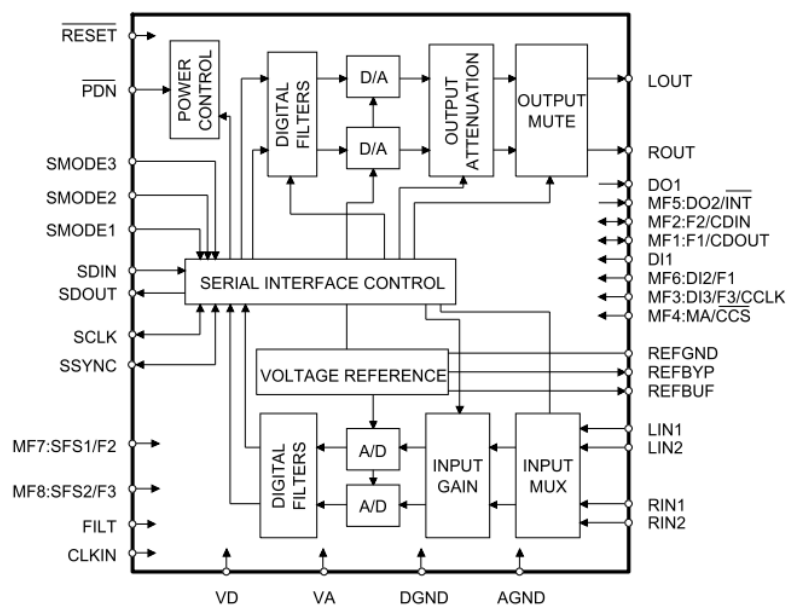


Ilustración 0.1: Esquema CS4218

El códec permite configurar una ganancia para las entradas analógicas y una atenuación o silenciado de las salidas analógicas. Por otro lado, posee una interfaz digital de comunicación denominada **SSI**, a través de la cual se comunica con el DSP56307 para recibir las muestras digitales salientes y transmitir las muestras digitales entrantes.

### Diagrama de Bloques

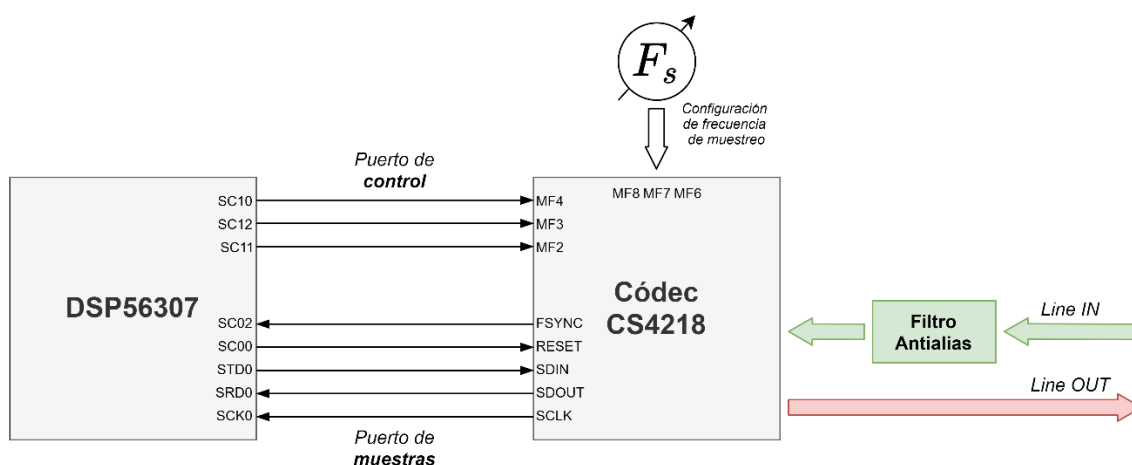


Ilustración 0.2: Diagrama en bloques

## Configuración de la interfaz del CS4218

El códec CS4218 tiene tres entradas para configurar el modo de operación de las comunicaciones, y en la placa de desarrollo DSP56307EVM, se encuentran conectadas con los estados lógicos  $SMODE_1 = 0$ ,  $SMODE_2 = 0$ ,  $SMODE_3 = 1$ . Esto configura al códec en el **Serial Mode 4 (SM4)**.

Bajo esta condición, el códec opera como maestro de la comunicación serial, por ende, controla la señal de clock y la señal de sincronismo. La señal **SDIN** recibirá las muestras para el convertor D/A, la señal **SDOUT** enviará las muestras del convertor A/D, y la señal **SSYNC** indica el comienzo de un marco (frame) de datos. La señal **SCLK**, será automáticamente generada según la frecuencia de muestreo.

Por otro lado, bajo este modo de operación se transmiten paquetes de 32 bits que se encuentran sincronizados con la señal **SSYNC**. Cada paquete contiene la muestra de 16 bits del canal izquierdo y el canal derecho. La Ilustración 0.3 muestra un diagrama temporal de la comunicación.

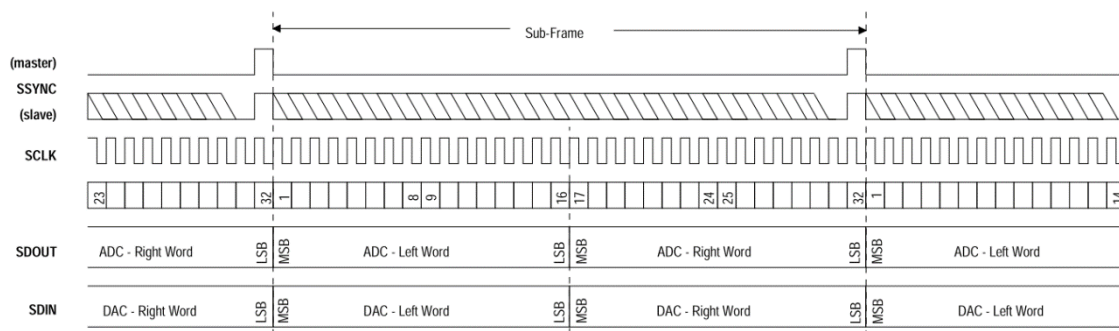


Ilustración 0.3: Diagrama temporal de la comunicación

La Tabla 1 muestra las opciones de configuración para la frecuencia de muestreo utilizando los pines **MF6**, **MF7** y **MF8**. Vale aclarar que esta tabla en particular es válida cuando  $CLKIN = 12,288MHz$ , lo cual se cumple en la placa de desarrollo DSP56307EVM. Esto se debe a que la frecuencia **CLKIN** debe ser al menos 256 veces más grande que la máxima frecuencia de muestreo utilizada (48kHz en este caso), por ende,  $CLKIN=12,288MHz$ .

MF6	MF7	MF8	Frecuencia de muestro (kHz)
0	0	0	48.00
0	0	1	32.00
0	1	0	24.00
0	1	1	19.20
1	0	0	16.00
1	0	1	12.00
1	1	0	9.60
1	1	1	8.00

Tabla 1: Frecuencias de muestreo configurables

Entre el DSP56307 y el códec CS4218, existen dos interfaces de comunicación. Una dedicada a la transferencia de muestras de audio entrantes y salientes, y otra dedicada al control del códec. Esta última se emplea para configurar opciones internas del dispositivo, como es la atenuación de la salida analógica del códec o la ganancia de su entrada analógica.

Se trata de una comunicación asincrónica, y sólo hace falta utilizarla durante la inicialización del códec y cuando se deban modificar opciones internas. En la Ilustración 0.4 se muestra el diagrama de tiempos de una comunicación de control. La señal **CCS** es activa baja, se debe poner en 0 para iniciar una secuencia de control. Se puede configurar si enmascarar las interrupciones del códec (**MASK**), atenuaciones de los conversores D/A de ambos canales (**D/A Att.**), silenciar las salidas (**MUTE**), ganancias de los conversores A/D de ambos canales (**A/D Gain**). Además, se selecciona de donde tomar la entrada multiplexada de datos (**ISL e ISR**), que para el caso de la placa de desarrollo es RIN2 y LIN2.

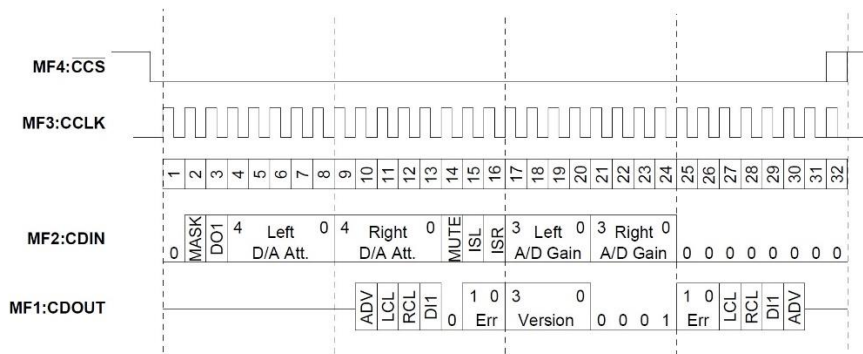


Ilustración 0.4: Señales de control en SM4

En resumen, los aspectos a tener presentes son,

- ✓ El CS4218 es el maestro de la comunicación
- ✓ El CS4218 genera la señal de clock
- ✓ El CS4218 genera el pulso de sincronismo, que será de duración de un bit, y estará un ciclo de clock antes de que empiece el marco de datos
- ✓ Los datos de audio están alineados de acuerdo con MSB first
- ✓ Los datos están en formato complemento a 2
- ✓ Los datos son válidos en el flanco descendente
- ✓ La comunicación de control debe hacer dos transferencias cuando se reinicia el códec, la primera se ignora y la segunda se utiliza para poder configurar las opciones como se desea

## Interfaz del DSP56307

El DSP cuenta con dos módulos para interfaz serie sincrónica llamados Enhanced Synchronous Serial Interface (ESSI), uno vinculado al Puerto C (ESSI0) y el otro al Puerto D (ESSI1). Ambos son usados, el primero para la interfaz de datos y el segundo para la interfaz de control con el códec. Por lo tanto, sus configuraciones serán distintas. La dirección de los pines de los puertos deberá corresponderse con el diagrama en bloques de la Ilustración 0.2.

La correspondencia entre las señales de las interfaces ESSI y los pines de los puertos C y D se muestran en la Ilustración 0.5.

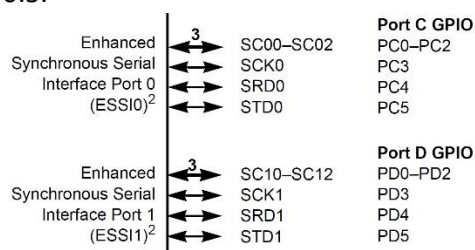


Ilustración 0.5: Pines de ESSI y GPIO

## Software

### Subrutina "ada\_init"

La subrutina "ada\_init" se encarga de inicializar los registros del DSP y enviar los mensajes de control necesarios al codec para la normal operación.

Primero se reinician los puertos ESSIO y ESSI1 (Enhanced Synchronous Serial Interface), lo que hace esto es setear los puertos C y D como GPIO, para no tenerlos configurados como ESSI mientras se configura el resto del periférico. Luego se configuran los registros de control de ESSIO para la interfaz de datos.

El CRA (Control Register A) vale \$101807, esto implica:

- PM[7:0] Bits 7-0: Setean divide ratio del prescale divider del ESSI clock en 8.
- Bits 8-10: Reservados, deben valer 0.
- PSR Bit 11: Como está en 1 se bipasea el prescaler fijo.
- DC[4:0] Bits 16-12: Estos bits setean el Divide Ratio en 2, esto determina el transfer rate de la palabra.
- Bit 17: Reservado.
- ALC Bit 18: las palabras recibidas están alineadas a la izquierda del MSB (bit 23); y las transmitidas deben estar alineadas a la izquierda en el transmit register. Esto es porque se eligen palabras de 16 bits y el ESSI maneja data de 24 bits, entonces, al ser la palabra de menos de 24 bits, la data debe estar alineada al MSB.
- WL[2:0] Bits 21-19: Es el largo de la palabra a transmitir. Vale 010 que significa que las palabras tienen largo 16 bits.
- SSC1 Bit 22: Vale 0 por lo tanto se elige el modo sincrónico y el transmisor 2 esta deshabilitado. SC1 actúa como el flag de serial I/O y la señal SC1 es el output.
- Bit 23: Reservado.

Luego, se setea el CRB (Control Register B) a \$ff330c que implica:

- Bit 0: Seteado a 0, la data que este bit tenga será escrita en SC0 al principio del próximo time slot. (por estar en network mode).
- Bit 1: Seteado a 0, la data que este bit tenga será escrita en SC1 al principio del próximo time slot. (por estar en network mode).
- Bit 2: Seteado a 1, pone a SC0 como output.
- Bit 3: Seteado a 1, pone a SC1 como output.
- Bit 4: Seteado a 0, pone a SC2 como input.
- Bit 5: Seteado a 0, selecciona un clock externo.
- Bit 6: Seteado a 0, la información entra y sale con el MSB primero.
- Bits 7-8: En 10 hacen que RX y TX tengan frame syncs del tamaño de un bit.
- Bit 9: Seteado a 1, frame sync sucede un ciclo de clock antes.
- Bit 10: Seteado a 0, frame sync tiene polaridad positiva.
- Bit 11: Seteado a 0, hace que la data y frame sync clocked out on rising edge and clocked in on falling edge.
- Bit 12: Seteando a 1, ESSI en modo sincronico.
- Bit 13: Seteado a 1, elige network mode, hace que se transmita una palabra en cada slot de tiempo.
- Bit 14: Seteando a 0, deshabilita TE2.
- Bit 15: Seteando a 0, deshabilita TE1.



- Bits 16-23: Seteando a 1, habilita TE0, RE, TIE, RIE, TLIE, RLIE, TEIE, REIE

```
;-----  
; Initialize the CS4218 codec  
; Serial Mode 4 (SM4), DSP Slave/Codec Master, 32-bits per frame  
;-----  
ada_init  
    ; reset ESSI ports  
    movep    #$0000,x:M_PCRC  
    movep    #$0000,x:M_PCRD  
  
    ; Set Control Register A and B  
    movep    #$101807,x:M_CRA0  
    movep    #$ff330c,x:M_CRB0
```

Luego, se configuran los pines correspondientes de los puertos C y D (ESSIO y ESS1) como salidas, según fue indicado previamente. Para configurar un bit de un puerto como salida, se debe escribir un 1 en su Port Direction Register (PRRx). El único bit que no se configura aquí como salida es STD0, ya que queda configurado como salida al configurar CRA y CRB de ESSIO, por usar modo sincrónico y SSC1=1.

```
; Configure GPIO pins  
movep    #$0001,x:M_PRRC  
movep    #$0007,x:M_PRRD
```

Se resetea el códec poniendo un 0 en el bit reset del PDRC y un 0 en el CCS del PDRD para poder modificar el CR. Luego se inserta un delay de 50ms para darle tiempo de resetearse al códec. Se vuelve a setear el bit de reset en estado inactivo.

```
; Codec Reset  
bclr     #CODEC_RESET,x:M_PDRC  
bclr     #CCS,x:M_PDRD  
  
; Delay to allow Codec to reset  
do        #1000,_delay_loop  
rep       #1000                      ; minimum 50 ms delay  
nop  
_delay_loop
```

Una vez reseteado el códec, se lo configura como la aplicación lo requiere. Para ello, en algún lugar del código se debe definir la secuencia de control a ser enviada al códec, para configurar lo indicado en la Ilustración 0.4. Estas configuraciones son dependientes de la aplicación, y la única que debe ser siempre respetada es poner un 1 en ISL e ISR para seleccionar LIN2 y RIN2 como entradas del códec, ya que son las utilizadas en la placa de desarrollo. Los valores de la configuración se guardan en la memoria reservada CTRL\_WD\_HI y CTRL\_WD\_LO para que pueda acceder a ellas la subrutina codec\_control. En primer lugar se envía un mensaje cuyo contenido no importa porque el códec lo ignora, y luego se envía el mensaje con las configuraciones deseadas.

```
; Setting up to send Codec control information  
bset     #CODEC_RESET,x:M_PDRC  
  
; Sending control words  
set_control  
    move     #CTRL_WD_12,x0
```

```

move      x0,x:CTRL_WD_HI
move      #CTRL_WD_34,x0
move      x0,x:CTRL_WD_LO
jsr       codec_control      ; dummy control information
jsr       codec_control      ; correct control information

```

Finalmente, se configura la prioridad de las interrupciones del ESSI0 a nivel 3 y se las habilita y se habilita el ESSI0 poniendo \$003e en el PCRC.

```

; Set and enable interrupts
movep     #$000c,x:M_IPRP
andi      #$fc,mr           ; enable interrupts

; Set ESSI functionality
movep     #$003e,x:M_PCRC
rts

```

### Subrutina “codec\_control”

La subrutina “codec\_control” se encarga de enviar al códec la señal de control cargada por “ada\_init”, la cual es guardada en las posiciones de memoria X indicadas por CTRL\_WD\_LO y CTRL\_WD\_HI. Se envían dos secuencias de 16 bits siguiendo la secuencia de la Ilustración 0.4. En primer lugar, se pone en 0 el bit CCS, luego se envía cada parte del mensaje cargando los 16 bits en A1 y llamando a la subrutina “send\_codec”. Finalmente, se vuelve a poner en 1 CCS.

```

;-----
; codec_control routine
;   Input: CTRL_WD_LO and CTRL_WD_HI
;   Output: CDIN
;   Description: Used to send control information to CODEC
;-----
codec_control
    clr      a
    bclr     #CCS,x:M_PDRD      ; assert CCS
    move     x:CTRL_WD_HI,a1    ; upper 16 bits control data
    jsr      send_codec         ; shift out upper word
    move     x:CTRL_WD_LO,a1    ; lower 16 bits control data
    jsr      send_codec         ; shift out lower word
    bset     #CCS,x:M_PDRD      ; deassert CCS
    rts

```

### Subrutina “send\_codec”

La subrutina “send\_codec” envía al códec una secuencia de 16 bits de control obtenida del registro A1. Para ello, como indica la Ilustración 0.4, envía de a un bit en la línea CDIN, generando un flanco ascendente de CCLK al comienzo y un flanco descendente cuando el valor del bit está listo. Lo que hace es bit a bit, comenzando por el más significativo, poner un 1 o un 0 en CDIN dependiendo del valor del mensaje, y así sucesivamente 16 veces hasta finalizar.

```

;-----
; send_codec routine
;   Input: a1 containing control information
;   Output: sends bits to CDIN
;   Description: Determines bits to send to CDIN
;-----
send_codec

```

```

do      #16,end_send_codec      ; 16 bits per word
bset    #CCLK,x:M_PDRD          ; toggle CCLK clock high
jclr    #23,a1,bit_low          ; test msb
bset    #CDIN,x:M_PDRD          ; send high into CDIN
jmp     continue
bit_low bclr    #CDIN,x:M_PDRD    ; send low into CDIN
continue rep    #2                ; delay
nop
bclr    #CCLK,x:M_PDRD          ; restart cycle
lsl     a                      ; shift left control word 1b
end_send_codec
rts

```

## Rutinas de interrupción

### Interrupciones por último slot

**Receive last slot:** Al estar el DSP en Network Mode y habilitado el bit **RLIE**, es interrumpido cuando termina de recibir el último slot de un frame, es decir, al terminar de recibir ambos canales.

```

,***** SSI receive last slot ISR *****
ssi_rxls_isr
    ; código de la aplicación
    rti

```

**Transmit last slot:** Al estar el DSP en Network Mode y habilitado el bit **TLIE**, es interrumpido cuando empieza a transmitir el último slot de un frame, es decir, la última muestra de ambos canales. Pone en 1 un bit para indicar que terminó la transmisión del canal derecho (ver en Ilustración 0.3 que el canal derecho va último), ese bit es observado por la interrupción de recepción. Entonces, si el bit vale 1 se terminó de transmitir el canal derecho.

```

,***** SSI transmit last slot ISR *****
ssi_txls_isr
    bset    #Left_ch, x:bits
    rti

```

### Interrupción por recepción

Al recibir una nueva muestra del códec, el DSP es interrumpido. En esta interrupción en primer lugar limpia el flag leyéndolo, luego lee la muestra del registro de datos de RX de ESSIO. Luego se fija en el bit mencionado anteriormente si se trata del canal derecho o izquierdo. Si es el canal izquierdo, debe correrse código de la aplicación que guarde la muestra de salida deseada en X1, como por ejemplo calcular la salida de un filtro. Además, se sube y se baja un pin, lo cual permite por ejemplo medir cuánto tarda la ejecución del código de la aplicación. Si es el canal derecho, escribe un 0 en X1, porque el código utilizado ignora uno de los canales. Se escribe lo que haya quedado en X1 al registro de datos de TX de ESSIO, el periférico se encargará de enviarlo.

Finalmente, se togglea el bit que indica si se trata de canal izquierdo o derecho.

```

,***** SSI receive ISR *****
ssi_rxe_isr
    bclr    #5,x:M_SSISR0      ; Clear exception flag reading it
ssi_rx_isr
    movep   x:M_RX0,y1         ; Read A/D data
    jclr    #Left_ch,X:bits,esright

```

```

        ORI        #08,MR
        movep      #0001,X:M_HDR    ;1->PB0, sube el pin
        ; código de la aplicación, escribe muestra en X1
        movep      #0000,X:M_HDR    ;0->PB0, baja el pin
        jmp        endisr

esright move      #0,x1              ;mute the other channel

endisr movep      x1,x:M_TX00        ; write d/a data
        bchg       #Left_ch,x:bits
        rti

```

### Interrupción por transmisión

Al terminar de transmitir una muestra de cualquiera de los dos canales, se interrumpe. Se limpia el flag leyéndolo y se inserta el código que la aplicación requiera ante esta situación.

```

,***** SSI transmit ISR *****
ssi_txe_isr
        bclr       #4,x:M_SSISR0    ; Clear exception flag reading it

ssi_tx_isr
        ; código de la aplicación
        rti

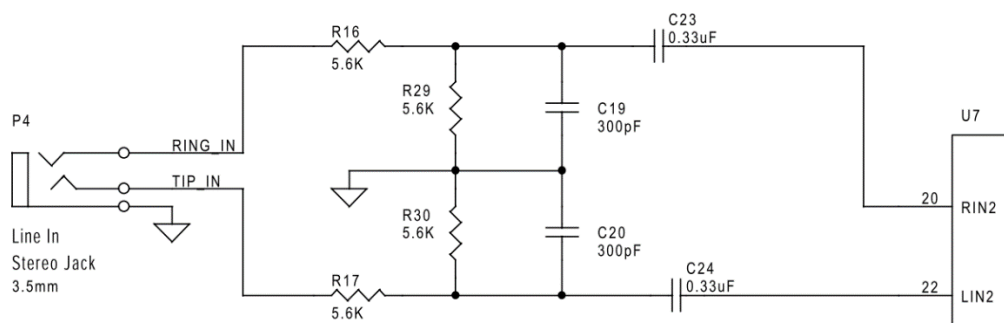
```

## Hardware

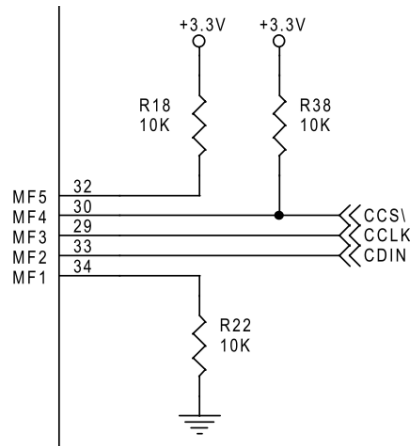
En esta sección se analizan los circuitos eléctricos empleados en la interconexión entre el DSP56307 y el códec CS4218, así como también los circuitos de soporte para garantizar el funcionamiento del CS4218.

### Filtro antialiasing

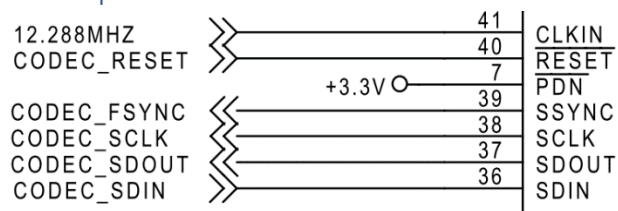
En la entrada de audio estéreo, en ambos canales, se coloca un divisor de tensión utilizando dos resistencias para atenuar al 50% la amplitud de la señal de entrada, permitiendo un máximo de  $2V_{RMS}$ . Por otro lado, se utiliza un filtro pasabajos de primer orden para filtrar las altas frecuencias y proteger al sistema de aliasing. Vale destacar que en este sistema se puede simplificar el filtro ya que el CS4218 utiliza oversampling, y eso reduce las exigencias sobre la pendiente de atenuación del filtro.



## Puerto de comunicación para control

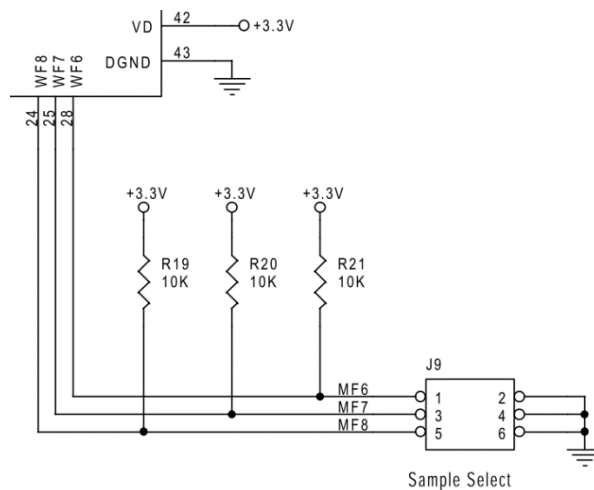


## Puerto de comunicación para muestras



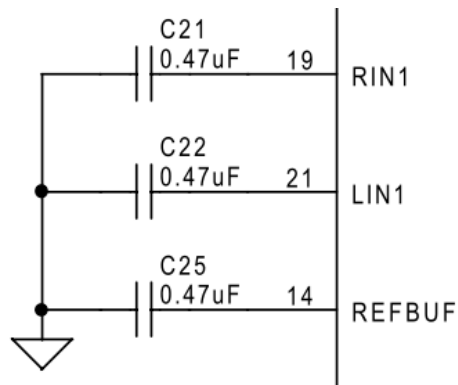
## Configuración de la frecuencia de muestreo

Las entradas **MF6**, **MF7** y **MF8** se pueden conectar a masa o a la alimentación para seleccionar una de las ocho alternativas posibles de frecuencia de muestreo con que trabajará el CS4218, siendo que está configurado en modo maestro y maneja el clock de la comunicación.



## Conexión de entradas LIN1 y RIN1

El códec CS4218 posee dos grupos de entradas estéreo, pero en la placa de desarrollo DSP56307EVM sólo se utiliza una de ellas. El fabricante recomienda que las entradas no utilizadas se conecten a la masa del circuito a través de un capacitor para evitar consumo de corriente continua.



### Configuración **SMODE**

En los pines **SMODE1**, **SMODE2** y **SMODE3**, se conectan los niveles de tensión para configurar el modo de comunicación SM4 (Serial Mode 4).

