

COMO CREAR UN PROYECTO EN DSK6713

Autor: Gonzalo Sad

A continuación se detallan los pasos a seguir para realizar un proyecto de propósito general en la placa de desarrollo DSK6713, el cual servirá de base para futuras aplicaciones. Para esto se generará un programa que simplemente recibirá muestras de audio por línea de entrada y las mandará a la salida de auriculares. A grandes rasgos, existen dos formas de hacer esto: mediante interrupciones o mediante polling. Aquí se utilizará el método de interrupciones. Se asume que ya se tiene instalado el software Code Composer Studio (CCS).

En los proyectos que se a realizarán en este curso, se utilizarán funciones para la configuración y comunicación con el codec embebido en la placa, para lo cual se usarán los siguientes archivos: **c6713dskinit.c** y **C6713dskinit.h**. También se deberán configurar las interrupciones utilizadas, para lo cual se utilizará el archivo: **Vectors_intr.asm**. Por último, se deberán configurar los espacios de memoria utilizados por el DSP, para lo cual se utilizará el siguiente archivo de configuración: **C6713dsk.cmd**.

Cabe mencionar que los proyectos pueden realizarse con la placa conectada o desconectada a la PC.

PROGRAMA

Como se mencionó antes, se generará un programa que simplemente repite lo que recibe por línea de entrada en la salida de auriculares. El código del programa es el siguiente:

```
#include "DSK6713_AIC23.h"

#define DSK6713_AIC23_INPUT_MIC 0x0014
#define DSK6713_AIC23_INPUT_LINE 0x0011
#define LEFT 1
#define RIGHT 0

Uint16 inputsource = DSK6713_AIC23_INPUT_LINE;
Uint32 fs = DSK6713_AIC23_FREQ_8KHZ;
```

```

interrupt void c_int11()
{
    // --- Variables -----
    float x;
    union{ Uint32 uint;
           short channel[2];
        }AIC_data;

    // --- Recibo datos del Codec -----
    x = (float)(input_left_sample());

    // --- Procesamiento -----
    // Acá se realizaría el procesamiento...

    // --- Mando datos al Codec -----
    AIC_data.uint = 0;
    AIC_data.channel[LEFT] = (short)x;
    AIC_data.channel[RIGHT] = (short)x;
    MCBSP_write(DSK6713_AIC23_DATAHANDLE,AIC_data.uint);
}

void main()
{
    comm_intr();
    while(1);
}

```

Básicamente, el programa está dividido en 3 partes:

Main

Aquí es donde va a parar el DSP al iniciar el programa. La función `comm_intr()` (definida en `c6713dskinit.c`) inicializa la placa en general, configura el vector de interrupciones y configura e inicializa el codec. Luego queda en un bucle infinito, ya que nos manejaremos mediante interrupciones que serán generadas por el codec.

Interrupciones

Aquí es donde va a parar el DSP cuando se genera una interrupción ocasionada por al codec. Se define un dato tipo union, para manejar los datos

estereo mandados al codec. Se recibe el dato muestreado por el codec mediante la función `input_left_sample()` (definida en `c6713dskinit.c`). Finalmente se manda el dato (sin modificar) al codec, por los dos canales (izquierdo y derecho), utilizando la función `MCBSP_write()` (definida en la librería `cs/6713.lib`, correspondiente al DSP).

Cabe remarcar las conversiones de datos realizadas, tanto al recibir como al mandar los datos. El codec trabaja con datos del tipo entero, mientras que el procesamiento, generalmente, conviene hacerlo en punto flotante debido a la mayor precisión que se obtiene.

Include, definiciones y variables

Aquí se incluyen los headers utilizados. En este caso solamente el `DSK6713_AIC23.h`, que contiene funciones para comunicarse con el codec. También se definen variables utilizadas para configurar el codec, como `inputsource` que define la entrada del codec (línea de entrada o micrófono) y `fs` que define la frecuencia de muestreo utilizada.

GENERACIÓN DEL PROYECTO

- 1. Crear la carpeta del proyecto.** Se crea una carpeta con el nombre del proyecto, por ejemplo: *ejemplo_DSK*, en la cual se ubicarán los archivos generados.
- 2. Copiar archivos necesarios.** Como ya se mencionó antes, se utilizarán funciones previamente creadas para la comunicación con el codec, como así también para el manejo de interrupciones. También se utilizará un archivo de configuración de memoria creado para propósitos generales. Por lo tanto, se deberá copiar en la carpeta *ejemplo_DSK* los archivos: ***c6713dskinit.c***, ***C6713dskinit.h***, ***Vectors_intr.asm*** y ***C6713dsk.cmd***.
- 3. Abrir CCS.** Si la placa no está conectada a la PC, aparecerá un mensaje advirtiéndolo: *“Error initializing emulator”*. Presionar el botón *Ignore*. En caso de estar conectada, dicho mensaje no aparece.
- 4. Crear un proyecto nuevo.** Seleccionar el menú *Project* → *New*, en el cual aparecerá una ventana de configuración. En *Project name* colocar el nombre del proyecto, por ejemplo *“DSK_ej1”*, y en *Location* la ubicación de la carpeta

ejemplo_DSK. En *Project Type* seleccionar *Executable (.out)*. En *Target* seleccionar *TMS320C67XX*.

5. **Crear el programa (main).** Este archivo (.c) contendrá la función main del proyecto. Seleccionar el menú *File* → *New* → *Source File*. Copiar el código que se desarrolló en la sección anterior. Guardamos el archivo: seleccionamos el menú *File* → *Save*, buscamos la carpeta *ejemplo_DSK*, en *Tipo* seleccionamos *C/C++ Source files (*.c*)* y en *Nombre* ponemos el nombre del archivo, por ejemplo “*DSK_ej1_main*”.
6. **Agregamos el programa al proyecto.** Seleccionamos el menú *Project* → *Add Files to Project...* y elegimos el archivo generado en el paso anterior (*DSK_ej1_main.c*). Presionamos *Abrir*. En el panel de la izquierda dentro de la carpeta *Source*, aparecerá ahora este archivo.
7. **Agregamos el archivo *c6713dskinit.c* al proyecto.** Utilizando la misma metodología que en el paso 6, agregamos ahora el archivo para comunicarnos con el codec: *c6713dskinit.c*.
8. **Agregamos las librerías necesarias.** Utilizando la misma metodología que en el paso 6, agregamos las siguientes librerías:
 - **rts6700.lib** → Run-time Support, funciones soporte para arquitectura C67x). Se encuentra en *C:\CCStudio_v3.1\C6000\cgtools\lib*.
 - **dsk6713bsl.lib** → Board Support Library, funciones soporte para la placa DSK6713. Se encuentra en *C:\CCStudio_v3.1\C6000\dsk6713\lib*.
 - **cs16713.lib** → Chip Support Library, funciones soporte para el proc. C6713. Se encuentra en el directorio *C:\CCStudio_v3.1\C6000\cs\lib*.
9. **Agregamos el archivo *Vectors_intr.asm* al proyecto.** Utilizando la misma metodología que en el paso 6, agregamos ahora el archivo para configurar las interrupciones: *Vectors_intr.asm*.
10. **Agregamos el archivo *C6713dsk.cmd* al proyecto.** Utilizando la misma metodología que en el paso 6, agregamos ahora el archivo para configurar la memoria: *C6713dsk.cmd*.
11. **Configuraciones de compilación.** Antes de compilar nuestro programa, debemos configurar nuestro proyecto. Seleccionamos el menú *Project* → *Build Options...* Aparecerá una ventana con diferentes pestañas, de las cuales solo configuraremos 2: *Compiler* y *Linker*.

Compiler

En el panel *Category* seleccionamos *Basic* y solamente cambiamos:

- *Target Version* → *C671x (-mv6710)*
- *Opt. Level* → *File (-o3)*

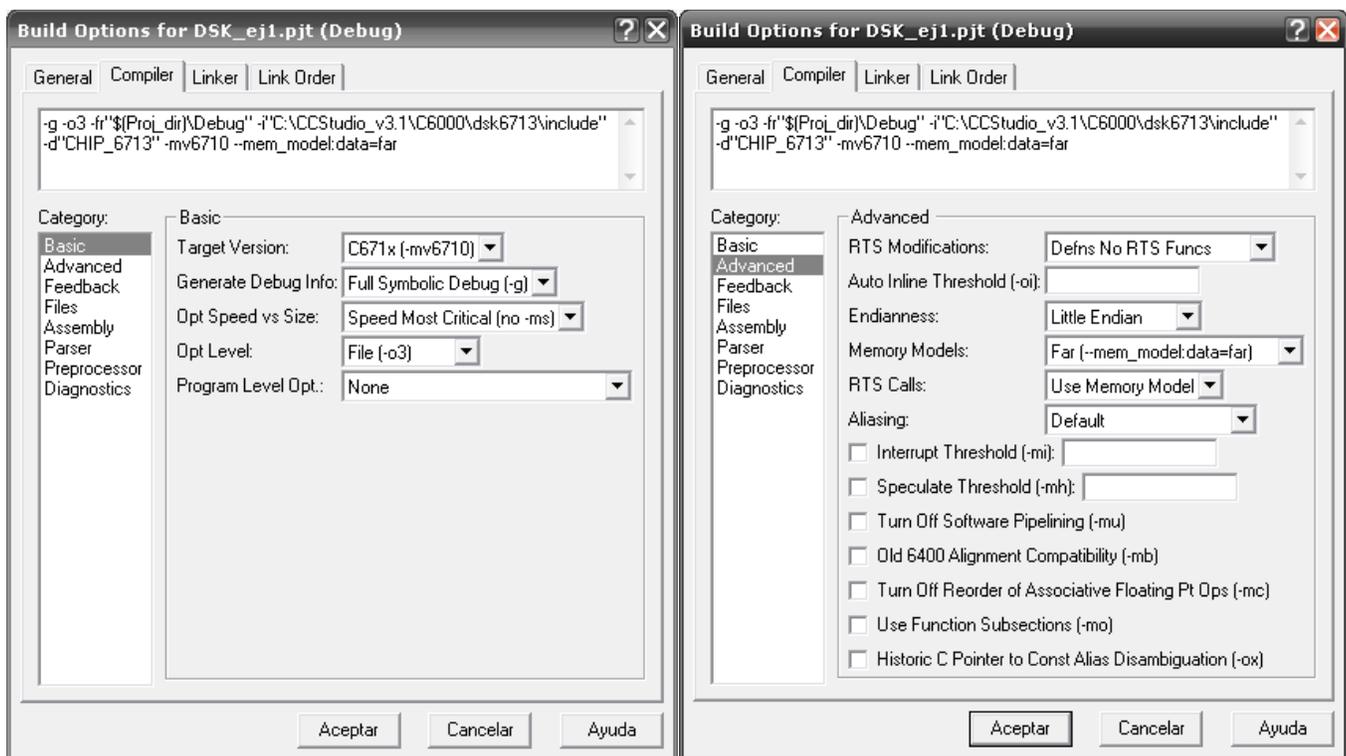
En el panel *Category* seleccionamos *Advanced* y solamente cambiamos:

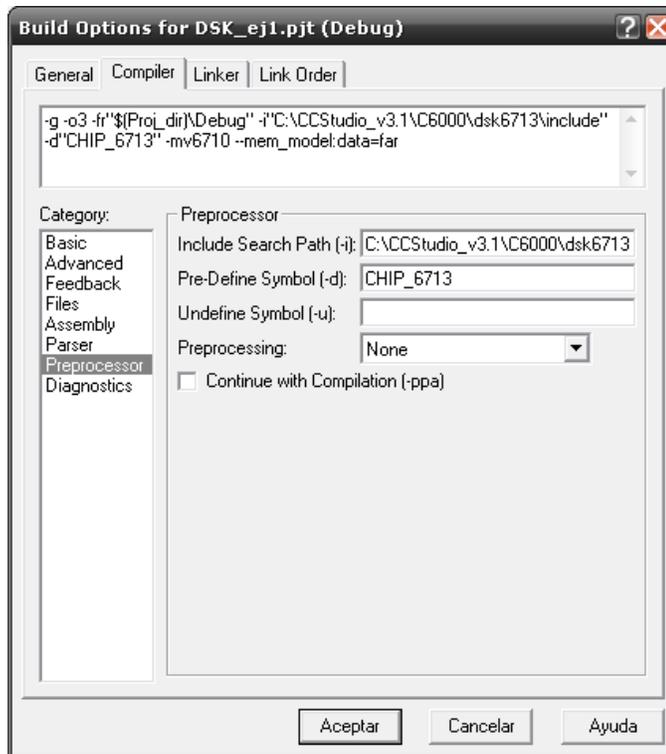
- *Memory Models* → *Far (--mem_model:data=far)*

En el panel *Category* seleccionamos *Preprocessor* y solamente cambiamos:

- *Include Search Path (-i)* → *C:\CCStudio_v3.1\C6000\dsk6713\include*
- *Pre-Define Symbol (-d)* → *CHIP_6713*

A continuación se muestra como deben quedar las configuraciones:



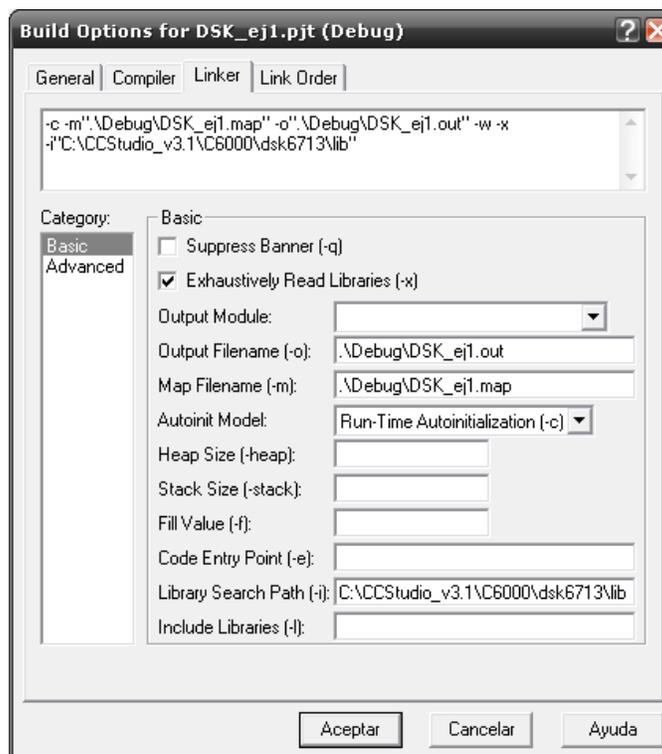


Linker

En el panel *Category* seleccionamos *Basic* y solamente cambiamos:

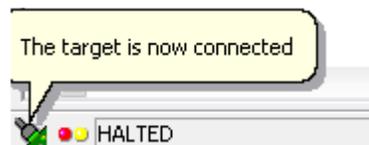
- *Library Search Path (-i)* → *C:\CCStudio_v3.1\CC6000\dsk6713\lib*

A continuación se muestra como deben quedar las configuraciones:



12. Compilamos el programa. Ahora que ya tenemos todos los archivos necesarios y todas las configuraciones seleccionadas, compilamos el programa seleccionando el menú *Project* → *Rebuild All*. Aparecerá una pestaña en la parte inferior con el nombre *Build*, en la cual se mostrarán los avances del proceso de compilación. Al final aparecerá el mensaje: “*Build complete. 0 Errors ...*” lo cual nos indica que se completó exitosamente la compilación. Puede observarse ahora en el panel de la izquierda, dentro de la carpeta *Include*, los headers utilizados en nuestro proyecto, dentro de los cuales se encuentra el *C6713dskinit.h*, que incluimos en nuestra carpeta del proyecto en el paso 2. Dentro de la carpeta de nuestro proyecto se habrá generado una nueva carpeta llamada *Debug*, dentro de la cual estará el archivo binario “*DSK_ej1.out*”, el cual será cargado al DSP.

13. Conectamos la placa. Se debe conectar la placa a la PC mediante el cable USB. Luego conectamos auriculares a la entrada de auriculares de la placa y la salida de auriculares de nuestra PC a la entrada de línea de la placa. Encendemos la placa, conectando el cable de alimentación. Una vez que terminan de titilar los leds de la misma (esto indica que la placa terminó el proceso de inicialización), se selecciona el menú *Debug* → *Connect*. Si la conexión fue exitosa, en la esquina inferior izquierda deberá aparecer el siguiente mensaje:



14. Cargamos el programa al DSP. Seleccionamos el menú *File* → *Load Program...* y buscamos el archivo *.out* generado en el paso 12. Presionamos *Abrir*.

15. Corremos el programa. Seleccionamos el menú *Debug* → *Run* y reproducimos música en nuestra PC. Por los auriculares conectados a la placa podrá escucharse la música.

16. Detener el programa. Seleccionamos el menú *Debug* → *Halt*.

17. Desconectamos la placa. Seleccionamos el menú *Debug* → *Disconnect*, desconectamos el cable de alimentación y desconectamos la placa.