



**Universidad Nacional Autónoma de México**  
**Facultad de Ingeniería**



---

## **Configuración de la conexión de LaunchPad Piccolo XL-F280049C a Code Composer Studio**

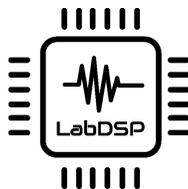
M.I. Larry H. Escobar Salguero  
M.I. Carlos I. García Sánchez

---

Tutorial

Laboratorio de Procesamiento Digital de Señales

[odin.fi-b.unam.mx/labdsp/](http://odin.fi-b.unam.mx/labdsp/)



División de Ingeniería Eléctrica  
Departamento de Procesamiento Digital de Señales

Ciudad Universitaria, Octubre 2019

## Resumen

En este tutorial se describe el procedimiento para configurar la conexión de la tarjeta de desarrollo LaunchPad Piccolo F28004x al entorno de desarrollo integrado Code Composer Studio versión 9, ambos pertenecientes a la compañía *Texas Instruments* (T.I.) y para verificar que el proceso se incluye la creación de un proyecto para hacer parpadear los LEDs de usuario de la tarjeta.

A diferencia de otros sistemas de prueba de esta misma empresa, para trabajar con la plataforma **launchxl-F280049C** es necesario realizar algunos pasos extras se especifican en este documento.

## Para citar este documento...

Larry Escobar, Carlos I. García, Conexión a Code Composer Studio del LaunchPad Piccolo XL-F280049C, tutorial. Universidad Nacional Autónoma de México, Facultad de Ingeniería, División de Ingeniería Eléctrica, depto. de procesamiento digital de señales. Ciudad Universitaria, Coyoacán, Ciudad de México, Octubre 2019.

```
@techreport{confXDS110F280049C,  
author = {Larry Escobar, Carlos García},  
title = {Conexión a Code Composer Studio del LaunchPad Piccolo XL-F280049C},  
institution = {Universidad Nacional Autónoma de México,  
Facultad de Ingeniería, División de Ingeniería Eléctrica,  
Departamento de Procesamiento Digital de Señales},  
address = {Ciudad Universitaria, Coyoacán, Ciudad de México},  
year = {2019},  
month = {octubre},  
url = {odin.fi-b.unam.mx/labdsp/}  
}
```

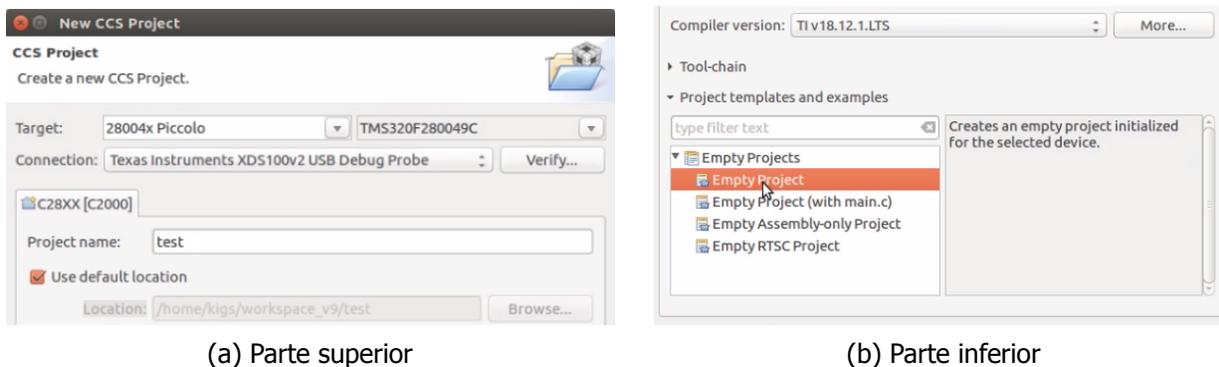
## 1 Introducción

A continuación se describe el procedimiento a seguir para configurar la conexión JTAG adecuada, para conectar y comunicar la tarjeta de desarrollo **launchxl-F280049C** con el entorno de desarrollo Code Composer Studio (CCS) versión nueve, creado por la compañía Texas Instruments (T.I.).

La configuración que se explica en este documento, se realiza después de crear un proyecto nuevo en CCS, de acuerdo a la metodología presentada en [1] para la versión seis, que técnicamente es igual para la versión ocho y nueve como se describe en [2]. Por esta razón, en la siguiente sección solo se presenta un resumen del proceso especificado para trabajar con la tarjeta launchxl-F280049C.

## 2 Crear un proyecto nuevo

La tarjeta de desarrollo mencionada en la Sección 1 tiene el dispositivo **TMS320F280049C**, catalogado por T.I. como parte de la familia **28004x Piccolo**. Por esta razón, los proyectos que se vayan a implementar en la plataforma launchxl-F280049C, se deben crear seleccionando la familia mencionada y la matrícula del dispositivo en la ventana `Proyecto nuevo CCS`, como se muestra en la figura 1a.



(a) Parte superior

(b) Parte inferior

Figura 1 Ventana de configuración para crear un proyecto nuevo.

En la opción `Connection` se debe **escoger alguno de los protocolos** de comunicación que aparecen en el listado, **sin importar cual sea** porque ninguno de ellos permite asociar la tarjeta de desarrollo a CCS y por la misma razón no se debe de hacer una prueba de vinculación con el botón `Verify`. En la figura 1a se muestra que para este escrito se eligió la conexión XDS 100 v2.

Para concluir la creación del proyecto, se selecciona alguna de las opciones de las plantillas de proyecto mostradas en la figura 1b, por ejemplo, en la imagen mencionada se observa que se optó por seleccionar *proyecto vacío*.

### 3 Configuración de la conexión XDS110 USB

Después de crear el proyecto con la configuración descrita en la Sección 2, en la ventana *Explorador de proyecto* (que se puede abrir desde el menú *Ver* o *View*) se resalta la carpeta que CCS crea en color naranja y con la leyenda **Active - Debug**, como se muestra en la figura 2a. La finalidad de este directorio es organizar los archivos necesarios para compilar, así como todos aquellos que puedan integrar el programa.

Al dar clic con el cursor al triángulo ubicado antes del nombre de la carpeta, se visualiza la estructura del directorio como se presenta en la figura 2b. Dentro de la carpeta **targetConfigs** se debe encontrar el archivo **TMS320F280049C.ccxml**, el cual contiene el conjunto de parámetros de configuración esenciales para conectar, probar y grabar códigos en los dispositivos y tarjetas soportados por CCS.

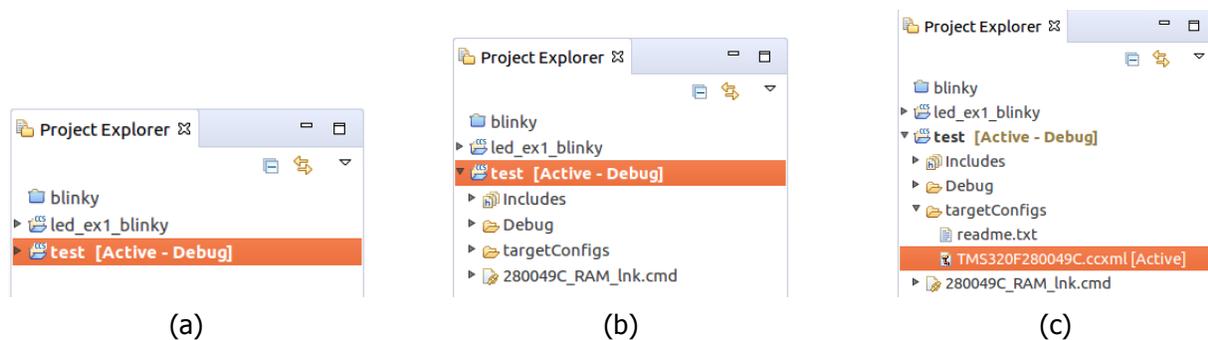


Figura 2 Estructura de archivos del proyecto nuevo.

Para visualizar este archivo dentro de CCS, se debe seleccionar el documento con el cursor, como se muestra en la figura 2c y dar doble clic sobre el nombre para abrir la ventana presentada en la figura 3 y proceder a reconfigurar adecuadamente dicho archivo para los proyectos que usen la tarjeta launchxl-F280049C.

La ventana mostrada en la figura 3 cuenta con tres pestañas ubicadas en su parte inferior izquierda (*Basic*, *Advanced* y *Source*), de éstas inicialmente se presenta la configuración básica, en donde se debe observar en la opción *Connection* el protocolo de comunicación seleccionado al crear el proyecto.

Dentro del menú de opciones de *Connection*, se encuentran otras alternativas que no estaban presentes en la ventana *Proyecto nuevo* CCS. Para conectar la tarjeta launchxl-F280049C se debe seleccionar la opción **XDS110 USB Debug Probe** para sustituir a la elegida en la Sección 2.

Después, se debe corroborar que se encuentre seleccionado el dispositivo declarado al crear el proyecto, indicado en la Sección 2. Para hacer esta verificación se puede escribir en la ventana de la opción *Board or Device* parte de la matrícula del dispositivo, como se observa en la figura 4 y verificar que esta seleccionado el **TMS320F280049C**.

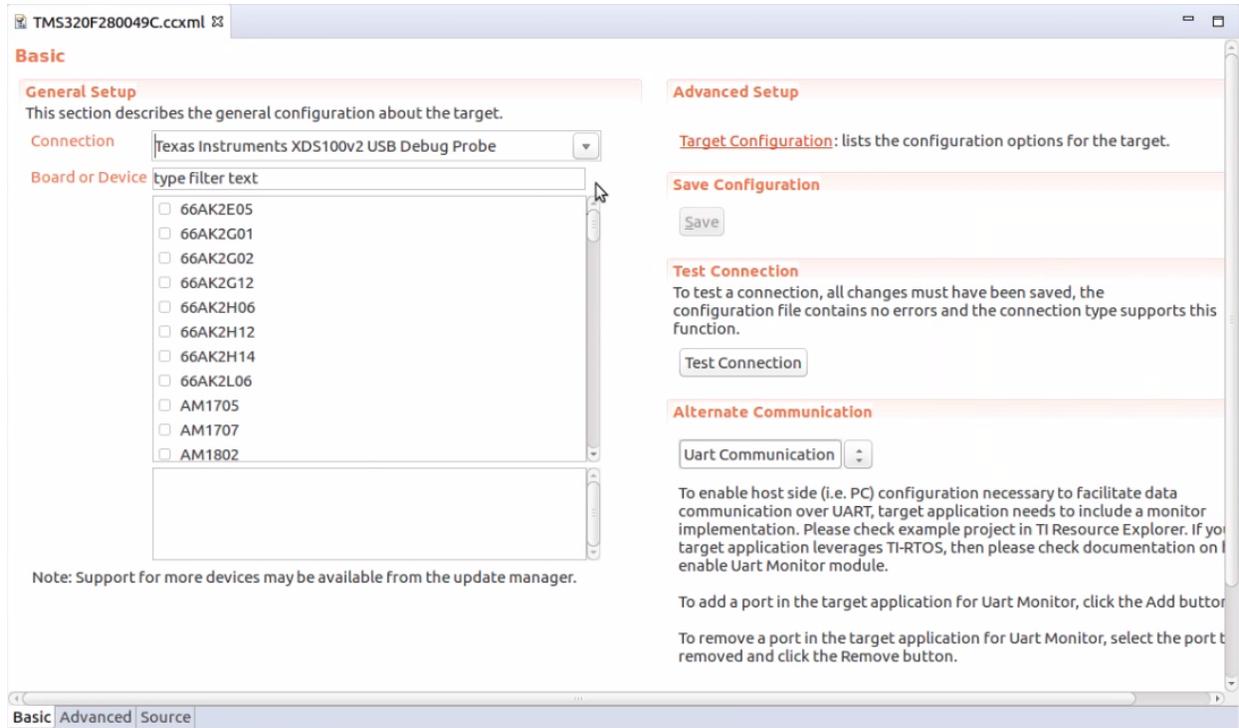


Figura 3 Conjunto de opciones de la configuración básica del archivo TMS320F280049C.ccxml.

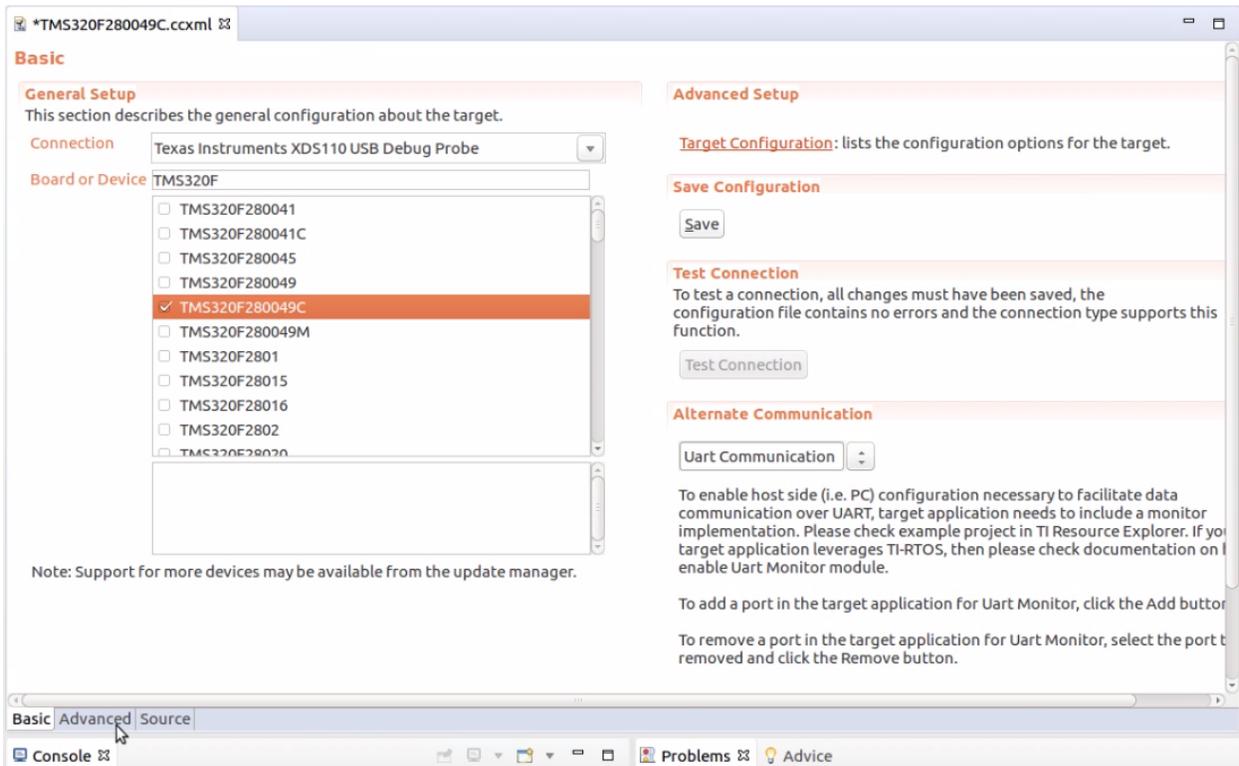


Figura 4 Selección de la conexión XDS110 USB y del dispositivo a conectar.

Posteriormente, se debe acceder a las opciones avanzadas dando clic con el cursor a la pestaña Avanzada o Advanced, ubicada en la parte inferior izquierda de la ventana que visualiza el archivo `TMS320F280049C.ccxml`, como se indica en la figura 4.

En la pestaña Advanced mostrada en la figura 5, se debe seleccionar con el cursor el protocolo de comunicación asignado *Texas Instruments XDS110 USB Debug Probe*, en la parte superior de la subventana All Connections. Después, en el conjunto de parámetros Connection Properties se debe seleccionar en JTAG/SWD/cJTAG Mode la opción **modo avanzado de 2 terminales cJTAG(1149.7)**, como se indica den la figura 5. Después de hacer esta elección se desplegarán otros campos de configuración que **no se deben modificar**.

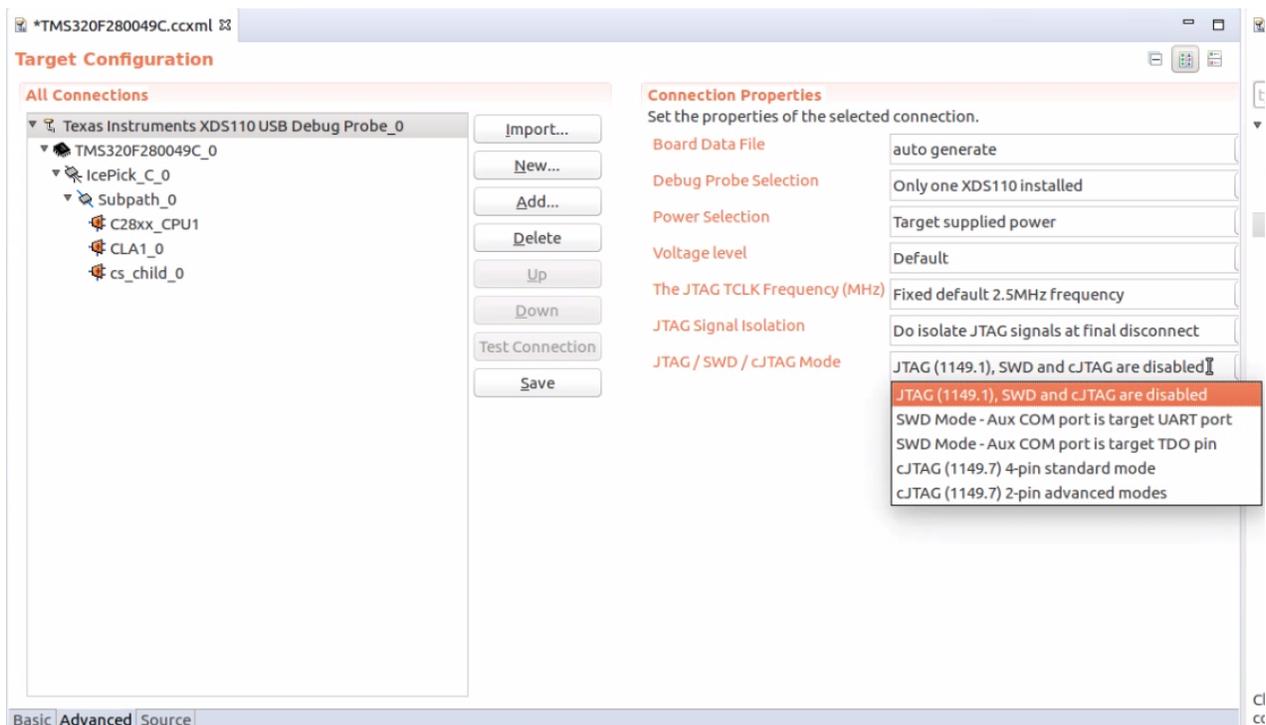


Figura 5 Selección de la configuración del cJTAG 1149.7 de 2 terminales.

Para guardar los cambios hechos en el archivo `TMS320F280049C.ccxml`, se debe dar clic al botón Save o Guardar que aparece en la parte central de la pestaña Advanced. Después de hacer esta acción se deshabilitan algunos botones de esa columna y se habilita el de Test Connection, que permite hacer una prueba de conexión usando los cambios realizados.

Si la configuración descrita fue realizada correctamente, la prueba de conexión debe ser exitosa hecho que se comprueba al visualizar la parte final de la ventana Test Connection (que aparece después de dar clic al botón Test Connection) y verificar que debe ser prácticamente igual a la que se muestra en la figura 6, donde debe observarse que no existieron fallas de conexión.

Y después de hacer este breve procedimiento, ya se puede trabajar en modo Debug o grabar diferentes programas en la tarjeta launchxl-F280049C.

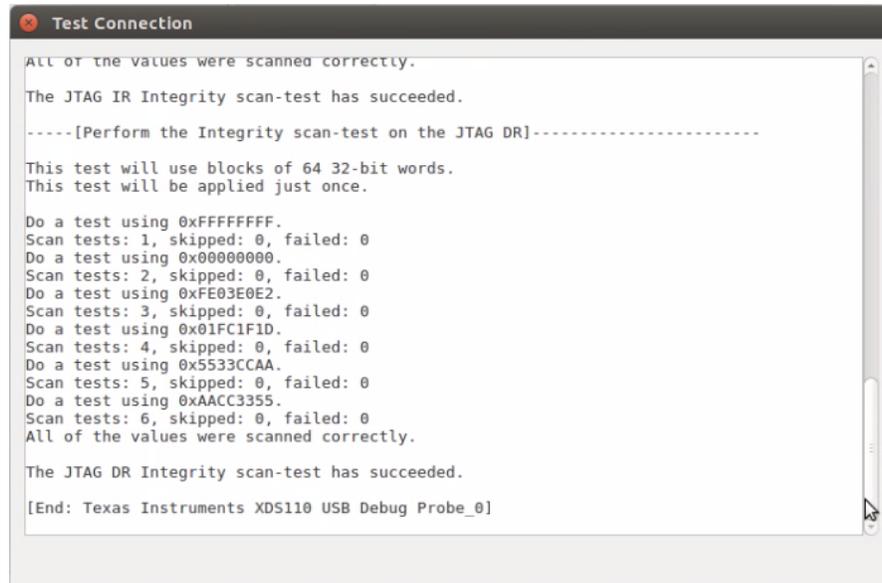


Figura 6 Reporte de conexión exitosa de la tarjeta launchxl-F280049C con CCS.

## 4 Reutilización de archivo TMS320F280049C.ccxml

El procedimiento descrito en las Secciones 2 y 3 puede evitar hacerse para cada proyecto nuevo a implementarse en la tarjeta launchxl-F280049C, siempre y cuando se cuente con un archivo TMS320F280049C.ccxml configurado adecuadamente y se conozca su ubicación en la computadora. A continuación se describe una manera de reutilizar el archivo \*.ccxml para proyectos nuevos, realizando todas las operaciones dentro de CCS.

Después de crear un proyecto nuevo de acuerdo a la Sección 2, en la ventana Explorador de Proyecto se debe visualizar la estructura de archivos del proyecto creado, como se muestra en la figura 7a. Posteriormente, colocando el cursor sobre el nombre de la carpeta del proyecto nuevo, se da clic derecho para desplegar el menú de opciones que se presenta en la figura 7b.

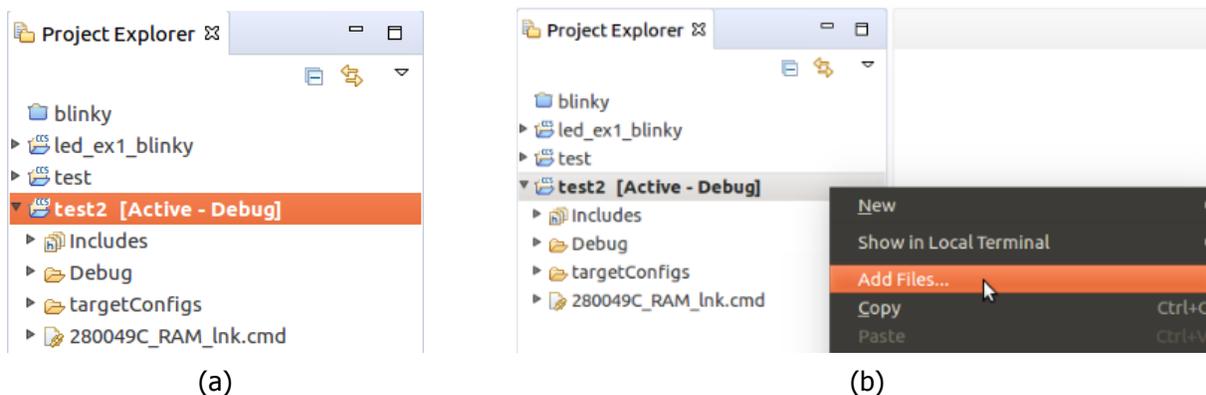


Figura 7 Agregar archivo \*.ccxml a un proyecto nuevo desde la ventana explorador de proyecto.

De esta lista de acciones se debe seleccionar la opción `Agregar Archivos...` y enseguida se presenta una ventana del manejador de archivos, como se muestra en la figura 8a, en la cual se debe buscar y seleccionar el archivo `TMS320F280049C.ccxml` configurado adecuadamente, que en este caso se guardó en la carpeta `targetConfigs` del anterior proyecto.

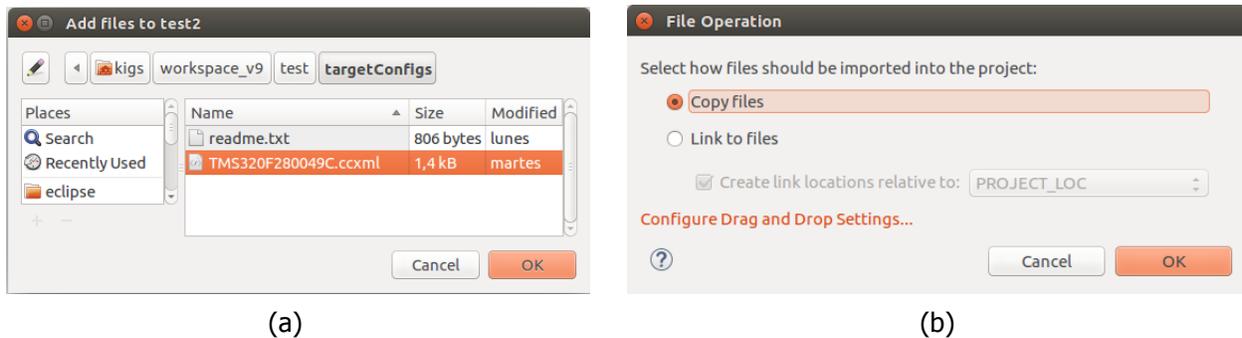


Figura 8 Ventanas para agregar archivos al proyecto nuevo.

Posteriormente, se debe dar clic izquierdo al botón `OK` de la ventana `Agregar Archivos...` para visualizar la ventana `Operación del archivo` (figura 8b), en donde la IDE pregunta si se quiere copiar el documento fuente o crear un archivo de vinculación. De estas opciones se sugiere elegir **copiar** y confirmar la operación dando clic izquierdo al botón `OK` de esta ventana.

Después de hacer las acciones anteriores, se debe visualizar en la ventana `Explorador de Proyecto` el archivo `*.ccxml` dentro de la carpeta del proyecto nuevo, como se puede observar en la figura 9a.

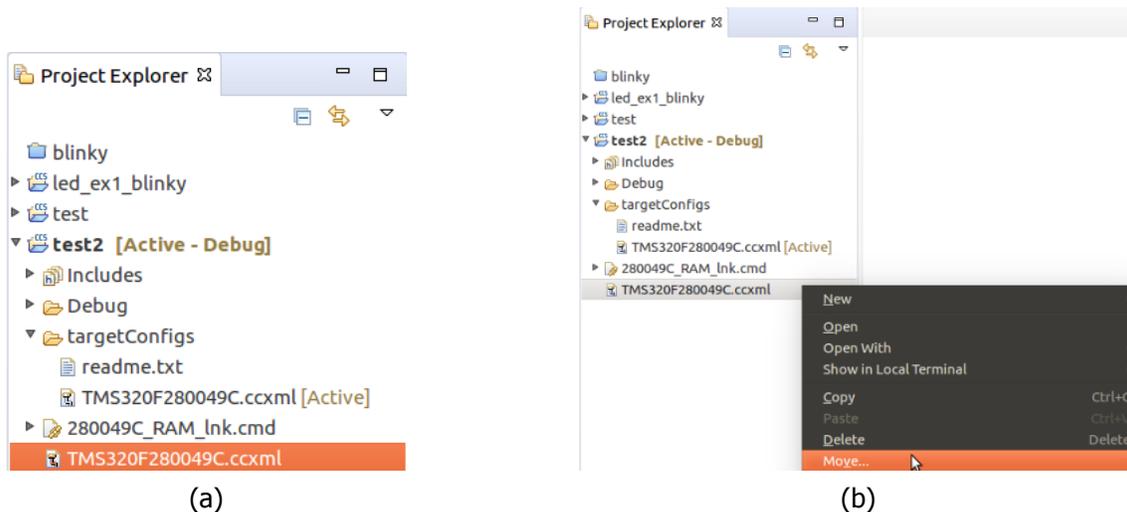
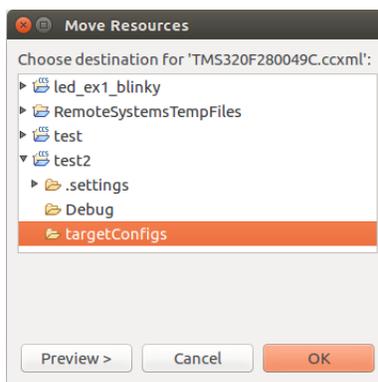


Figura 9 Ventanas para agregar archivos al proyecto nuevo.

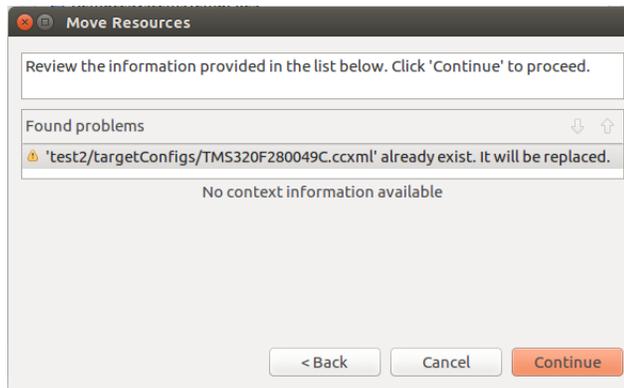
Y el último paso que resta por hacer consiste en mover el archivo `TMS320F280049C.ccxml` a la carpeta `targetConfigs` de nuestro proyecto nuevo, acción que se puede realizar de dos formas:

### 1. Utilizando las herramientas de la IDE

Este modo consiste en dar clic derecho sobre el archivo que acabamos de agregar, para desplegar un menú en el cual se debe seleccionar la opción `Move...`. Después de hacerlo se presenta la ventana mostrada en la figura 10a, en donde se debe seleccionar la carpeta destino del archivo y confirmar la acción al presionar el botón `OK`.



(a)



(b)

Figura 10 Ventanas relacionadas al cambio de ubicación del archivo \*.ccxml.

### 2. De forma gráfica en la ventana Explorador de Proyecto

Con el cursor se debe seleccionar el archivo que queremos mover, dar clic izquierdo sobre el objeto y sin soltar el botón, arrastrar con el mouse el archivo hasta la carpeta deseada.

En ambos casos se nos presentará una ventana (como la que se muestra en la figura 10b) que nos dirá que ya existe por lo que pregunta si queremos sobrescribirlo, acción que se debe confirmar para que la tarjeta launchxl-F280049C logre conectarse a CCS.

## 5 Síntesis de configuración del archivo .ccxml

A continuación se describe un resumen del proceso explicado en las Secciones 2 y 3, con la finalidad de proponer una guía de apoyo de los pasos concretos.

### 1. Crear un proyecto nuevo, seleccionando los siguientes parámetros:

- a) Como `Target` la familia **28004x Piccolo**
- b) El dispositivo **TMS320F280049C**
- c) Y cualquier protocolo de comunicación, en el campo `Connection` y **no verificar conexión**.

### 2. Visualizar archivo TMS320F280049C.ccxml

- a) En la ventana `Explorador de proyecto`, visualizar la estructura de archivos del proyecto creado como se muestra en la figura 2b

- b) Ubicar la carpeta `targetConfigs`, mostrada en la figura 2c y abrir el archivo `.ccxml` que se encuentra adentro de este directorio, dando doble clic sobre el nombre del archivo.

### 3. Cambiar configuración básica del archivo `TMS320F280049C.ccxml`

- a) Cambiar la opción que tenga el campo `Connection` por **XDS110 USB Debug Probe**
- b) Verificar que el parámetro `Board` or `Device` tenga seleccionada la opción **TMS320F280049C**.

La configuración debe ser como se muestra en la figura 4.

### 4. Selección de la modalidad de JTAG en configuración avanzada

- a) Entrar a la pestaña de configuración avanzada, mostrada en la figura 5 ubicada en la parte inferior izquierda de la ventana `TMS320F280049C.ccxml`, como se indica con el cursor en la figura 4
- b) Seleccionar en el campo `JTAG/SWD/cJTAG Mode` la opción **cJTAG(1149.7) 2-pin advanced modes**
- c) Dar clic en el botón `Save` o `Guardar` para escribir los cambios realizados al archivo `TMS320F280049C.ccxml`.

### 5. Probar conexión

- a) Hacer la prueba de conexión dando clic al botón `Test Connection` que se encuentra en la parte central de la pestaña de configuración avanzada
- b) Revisar que la prueba de conexión sea exitosa y sin fallas, observando la parte final del reporte que se visualiza en la ventana `Test Connection`. La información del reporte debe ser semejante a la mostrada en la figura 6.

## 6 Programa de prueba

Si la configuración que se describió en las secciones anteriores fue realizada correctamente, es posible probar el funcionamiento de la tarjeta con algún código. Para finalizar este reporte técnico, a continuación se presenta un programa en lenguaje ensamblador que hace parpadear los LEDs de usuario disponibles en la tarjeta launchxl-F280049C, los cuales están conectados a los GPIO 23 y 34.

Para incorporar este código a un proyecto nuevo (creado como se especificó previamente en este documento), es necesario crear un archivo, como se muestra en la figura 11a o seleccionando la opción `File` del mismo menú. El nombre del archivo es libre, pero debe tener la terminación `.asm` como se observa en la figura 11b.

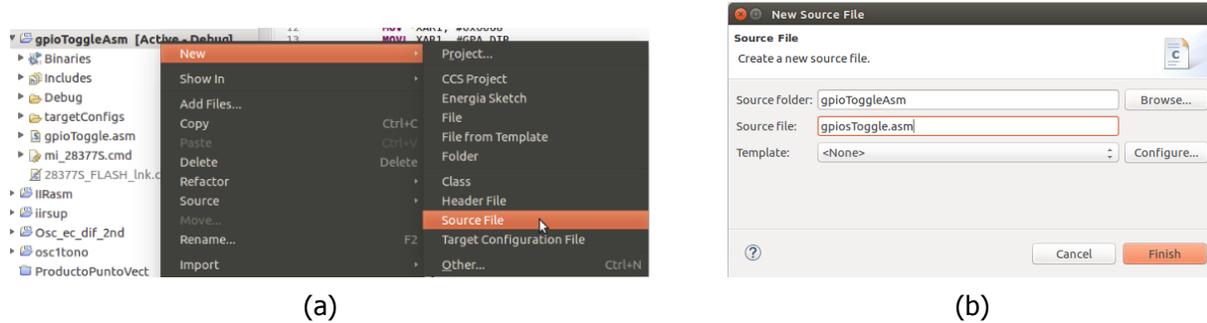


Figura 11 Pasos para crear un archivo de código dentro de un proyecto de CCS.

Posteriormente, se debe transcribir el programa que se presenta a continuación en el archivo creado y si todo fue escrito y configurado adecuadamente, el proyecto debe compilar y poder ejecutarse en la tarjeta hecho que se verifica al observar que los LEDs de usuario parpadean de forma intercalada a la misma frecuencia.

```

1      .global _c_int00
2
3  ; Direcciones de registros para configurar el hardware a ocupar
4  WDCT      .set 0x7029      ; WatchDog timer
5  GPA_DIR   .set 0x7C0A      ; Selección de pines GPIOs del puerto A para
6                                     ; funcionar como entradas o salidas
7  GPB_DIR   .set 0x7C4A      ; Selección de pines GPIOs del puerto B para
8                                     ; funcionar como entradas o salidas
9  GPA_AMSEL .set 0x7C14      ; Permite seleccionar el modo de uso de los
10                                     ; GPIOs 22 y 23 (análogo o digital)
11 GPA_TOGG  .set 0x7F06      ; Conmuta el estado digital de los GPIOA
12 GPB_TOGG  .set 0x7F0E      ; Conmuta el estado digital de los GPIOB
13
14 GPIO_23   .set 0x0080      ; No. hexadecimal para usar GPIO23 como salida
15 GPIO_34   .set 0x0004      ; No. hexadecimal para usar GPIO34 como salida
16
17      .text                  ; Inicio de programa
18 _c_int00
19      EALLOW                  ; Habilita escritura en registros protegidos
20      MOVL XAR1, #WDCT        ; Apunta al registro 0x7029
21      MOV  *XAR1, #0x0068     ; Escribir el no. hexa para desactivar WDCT
22
23 ; Selección de modo digital del GPIO23
24      ZAPA                    ; Escribe cero en los registros ACC y P
25      MOVL XAR3, #GPA_AMSEL   ; Apunta al registro 0x7C14
26      MOVL *XAR3, ACC         ; Escribe ceros en el registro GPA_AMSEL
27
28 ; Selecciona pines como salidas
29      MOVL XAR3, #GPA_DIR     ; Apunta al registro 0x7C0A
30      MOV  AH, #GPIO_23       ; Escribe 0x0080 en la parte alta de ACC
31      MOVL *XAR3, ACC         ; Habilita GPIO23 como salida
32
33      MOVL XAR3, #GPB_DIR     ; Apunta al registro 0x7C4A
34      MOV  *XAR3, #GPIO_34    ; Habilita GPIO34 como salida
35      EDIS                    ; Deshabilita escritura en registros protegidos
36
37      MOVL XAR7, #GPA_TOGG    ; Apunta al registro 0x7F06

```

```
38          MOVL XAR1, #GPB_TOGG      ; Apunta al registro 0x7F0E
39
40 CIC_A
41          MOVL XAR2, #0x0050      ; Valor máximo del contador A
42          MOVL *XAR7, ACC          ; Permutación del estado digital de GPIO23
43 CIC_B
44          MOVL XAR0, #0xFFFF      ; Valor máximo del contador B
45 CIC_C
46          NOP
47          NOP
48          BANZ CIC_C, AR0---       ; Ciclo de operaciones nulas hasta que
49                                     ; contador B sea igual a cero
50          NOP
51          BANZ CIC_B, AR2---       ; Ciclo hasta que contador A sea igual a cero
52
53          MOV *XAR1, #GPIO_34      ; Permutación del estado digital de GPIO34
54          LB CIC_A                  ; Salto para reiniciar contadores
55
56          .end                      ; Fin de programa
```

## Referencias

- [1] Luis Álvarez, Ivan Menendez y Larry Escobar, "Creación de un proyecto en CCS v6 para el TMS320F28377S," Reporte técnico, Universidad Nacional Autónoma de México, Facultad de Ingeniería, División de Ingeniería Eléctrica, Departamento de procesamiento digital de señales, Ciudad Universitaria, Coyoacán, Ciudad de México, 2016, [http://odin.fi-b.unam.mx/labdsp/files/PDSL/archivos/prac\\_2\\_delfino.pdf](http://odin.fi-b.unam.mx/labdsp/files/PDSL/archivos/prac_2_delfino.pdf).
- [2] Luis Álvarez, Larry Escobar, et al., *Manual de algoritmos y aplicaciones de procesamiento digital de señales, empleando la familia TMS320F2837xS*, Facultad de Ingeniería, Universidad Nacional Autónoma de México, Febrero 2019.