

SISTEMAS BASADOS EN MICROPROCESADOR
PRÁCTICA 1. ENTORNO DE PROGRAMACIÓN

Grado en Ingeniería Electrónica Industrial (3^{er} curso)

Curso 2025/2026

1 INTRODUCCIÓN

Code Composer Studio es un entorno integrado de desarrollo de aplicaciones multiplataforma de *Texas Instruments*. Permite, entre otras cosas, la escritura del código en ensamblador y/o C/C++, ensamblado de programas, depuración, así como acceder a multitud de recursos tales como documentación del fabricante, hojas de datos del micro (*datasheet*), notas de aplicación, ejemplos, librerías de funciones, y un largo etcétera.

A partir de la versión 7, este programa se distribuye libremente sin restricciones.

1.1 Objetivos

- Familiarizarse con el entorno de desarrollo *CCS v12*.
- Profundizar en el manejo del conjunto de instrucciones del *MSP430*.
- Comprender la forma en que el microcontrolador ejecuta los programas y prever su comportamiento.

2 ESTUDIO TEÓRICO

Para la realización de esta práctica no se requiere realizar un trabajo previo, salvo haber asimilado los conceptos de la arquitectura y lenguaje ensamblador explicados en teoría. En particular:

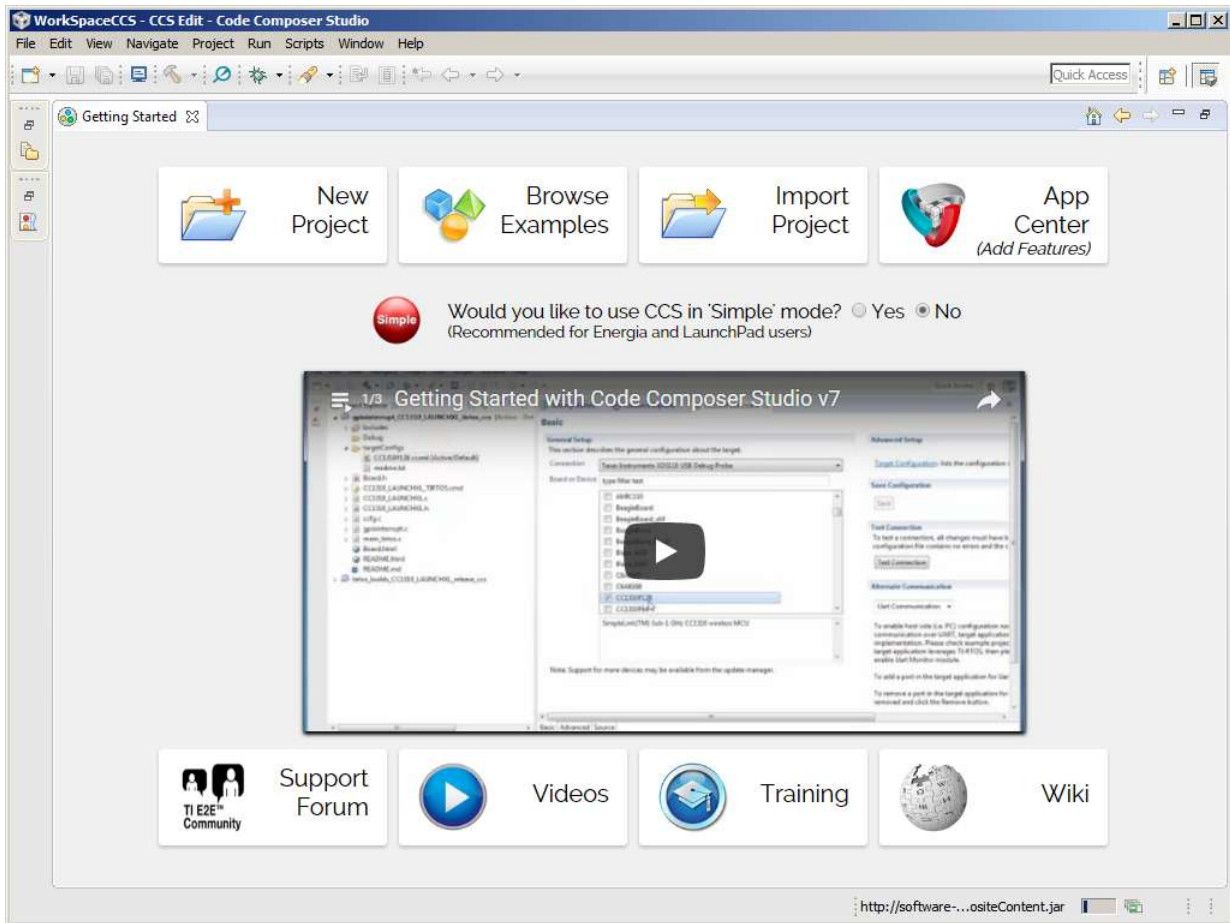
- Modelo del programador del MSP430: conjunto de registros, conjunto de instrucciones, modos de direccionamiento.
- Ensamblador del MSP430: formato de instrucciones, directivas de ensamblado.
- Sistema de interrupciones.
- Puertos de E/S digital.
- Anote aquí los puertos en los que se encuentra el led 1 (____) y el pulsador 1 (____).

3 ESTUDIO PRÁCTICO

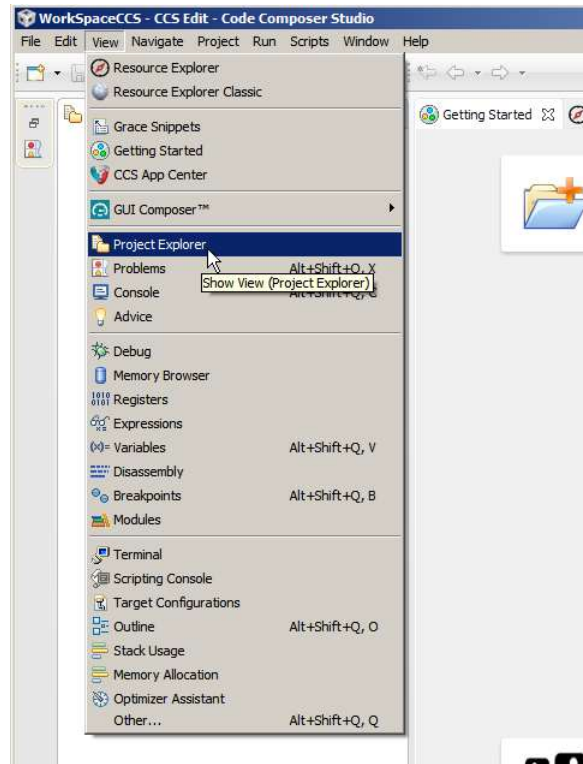
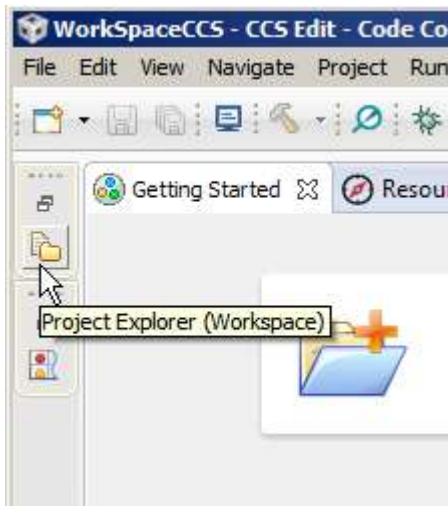
Abra el *Code Composer Studio* (CCS). Para ello utilice el icono del escritorio. 

3.1 Conociendo el entorno de desarrollo

Una vez abierto el programa, identifique la ventana *Getting Started*:



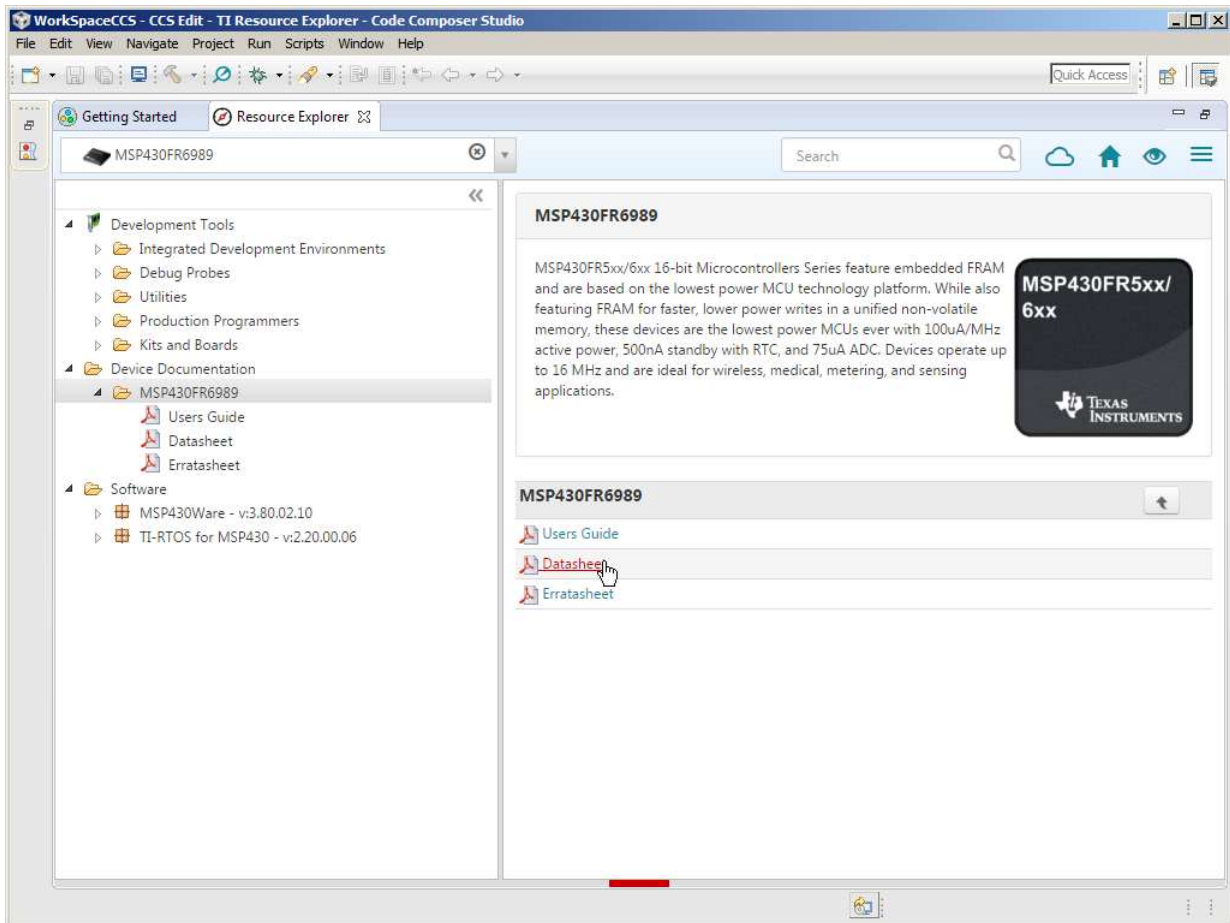
y el botón *Project Explorer*. Este *Explorador de Proyectos*, así como el resto de ventanas, también son accesibles desde el menú *View*. Preste atención a los dos iconos que hay arriba a la derecha. Son *CCS Edit* y *CCS Debug*. El entorno cambia automáticamente entre estos dos modos, pero es posible usar estos accesos directos para hacerlo manualmente. El primero se usa cuando se está editado el código. El segundo, en cambio, se usa cuando se está ejecutando el programa en la plataforma para poder depurarlo.



En la pantalla *Getting Started*, identifique el botón *Browse examples*. Al pinchar en él se abre el *Resource Explorer* (explorador de recursos) que da acceso a una amplia variedad de recursos que el fabricante pone a disposición de los desarrolladores. Debe tenerse en cuenta que estos recursos son remotos (*online*) y lentos en cargar¹.

1. TI cambia con frecuencia el explorador de recursos. Si no carga automáticamente y/o recibe un mensaje de error, siga las instrucciones de http://processors.wiki.ti.com/index.php/TIREX_Update para actualizarlo. El proceso puede tardar unos 3 minutos

Para simplificar la búsqueda, ponga el modelo del micro (*MSP430FR6989*) en la casilla *Select a Device or Board* y seleccione. Debido a la amplia diversidad de recursos y plataformas disponibles, es preferible escribir el modelo que buscarlo entre la lista.



Una vez cargados los recursos de nuestro microcontrolador (es un proceso lento que puede tardar), identifique la información disponible: manual de la familia, hoja de datos, hoja de errata, manual del launchpad, ejemplos,...

Despliegue *MSP430Ware/Devices/MSP430FR6989/Peripheral Examples/Assembly*. Puede ver varios ejemplos de código. Puede incluso importar el que quiera en el entorno aunque en esta sesión no lo haremos porque requiere instalar un paquete y lleva tiempo.

Los códigos de los ejemplos son accesibles pinchando en cada uno de ellos.

3.2 Ejemplo 1. Led parpadeante

- Seleccione y copie todo el texto de *msp430fr69xx_1.asm* (primero de la lista) en el portapapeles.
- En la barra de menú pulse *Project/New CCS Project...*
 - ♦ En *Target* ponga el modelo del micro si no aparece automáticamente.
 - ♦ En *Project name* ponga el nombre del proyecto, por ejemplo "P1.1".
 - ♦ En *Project Templates and Examples* seleccione *Empty Assembly-only Project*.
 - ♦ Pulse *Finish*.

- Sustituya todo el código del fichero recién abierto (*main.asm*) por el del portapapeles (seleccione todo y pulse Ctrl-V).
- Ensamble el programa. Para ello, pulse sobre el icono del martillo  o Ctrl-B (si es la primera vez desde la instalación, puede tardar varios minutos mientras se compilan las librerías).
- Una vez terminado, en el cuadro inferior derecho *Problems* aparecerá un resumen con los errores y avisos que tiene el código. En este caso aparece un *warning* (aviso) que se debe a un error conocido del procesador. No es crítico y no requiere hacer nada.
- Conecte el *Launchpad* al puerto USB del PC con el cable.
- Pulse sobre el icono del insecto  o pulse F11 (*debug*). El entorno cambia al modo *CCS Debug (Depuración)*. El programa se carga en el micro. Se muestra el código fuente y se resalta la primera instrucción a ejecutar.
- Pulse el icono *Play* o pulse F8 (*resume*). El led inferior izquierdo del Launchpad debe parpadear. **Avise al profesor.**
- Pulse el icono *Stop* o pulse Ctrl-F2 (*terminate*). El entorno vuelve al modo edición (*CCS Edit*).
- Haga doble click a la izquierda del número de línea que empieza por *MainLoop*. Eso activa un punto de ruptura (*breakpoint*). Cuando el procesador llegue a esa línea, se parará automáticamente y mostrará el estado del procesador en el CCS.
- Vuelva a depurar (F11 o ).
- Vuelva a ejecutar (pulsar F8 o icono de play). El programa se parará en la línea en la que se puso el punto de ruptura.
- En el cuadro superior derecho está la ventana de visualización de valores. Hay una pestaña para *Variables*, otra para *Expressions* y otra para *Register* (es posible que la ventana esté contraída y sólo se muestren los iconos en vertical, en el margen derecho). Pulse en *Registers*. Deslice la ventana para ver *Port_1_2*. Despléguelo pulsando sobre el signo >.
- Pulse sobre *Step Into* (F5 o icono de flecha amarilla). En el cuadro superior derecho se resaltan con fondo amarillo los registros que cambian. **¿Es correcto?**
- Deslice ahora esa ventana arriba y despliegue *Core Registers*. Deslice para que se vea R15.
- Pulse F5 (*Step Into*). R15 debe cambiar. **¿Está bien?**
- En este modo no se pueden ver R15 y P1OUT a la vez. Si desea hacerlo, pulse con el botón derecho del ratón sobre R15 y pulse en *Watch*. Se abrirá la pestaña *Expressions* del cuadro superior derecho y aparece el valor de R15. Repita el proceso con P1OUT. Puede seleccionar el formato de visualización seleccionando el registro y pinchando con el botón derecho del ratón sobre la columna *Value*. Ponga el R15 en decimal y P1OUT en binario.
- Pulse varias veces F5 (*Step Into*) y observe los cambios en los registros.
- Termine la depuración con Ctrl-F12 (*Terminate*).



- Modifique el programa para que el led parpadee el triple de rápido. No olvide quitar el breakpoint. Ensamble, pase a depuración y ejecute el programa. **Muestre el resultado al profesor.**

3.3 Programa de ejemplo 2. Led controlado por pulsador

- Repita los pasos 1 a 3 del ejemplo 1 con *msp430fr69xx_p1_01.asm* y llame al proyecto "P1.2".
- El programa enciende el led izquierdo del Launchpad y lo apaga cuando se pulsa el botón izquierdo. Ensamble  y pruébelo . ¿Funciona?
- Descubra el/los errores del programa. Ayúdese de lo aprendido sobre depuración en el ejemplo anterior. Consulte las documentaciones del *Launchpad* y del microcontrolador si es necesario en el *Resource Explorer*.
- Una vez que funcione, cambie la lógica del pulsador, para que el led se encienda cuando se pulse el botón y se apague en caso contrario. **Avise al profesor.**

3.4 Programa de ejemplo 3. Led controlado por pulsador (por interrupciones)

- Pulse con el botón derecho del ratón sobre el proyecto P1.2 y seleccione *Copiar* (o pulse Ctrl-C).
- Sobre el espacio vacío del *Project Explorer* pulse botón derecho de ratón/Pegar (o pulse Ctrl-V). En la ventana de diálogo, ponga como nombre "P1.3" y Acepte. Esta acción abre automáticamente *main.asm* en la ventana de edición.
- Modifique el programa para que se ejecute por interrupciones. Debe inicializar el vector `PORT1_VECTOR` con la dirección de la subrutina de interrupción (llámela `P1ISR` y siga el formato del vector de reset). Entre otras cosas, tendrá que configurar correctamente el P1 para que se permitan las interrupciones.
- Cuando lo consiga, **avise al profesor.**