

Trabajar con DSP TMS320C6748 de Texas Instrument

Javier Romero e-mail: javier.romeros@alumnos.usm.cl

Gonzalo Carrasco e-mail: gonzalo.carrasco@alumnos.usm.cl

March 10, 2018

Contents

1	Preparación	3
a	Pre-requisitos	3
b	Instalación	3
b.1	CCS	3
b.2	Librerías BSL y LCDK-AIC	5
2	Creación de un proyecto	6
a	Enlace de librería compilada del BSL	7
b	Inclusión de archivos fuente AIC	10
c	Código de ejemplo y prueba simple	10
3	Conectando la DSP y ejecutando código	11
4	Notas Finales	11

1 Preparación

a Pre-requisitos

Para trabajar con el kit DSP TMS320C6748 siguiendo este instructivo se necesita cumplir con los siguientes requisitos:

- Hardware:
 - TMS320C6748, específicamente el kit TMDSLCDK6748
 - JTAG Debugger, en esta guía se utilizará el TMS320-XDS100-V3 de Olimex
- Software:
 - CCStudio Version 7.4 (http://processors.wiki.ti.com/index.php/Download_CCS)
 - Board Support Library (BSL) (<http://support.logicpd.com/downloads/1478/>)
 - Librería AIC para LCDK (ftp://ftp.wiley.com/sci_tech_med/signal_processing/)

Como opción se puede utilizar como complemento el libro “Digital Signal Processing and Applications with the OMAP - L138 eXperimenter” de Donald Reayauth, además de la librería DSPLIB (http://processors.wiki.ti.com/index.php/C674x_DSPLIB), la cuál se utiliza en la última parte del libro.

b Instalación

b.1 CCS

Primero se instala CCStudio, para esto se debe aceptar las condiciones de uso y seguir las instrucciones en pantalla, es importante elegir las opciones de instalación para incluir *C6000 Power-Optimized DSP* y *TI XDS Debug Probe Support*. Las Fig. 1 y 2 retratan la selección de estas opciones.

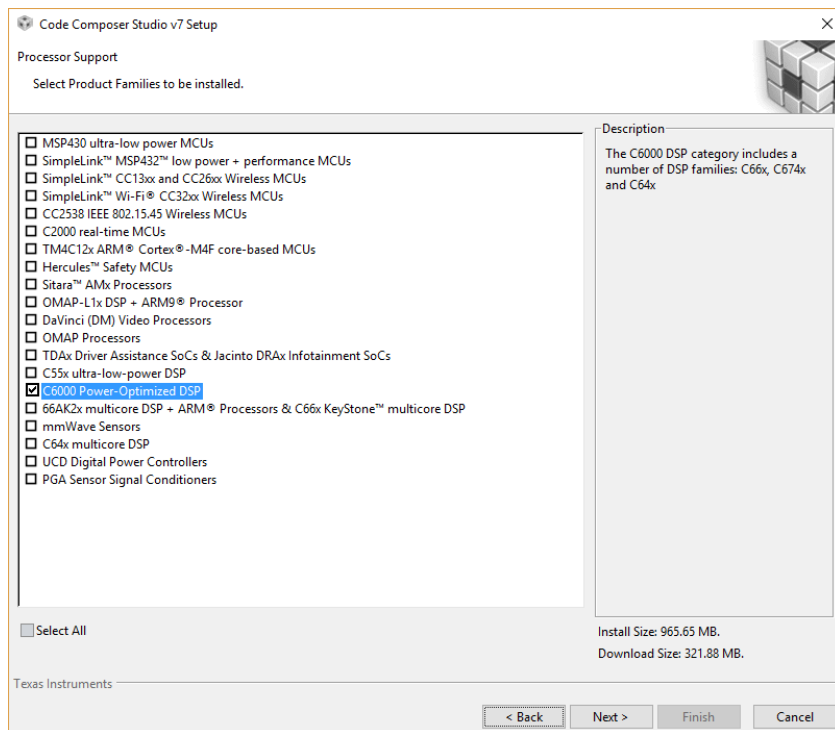


Figure 1: Selección de instalación de Familias de Productos requeridas.

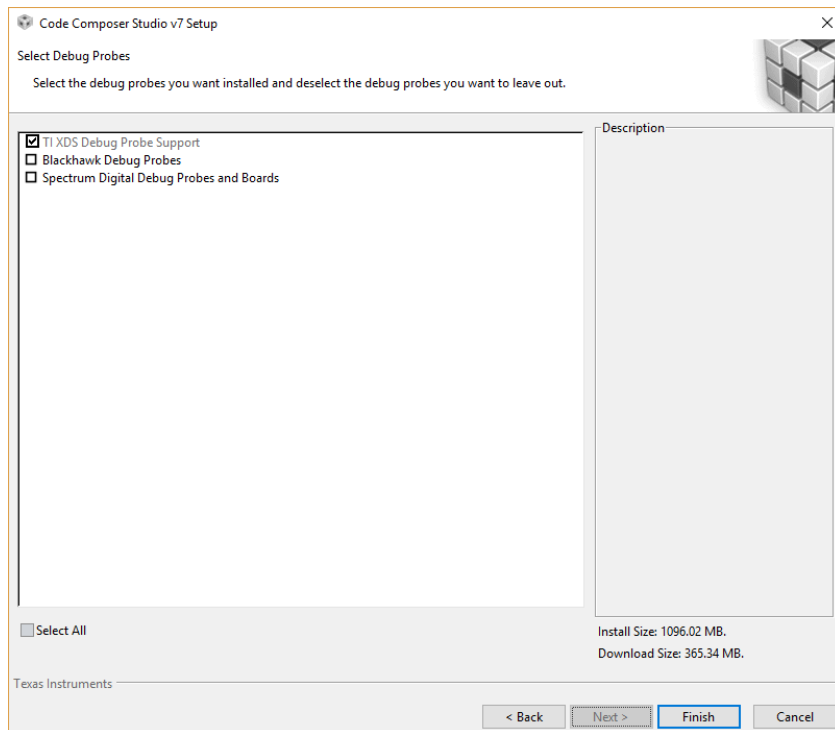


Figure 2: Selección de instalación de herramientas de debugging requeridas.

Una vez instalado CCStudio se debe instalar las herramientas de compilado, para esto se debe ingresar a CCS App Center mediante *View* → *CCS App Center*. Una vez dentro se debe seleccionar

C6000 Compiler (v7) y presionar *Install Software* en la esquina superior de CCS App Center, como se muestra en la Fig. 3.

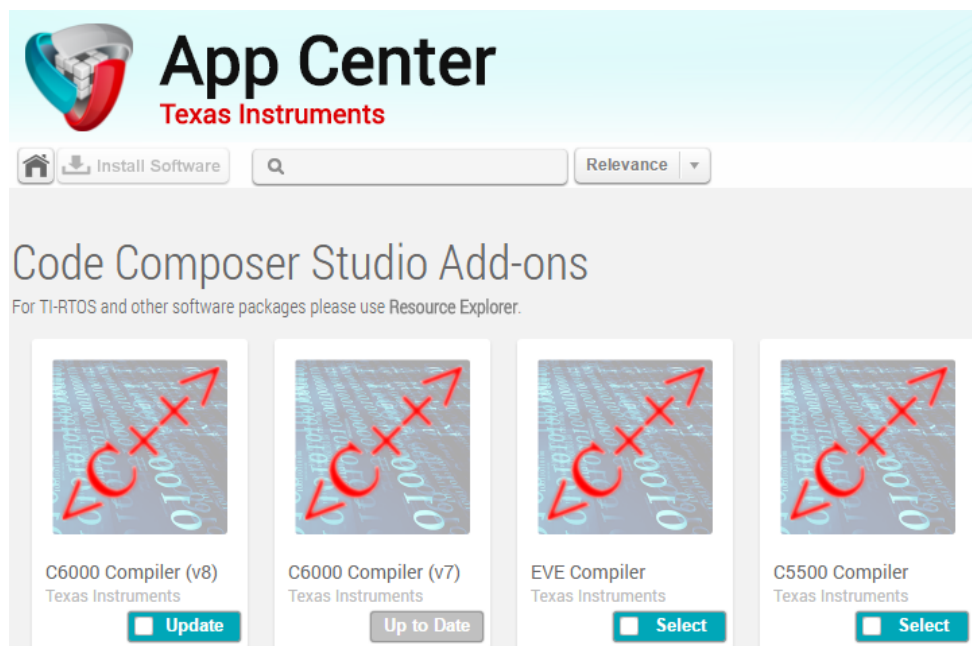


Figure 3: Instalación de compilador (se muestra ya instalado).

b.2 Librerías BSL y LCDK-AIC

Se debe extraer el Board Support Library y la librería AIC para LCDK en alguna ubicación de fácil acceso. Por ejemplo se puede extraer en

- D:\LCDK_C6748_codec_libraries\BSL, y
- D:\LCDK_C6748_codec_libraries\AIC.

Del BSL la carpeta fundamental es `CCSv4-2-4_with_SOM-M1_BSL_WS\DSP BSL\`, el resto pueden ser eliminadas. Del AIC la carpeta de mayor interés es la `LCDK_files_CCSv5\LCDK\L138_support\`, y en el resto de las carpetas se encuentran ejemplos útiles.

2 Creación de un proyecto

Una vez cumplido los pre-requisitos se puede continuar con la creación de un proyecto. Para esto se debe abrir el programa y seleccionar como workspace una carpeta destinada a contener los proyectos, pudiendo crear una en cualquier ubicación y con un nombre “luminso”.

En el laboratorio se ha configurado como workspace la misma carpeta de los proyectos de la librería BSL.

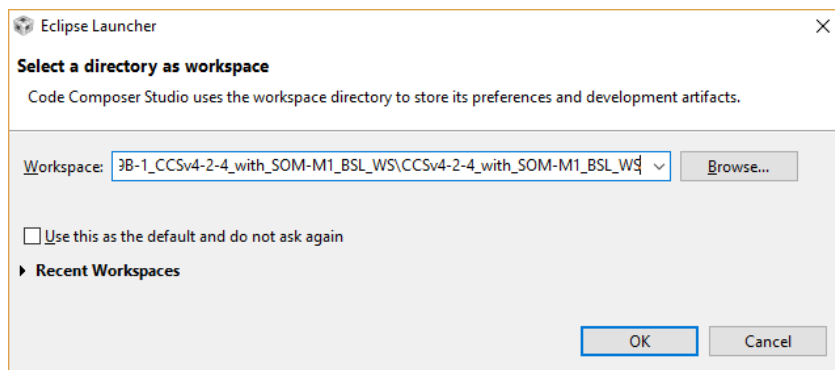


Figure 4: Selección del workspace para el laboratorio DSP.

Luego se debe crear un nuevo proyecto, para esto en *View* → *Project Explorer* se hace click derecho y *New Project*. En la ventana que se abrirá seleccionar *Code Composer Studio* → *CCS Project*.

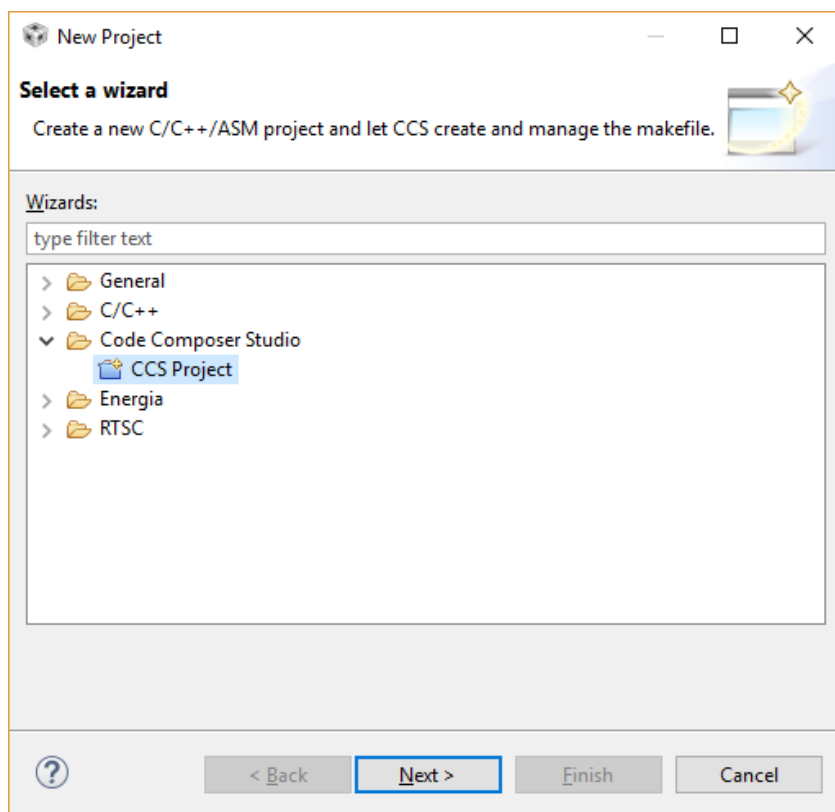


Figure 5: Selección de tipo de Proyecto.

En la siguiente ventana se debe elegir el nombre del proyecto (se creará una carpeta de proyecto con este nombre en el workspace), seleccionar el dispositivo con el que se trabajará (LCDKC6748), la herramienta de debugging (Texas Instrument XDS100v3 USB Debug Probe) y el compilador (TI v7.4.24 a la fecha de escritura). Mientras esté marcada la opción *Use default location*, el proyecto se guardará en el workspace previamente indicado.

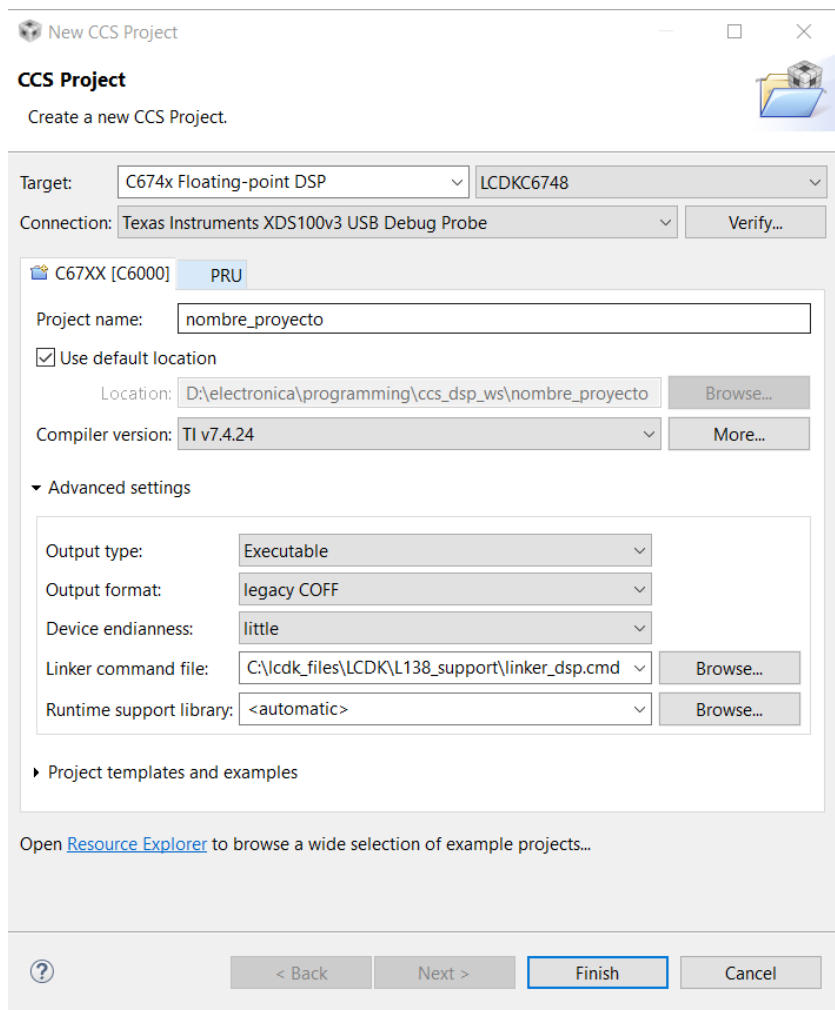


Figure 6: Opciones de Proyecto.

En la misma ventana, en *Advanced settings* es necesario configurar *Output type* como *Executable*, como *Output format* elegir *legacy COFF*, y seleccionar el *linker command file* desde la carpeta *L138_support* dondequiera se haya descomprimido.

a Enlace de librería compilada del BSL

Para indicar al compilador la ruta de búsqueda de los header files, se ingresará a las propiedades de éste, para esto se debe hacer click derecho en el proyecto en la ventana *Project Explorer* y elegir *Properties*. Una vez allí ingresará a *Build* → *C6000 Compiler* → *Include Options* y hacer click en el icono con un signo de suma de color verde en la sección *Add dir to #include search path*. En la ventana que se abrirá elegirá *Workspace...*, buscar *DSP BSL* → *inc* y aceptar.

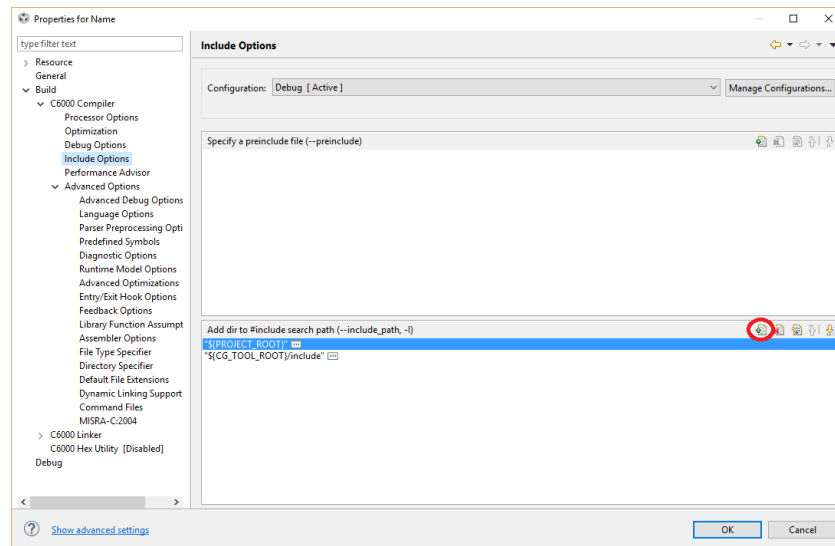


Figure 7: Botón para agregar carpeta de búsqueda indicado con un círculo rojo.

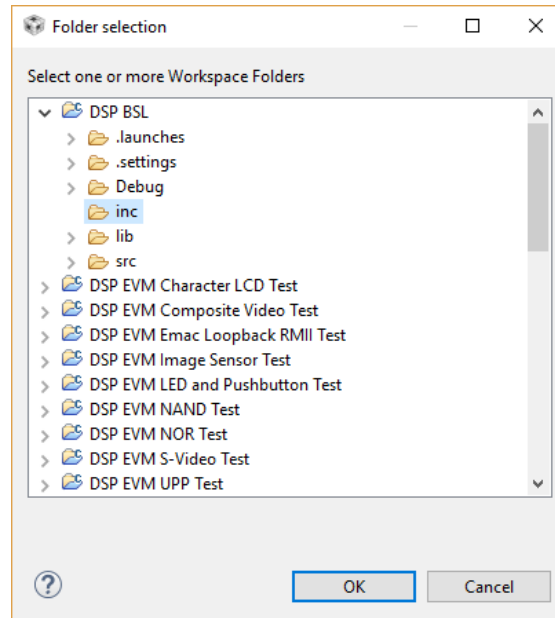


Figure 8: Selección de carpeta correspondiente.

Luego se debe hacer el vínculo con la librería correspondiente, para esto ingresar a *Build* → *C6000 Linker* → *File Search Path*. En la sección *Include library file or command file as input* hacer click en el botón con el signo de suma de color verde, seleccionar *Workspace...* y buscar *DSP BSL* → *lib* → *evmomapl138_bsl.lib*.

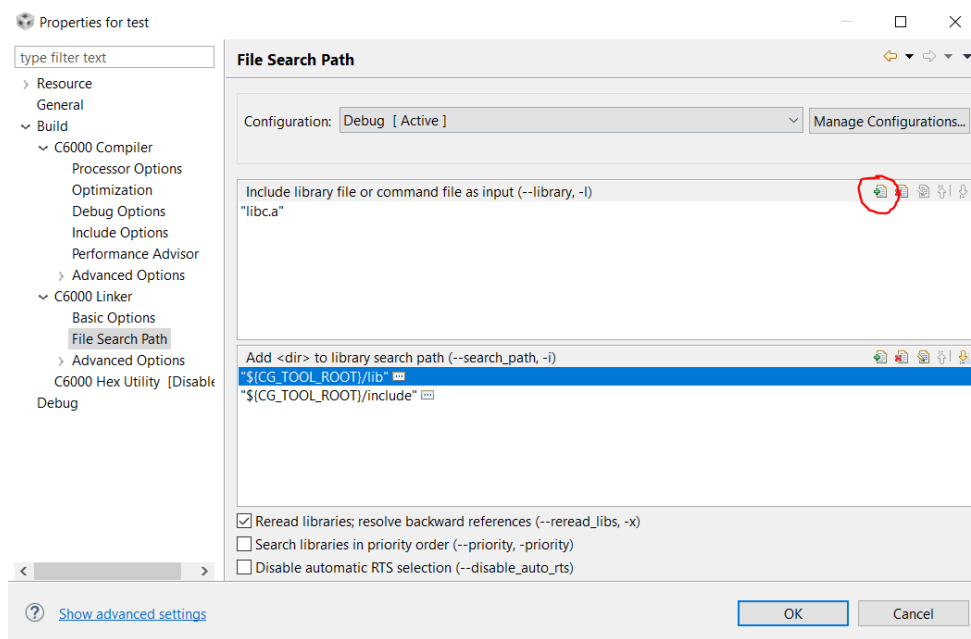


Figure 9: Botón para agregar librería indicado con un círculo rojo.

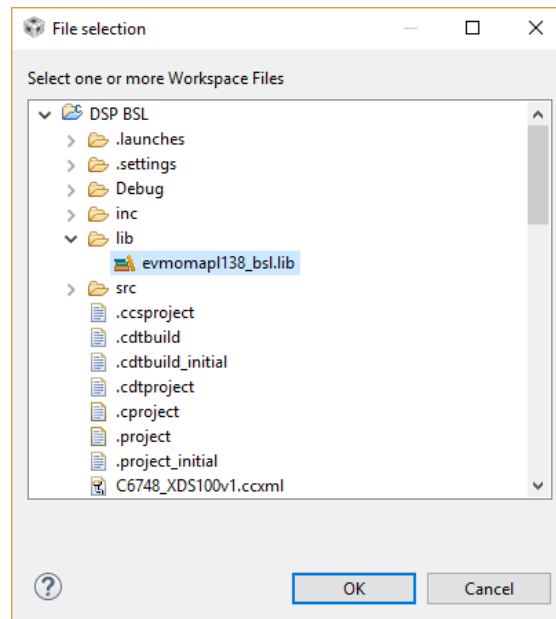


Figure 10: Selección de librería.

b Inclusión de archivos fuente AIC

Para poder utilizar las funciones necesarias para inicializar el Codec de audio y obtener input y output a partir de éste se debe agregar los archivos necesarios al set de compilación. Para esto hacer click derecho en el proyecto y seleccionar *Add files...* y navegar a la carpeta *LCDK_files_CCSv5\ LCDK\ L138_support*, seleccionar los archivos:

- *L138_LCDK_aic3106_init.c*,
- *L138_LCDK_aic3106_init.h*, y
- *vector_intr.asm* (en ocasiones puede ser necesario *vector_poll.asm* a cambio dependiendo si se trabajará con polling en vez de interrupciones).

c Código de ejemplo y prueba simple

A continuación se muestra un ejemplo de código simple que genera una señal sinusoidal en la salida izquierda. Para compilar basta con hacer click derecho en el proyecto y presionar *Build Project* (también se encuentra en el menú *Project*).

```
#include "L138_LCDK_aic3106_init.h"
#define LOOPLength 48
#define BUFLength 256

int16_t sine_table[LOOPLength] =
{0, 1305, 2588, 3827, 5000, 6088, 7071, 7934,
 8660, 9239, 9659, 9914, 10000, 9914, 9659, 9239,
 8660, 7934, 7071, 6088, 5000, 3827, 2588, 1305,
 0, -1305, -2588, -3827, -5000, -6088, -7071, -7934,
 -8660, -9239, -9659, -9914, -10000, -9914, -9659, -9239,
 -8660, -7934, -7071, -6088, -5000, -3827, -2588, -1305};

int16_t sine_ptr = 0; // pointer into lookup table

int32_t buffer[BUFLength];
int16_t buf_ptr = 0;

interrupt void interrupt4(void) // interrupt service routine
{
    int16_t sample;

    sample = sine_table[sine_ptr]; // read sample from table
    output_left_sample(sample);    // output sample
    sine_ptr = (sine_ptr+1)%LOOPLength; // increment table index
    buffer[buf_ptr] = (int32_t)(sample); // store sample in buffer
    buf_ptr = (buf_ptr+1)%BUFLength; // increment buffer index
    return;
}
```

```
int main(void)
{
    L138_initialise_intr(FS_16000_HZ,ADC_GAIN_ODB,DAC_ATTEN_ODB,LCDK_MIC_INPUT);
    while(1);
}
```

3 Conectando la DSP y ejecutando código

Primero se debe conectar el dispositivo emulador JTAG (XDS100V3) a la DSP; luego conectar el cable USB del JTAG al computador host, y finalmente conectar la alimentación del DSP.

Una vez que la tarjeta este alimentada será posible entrar al modo *Debug* del CCS, en el menú *Run* (presionar F11 o presionar el ícono de acceso directo en la barra de herramientas de debugging). Luego de unos segundos, CCS se debe conectar el dispositivo, cargarle el archivo ejecutable en memoria RAM, bootear el DSP y correr la rutina `_cin00()` para quedar el contador de programa al comienzo de la aplicación del `main.c`. Aparecerá una nueva ventana llamada *Debug* donde aparecerá el emulador JTAG usado y el dispositivo con sus núcleos.

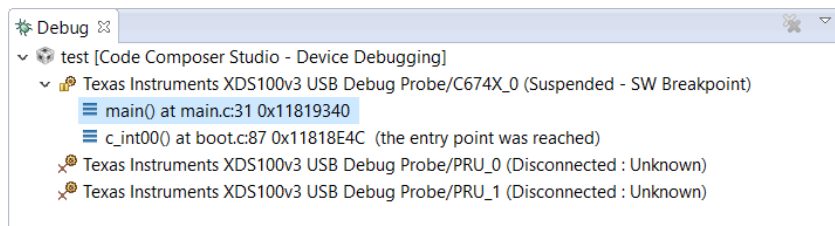


Figure 11: Conectar DSP a computador host.

4 Notas Finales

Tener las siguientes consideraciones al trabajar con la DSP y las instrucciones dadas en esta guía.

- Si el programa requiere utilizar arreglos grandes que ocupen más memoria de la disponible en la DSP incluir la siguiente línea de código `#pragma DATA_SECTION(name, ".EXT_RAM")`, donde `name` corresponde al nombre del arreglo a relocalizar.
- Para ver el nivel de uso de las memorias internas del DSP, se puede ver el resumen inicial de los segmentos de memoria en el archivo `<nombre_proyecto>.map` en la carpeta *Debug* del proyecto.
- Tener cuidado al utilizar las opciones *Suspend* y *Resume* en la ejecución del programa ya que no funcionan correctamente cuando se usan interrupciones en el proyecto.
- No desconectar cables mientras el programa está en funcionamiento, si se requiere cambiar algún cable de audio, conectar osciloscopio o generador de señales, primero se debe detener el programa con *Stop*.