Guía de instalación y puesta a punto de herramientas para el Laboratorio de Sistemas Embebidos para Tiempo Real

Esta guía busca orientar al estudiante en la instalación y puesta a punto de todas las herramientas necesarias para desarrollar el Laboratorio de Sistemas Embebidos para Tiempo Real. En particular, se cubren tres aspectos fundamentales

- 1) Instalación de Code Composer Studio
 - a) Guía de instalación para Windows 10/11
 - b) Guía de instalación para Ubuntu 18.04
 - c) Inicialización de CCS
- 2) Instalación de Git
 - a) Guía de instalación para Windows 10/11
 - b) Guía de instalación para Ubuntu 18.04
- 3) Puesta a punto del repositorio del curso
 - a) Creación del repositorio del curso
 - b) Primer push al repositorio del curso
 - c) Clonación del repositorio del curso
 - d) Creación de un nuevo proyecto de CCS

Los primeros dos puntos deben ser ejecutados por aquellos estudiantes que deseen utilizar sus propios PC para desarrollar el laboratorio. El paso 3 es obligatorio para todos los grupos.

1. Instalación de Code Composer Studio

Code Composer Studio (CCS) es una herramienta de desarrollo de software que soporta la mayor parte de los microcontroladores de Texas Instruments. Este IDE unifica una serie de herramientas de edición de código, compilación, profiling y debugging, lo que facilita el desarrollo.

Requerimientos de Hardware

	Mínimo	Recomendado
RAM	2GB	6GB
Espacio en disco	900MB	2 GB para soporte MSP430 y MSP432
Procesador	1.0 GHz x86	Dual core x86

Requerimientos de SO

- Windows
 - o Requiere Windows 7 SP1 o superior
 - La instalación fue probada en Windows 10 Pro 1809
- Linux
 - o Requiere distribuciones de 64 bits
 - \circ $\;$ La instalación fue probada en Ubuntu 18.04 LTS $\;$
 - Se reporta compatibilidad con Ubuntu 17.10, Ubuntu 16.04LTS, Mint 18 y Debian GNU
 9
- Mac
 - o Funciona con distribuciones más recientes

Guía de instalación Windows 10/11

- 1. Descargar el instalador de http://software-dl.ti.com/ccs/esd/CCSv8/CCS_8_3_0/exports/CCS8.3.0.00009_win32.zip
- 2. Descomprimir el archivo CCS8.3.0.00009_win32.zip
- 3. Dentro de la carpeta descomprimida, ejecutar ccs_setup_8.3.0.00009.exe
- 4. Elegir un directorio de instalación (en caso de no tener una preferencia, puede usarse el por defecto)

😚 Code Composer Studio v8 Setup					×
Choose Installation Location					
Where should Code Composer Studio v8 be instal	led?				
To change the main installation folder click the Bro	wse button.				
CCS Install Folder					
<u>c:\ti</u>					Browse
T					
rexas instruments			1		
		< Back	Next >	Finish	Cancel

5. Seleccionar soporte para la familia MSP430 y para la familia MSP432

Processor Support Select Product Families to be installed. MSP 430 ultra-low power MCUs SimpleLink MSP32 ^{Ult} Low power + performance MCUs SimpleLink MSP32 ^{Ult} Low power + performance MCUs Cortex ®-M4 core of SimpleLink MCUS 400 real-simpleLink Core activities MCUs CC2538 IEE 802.154 Wireless MCUs Compilers are insta CC000 real-time MCUs Compilers are insta CC000 real-time MCUs Compilers are insta CC000 real-time MCUs COMPA-11.7 DSP + ABM98 Processor	M® optimized for sensor nodes and processing gnals without
Select Product Families to be installed. Description ZIMSP430 ultra-low power MCUs Description ZimpleLink th MSP432 th low power * performance MCUs Cortex®-M4 core SimpleLink th W1-Fi® CC3bx wireless MCUs Cortex®-M4 core CC2530 tell CC3bx and CC2bx Wireless MCUs Intermet-of-Things CC2530 tell CC3bx wireless MCUs enable exquisition. CC2030 tell C02154 Wireless MCUs of high precision is CC2530 tell C02154 Wireless MCUs of high precision is C1000 real-time MCUs of high precision is TMAC12x ARM& Cortex®-M4F core-based MCUs sacrificing power. B Harcules th Safety MCUs Compilers are insta OMAP-Lits DP + ARM®P Processor Compilers are insta	M® optimized for sensor nodes and processing gnals without
MSP430 ultra-low power MCUs Description SimpleLink" MSP432= low power + performance MCUs MSP MCUs with AR SimpleLink" CG13x and CC26x Wireless MCUs Cortex® - M4 core SimpleLink" W-File CG3xa wireless MCUs Intermet-of-Things CC2538 URE CG22x Wireless MCUs enable causition CC2538 URE CG2154 Wireless MCUs of high precision si C2008 real-time MCUs of high precision si TM4C12x ARM® Cortex®-M4F core-based MCUs Sartficing power. B Sitara" AMK Processor Compilers are insta OMAP-Lins DP + ARM9® Processor Compilers are insta	M® optimized for sensor nodes and processing gnals without
MSF430 utitra-low power MCUs SimpleLink [™] (XC13x and CC26x Wireless MCUs SimpleLink [™] (XC13x and CC26x Wireless MCUs SimpleLink [™] Wireless MCUs CC2338 IEEE 802.15.4 Wireless MCUs CC2338 IEEE 802.15.4 Wireless MCUs CC2000 real-time MCUs TMAC12x ARM® Contex® -M4F core-based MCUs Sitara [™] AMx Processors Sitara [™] AMx Processors	RM® optimized for sensor nodes and processing gnals without
SimpleLink" Collsav and CZSAW Wireless MCUs Cortex®-AH core SimpleLink" Wi-Fi® CC32xx Wireless MCUs Internet-of-Things CC338 IEE BOLSAV Wireless MCUs enable acquisition IC239 IEE BOLSAV SimpleLink" Wireless MCUs CC348 IEE BOLSAV of high precisions IZ2000 real-time MCUs of high precisions IMMC12A RAM & Cortex ®-AHF core-based MCUs sacrificing power.B IHercules" Safety MCUs Compilers are insta Sitara" AMx Processors Compilers are insta	optimized for sensor nodes and processing gnals without
SimpleXink [™] Wi-Fi® CC32xx Wireless MCUs Internet-of-Things CC2536 IEEE 802.15.4 Wireless MCUs enable acquisition. CC200 real-time MCUs of high precision si TIM4C12x ARM® Cortex®-M4F core-based MCUs sacrificing power. Bit Strapel Ark/MS Core-based MCUs sacrificing power. Bit Stata [™] AMx Processors Compilers are insta OMAP-L1x DSP + ARM9® Processor Compilers are insta	sensor nodes and processing gnals without
C22538 IEEE 802.15.4 Wireless MCUs enable acquisition IC2000 real-time MCUs of hingh precisions in ITMAC12x ARMS Cortex®—MAF core-based MCUs sacrificing power. B Hercules" Safety MCUs Sitara" MAX Processors DMAP-L1x DSP + ARMS® Processor	and processing gnals without
C2000 real-time MCUs of high precision si TM4C12x ARM ® Cortex ®-MAF core-based MCUs sacrificing power. Bi Hercules ** Safety MCUs Compilers are insta Stara ** AMx Processors Ompilers are insta	ignals without
TM4C12x ARM ® Cortex ® -M4F core-based MCUs sacrificing power. B Hercules [®] Safety MCUs Compilers are insta Sitram [®] MM Processors OMAP-L1x DSP + ARM9 ® Processor	when The stand C.C.
Hercules" Safety MCUs Compilers are insta Sitara" MAK Processors OMAP-L1x DSP + ARM9® Processor	Joth II and GC
Sitara™ AMx Processors OMAP-L1x DSP + ARM9® Processor	alled.
OMAP-L1x DSP + ARM9® Processor	
DaVinci (DM) Video Processors	
OMAP Processors	
TDAx Driver Assistance SoCs & Jacinto DRAx Infotainment SoCs	
C55x ultra-low-power DSP	
C6000 Power-Optimized DSP	
66AK2x multicore DSP + ARM® Processors & C66x KeyStone™ multicore DSP	
mmWave Sensors	
I C64x multicore DSP	
Pok Sensor Signal Conditioners	
Select All Install Size: 1338.31	MB.
ras Instruments	

6. Seleccionar soporte únicamente para los debugger por defecto



Guía de instalación Ubuntu 18.04 LTS

- 1. Descargar el instalador de <u>http://software-</u> dl.ti.com/ccs/esd/CCSv8/CCS 8 3_0/exports/CCS8.3.0.00009_linux-x64.tar.gz
- 2. Ejecutar sudo apt update sudo apt install libc6:i386 libusb-0.1-4 libgconf-2-4 buildessential
- 3. Ir al directorio en el que se descargó el instalador y descomprimirlo tar xzvf file.tar.gz
- 4. Dentro de la carpeta del instalador, ejecutar ./ccs_setup_linux64_8.3.0.00009.bin
- 5. Elegir un directorio de instalación (en caso de no tener una preferencia, puede usarse el por defecto)

	Code (Composer Studio v	8 Setup	••
hoose Installation Loca	tion			52
Where should Code Compose	r Studio v8 be installe	d?		
To change the main inst	allation folder c	lick the Browse l	outton.	
CCS Install Folder				
/home/mauricio/ti	•			Browse
		< Back	Next >	Cancel

6. Seleccionar soporte para la familia MSP430 y para la familia MSP432

Code Con	nposer Studio v	v8 Setup		000
Processor Support Select Product Families to be installed.				
MSP430 ultra-low power MCUs SimpleLink [™] MSP432 [™] low power + performance M SimpleLink [™] MSP432 [™] low power + performance M SimpleLink [™] WF+I® CC23xx Wireless MCUs C2000 real-time MCUs TMAC12x ARM® Cortex®-M4F core-based MCUs Hercules [™] Safety MCUs Sitara [™] AMK Processors OMAP-L1x DSP + ARM9® Processors OMAP Processo	nent SoCs eyStone [™] multici	ore DSP	Description MSP MCUs with ARM € core optimized for Inte sensor nodes enable a processing of high pre without sacrificing pov GCC Compilers are inst	Cortex®-M4 rmet-of-Things cquisition and cision signals ver. Both TI and alled.
Select All		1	Install Size: 1411.84 ME	3.
Texas Instruments	< Back	Next >	Finish	Cancel

7. Seleccionar soporte únicamente para los debugger por defecto

Code Co	omposer Studio	v8 Setup		۵ 🕲
Select Debug Probes Select the debug probes you want installed and des	elect the debug pr	obes you want to lea	ve out.	
IT XDS Debug Probe Support Blackhawk Debug Probes Spectrum Digital Debug Probes and Boards MSP Flash/FRAM Emulation Tools (MSP FETs) SEGGER J-Link		Descr	iption —	
Select All		Install	Size: 1563.23 M	В.
Texas Instruments				
	< Back	Next >	Finish	Cancel

- 8. Verificar que su launchpad no esté conectado a su PC
- 9. Desde un terminal, ejecutar cd <DIRECTORIO DE INSTALACIÓN>/ccsv8/instal_scripts sudo ./install_drivers.sh

Inicialización de CCS

- 1. Abrir el programa
- En la primera ejecución el programa permite seleccionar un Workspace. Elegir la carpeta que considere adecuada (sugerimos C:\Workspace\CCS opara Windows y ~/Workspace/CCS para linux) y seleccionar "Use this as the default and do not ask again"

Eclipse Launcher	8
Select a directory as workspace	
Code Composer Studio uses the workspace directory to store its preferences and development artifacts.	
Workspace: //home/ /workspace_v8 Browse	
Use this as the default and do not ask again	
Cancel Launch	

3. Cerrar CCS

2. Instalación de Git

Git es un software de control de versiones, cuyo objetivo es la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones, particularmente cuando éstas tienen un gran número de archivos de código fuente. Consiste en llevar un registro de los cambios que ocurren en los archivos de código fuente de una aplicación, permitiendo coordinar el trabajo que varios desarrolladores realizan sobre estos archivos.

Dependiendo del sistema operativo que use, se sugieren diferentes versiones de la herramienta.

Guía de instalación e inicialización Windows 10/11 (Git for Windows)

- 1. Descargue la última versión de Git for Windows de https://gitforwindows.org/
- 2. Ejecutar el .exe
- 3. Mantenga los componentes de instalación por defecto

🚸 Git 2.21.0 Setup — 🗆	×
Select Components Which components should be installed?	*
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.)
Additional icons	
On the Desktop	
Windows Explorer integration	
Git Bash Here	
\square Associate git* configuration files with the default text editor	
Associate .sh files to be run with Bash	
Use a TrueType font in all console windows	
Check daily for Git for Windows updates	
Current selection requires at least 246.5 MB of disk space.	
https://gitforwindows.org/	
< Back Next >	Cancel

- Seleccione el editor de texto por defecto de su preferencia. Se sugiere seleccionar Sublime (requiere instalación) o Nano, pero aprender a usar Vim puede resultarle útil en algún momento. Si no está seguro, investigue.
- 5. Seleccione la opción "Git from the command line and also from 3rd party software". La primera opción también es aceptable, en caso de que la prefiera.

🚸 Git 2.21.0 Setup			×	
Adjusting your PATH environment How would you like to use Git from the command line?			3	
◯ Use Git from Git Bash only				
This is the safest choice as your PATH will not be modified at a able to use the Git command line tools from Git Bash.	ll. You w	ill only be		
Git from the command line and also from 3rd-party software	are			
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.				
◯ Use Git and optional Unix tools from the Command Promp	t			
Both Git and the optional Unix tools will be added to your PATH Warning: This will override Windows tools like "find" and "sort", use this option if you understand the implications.	l. . Only			
< Back Nex	(t >	Can	cel	

6. Seleccione la opción "Use the OpenSSL library"

🚸 Git 2.21.0 Setup			×
Choosing HTTPS transport backend Which SSL/TLS library would you like Git to use for HTTPS connection	ns?	1	
Use the OpenSSL library Server certificates will be validated using the ca-bundle.crt file.	•		
○ Use the native Windows Secure Channel library Server certificates will be validated using Windows Certificate S This option also allows you to use your company's internal Root distributed e.g. via Active Directory Domain Services.	Stores. t CA certi	ficates	
https://gitforwindows.org/			
< Back Nex	kt >	Cancel	

7. Seleccione la primera opción de conversión de fin de línea

🚸 Git 2.21.0 Setup		-		
Configuring the line ending conversions How should Git treat line endings in text files?			*	
Checkout Windows-style, commit Unix-	style line endings			
Git will convert LF to CRLF when checking text files, CRLF will be converted to LF. For this is the recommended setting on Windo	out text files. When c cross-platform proje ws ("core.autocrlf" is	cts, set to "tru	g ue").	
○ Checkout as-is, commit Unix-style line e	endings			
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").				
○ Checkout as-is, commit as-is				
Git will not perform any conversions when text files. Choosing this option is not recom projects ("core.autocrlf" is set to "false"). https://gitforwindows.org/	checking out or comm mended for cross-pla	nitting atform		
	< Back Next	t >	Cancel	

8. Selecciones "Use MinTTY"

🚸 Git 2.21.0 Setup			×
Configuring the terminal emulator to use with Git Bash Which terminal emulator do you want to use with your Git Bash?			>
Use MinTTY (the default terminal of MSYS2)			
Git Bash will use MinTTY as terminal emulator, which sports a non-rectangular selections and a Unicode font. Windows cons as interactive Python) must be launched via `winpty` to work	resizable ole progra in MinTTY	window, ms (such (.	
○ Use Windows' default console window			
Git will use the default console window of Windows ("cmd.exe with Win32 console programs such as interactive Python or no very limited default scroll-back, needs to be configured to use order to display non-ASCII characters correctly, and prior to V window was not freely resizable and it only allowed rectangul	e"), which ode.js, but e a Unicode Nindows 1 ar text sele	works wel has a e font in 0 its ections.	I
https://gitforwindows.org/			
< Back No	ext >	Can	cel

9. Habilite todas las opciones adicionales

A continuación, debe configurar su identidad en su versión de Git.

10. Ejecute Git Bash

Git Bash es un emulador de Bash para Windows, que le permitirá utilizar esta herramienta de control de versiones tal cual se usaría en un entorno Linux.

11. En el terminal, ejecute

\$ git config --global user.name "Julio Vilamajó" \$ git config --global user.email jvilamajo@fing.edu.uy Sustituyendo su nombre y correo. En caso de que desee usar otros repositorios de Git con otras credenciales, puede sustituir el flag --global con --local (requiere crear previamente el repositorio a trabajar)

Al terminar de instalar Git for Windows, es conveniente verificar si su PC tiene disponible un conjunto de claves SSH. Las claves SSH se utilizan para acceder de manera segura y rápida a los servidores remotos de Git desde una PC confiable.

En caso de que quiera acceder a su repositorio desde una PC de uso común, se sugiere utilizar acceso HTTPS

- 12. En el terminal de Git, ingrese
 - \$ ssh-keygen -t rsa -b 4096 -C your_email@example.com
- 13. Acepte todas las opciones por defecto

Guía de instalación e inicialización Ubuntu 18.04 LTS

 En un terminal, ejecute sudo apt install git-all

A continuación, debe configurar su identidad en su versión de Git.

- 2. En el terminal, ejecute
 - \$ git config --global user.name "Julio Vilamajó" \$ git config --global user.email jvilamajo@fing.edu.uy Sustituyendo su nombre y correo. En caso de que desee usar otros repositorios de Git con otras credenciales, puede sustituir el flag --global con --local (requiere crear previamente el repositorio a trabajar)

Es conveniente verificar si su PC tiene disponible un conjunto de claves SSH. Las claves SSH se utilizan para acceder de manera segura y rápida a los servidores remotos de Git desde una PC confiable.

En caso de que quiera acceder a su repositorio desde una PC de uso común, se sugiere utilizar acceso HTTPS

- 3. En el terminal, ingrese
 - \$ ssh-keygen -t rsa -b 4096 -C your_email@example.com
- 4. Acepte todas las opciones por defecto

3. Puesta a punto del repositorio del curso

El repositorio de proyecto debe ser creado solamente por uno de los integrantes de cada grupo de laboratorio, mientras que será clonado por los restantes (ver el apartado "Clonación del repositorio del curso")

1. Diríjase a https://gitlab.fing.edu.uy e ingrese con su nombre de usuario y contraseña

Los pasos 2 al 6 solamente aplican en caso de que esté usando un PC personal

Diríjase al directorio que contiene sus claves SSH:
 C:\Users\usuario\.ssh en Windows

~/.ssh en Linux

- 3. Abra el archivo id_rsa.pub y copie todo el contenido.
- 4. En <u>https://gitlab.fing.edu.uy</u>, haga clic en la figura de la esquina superior derecha y en Settings...
- 5. En la columna de la izquierda, haga clic en SSH Keys



- 6. Pegue el contenido de la clave SSH pública y asígnele un nombre
- 7. En Projects, haga clic en New Project
- 8. Asigne el nombre de proyecto "Laboratorios Sisem"
- 9. Seleccione el Nivel de Visibilidad en Privado, y haga clic en Create Project

Blank project	Create from template	Import project	CI/CD for external repo
Project name			
Laboratorios Sisem			
Project URL		Project slug	
https://gitlab.com/mgonzalezgenta/		laboratorios-sisem	
Want to house several depende	nt projects under the same name	espace? Create a group.	
Project description (optional)			
Description format			ß
 Private Project access must be 	granted explicitly to each user.		
Internal The project can be acce	ssed by any logged in user.		
Public The project can be acce	ssed without any authentication.		
Initialize repository with a Allows you to immediately c	README lone this project's repository. Skij	o this if you plan to push up an exis	ting repository.
Create project			Cancel

10. En la columna de la izquierda, seleccione Settings -> Project Information -> Members

11. Desde la pestaña "Invite group" Otorgue acceso de "Maintainer" al grupo de docentes. Los usuarios son:

juliano@fing.edu.uy leo@fing.edu.uy lbarboni@fing.edu.uy rodrigog@fing.edu.uy jlema@fing.edu.uy rcabral@fing.edu.uy isabaj@fing.edu.uy

Desde la pestaña "Invite meber" otorgue acceso de "Maintainer" a sus compañeros de laboratorio.

12. Obtenga la URL de su proyecto. En la web del proyecto, seleccione Clonar y copie el acceso SSH o HTTPS según corresponda

		۵ ×	🔂 Star	0	Clone 🗸
Defen Tilen	Clone with SSH				
bytes Files	git@gitl	ab.com:mg	onzalezge	nta/l	G
	Clone with	HTTPS			
e instructions below.	https://	gitlab.co	m/mgonzal	.ezgen	G

- 13. Abra un terminal o Git Bash, según corresponda
- 14. En el terminal, diríjase al Workspace de CCS
 - En caso de haber seguido la sugerencia, el comando es
 - cd C:\Workspace\CCS para Windows
 - cd ~/Workspace/CCS para Linux
- 15. Ejecute el comando
 - git clone <enlace-al-proyecto>
- 16. Podrá observar que dentro del Workspace de CCS se generó una carpeta de nombre laboratorios-sisem, que a su vez contiene una carpeta oculta llamada .git
- 17. Dentro de la carpeta laboratorios-sisem, cree la siguiente estructura de directorios
 - includesrctest
- 18. Dado que Git no prevé la inclusión de directorios vacíos, pero que en nuestro caso deseamos compartir algunas carpetas que no contienen archivos, cree un archivo vacío de nombre .keep dentro de cada uno de estos directorios.

Primer push al repositorio del curso

- 1. Abra un terminal o Git Bash, según corresponda.
- 2. Navegue hasta el directorio raíz del proyecto.
 - cd C:\Workspace\CCS\laboratorios-sisem para Windows
 - cd ~/Workspace/CCS\laboratorios-sisem para Linux
- 3. Introduzca la siguiente secuencia de comandos
 - git add *
 - Agrega todos los archivos modificados de nuestro repositorio (los tres .keep) al área de stage
 - git commit -m "Commit inicial del laboratorio"
 - Realiza un commit con todos los archivos que se encuentran en el área de stage bajo el mensaje "Commit inicial del laboratorio"
 - git push origin main
 - Sube el último commit al repositorio remoto

En caso de solicitarse, ingrese sus credenciales

Clonación del repositorio por parte de sus compañeros

1. Obtenga la URL del proyecto. En la web del proyecto, seleccione Clonar y copie el acceso SSH o HTTPS según corresponda

		Clone 🗸
Partice Files	Clone with SSH	
bytes Files	git@gitlab.com:mgonzalezgenta/l	G
	Clone with HTTPS	
instructions below	https://gitlab.com/mgonzalezgen	G

about protected branches

- 2. Abra un terminal o Git Bash, según corresponda
- 3. En el terminal, diríjase al Workspace de CCS En caso de haber seguido la sugerencia, el comando es
 - cd C:\Workspace\CCS para Windows
 - cd ~/Workspace/CCS para Linux
- Ejecute el comando git clone <enlace-al-proyecto>

Creación de un nuevo proyecto de CCS

Este proceso debe realizarse previo a cada laboratorio.

- 1. Abra CCS
- 2. Seleccionar File/New/Project...
- 3. En el cuadro de selección de Wizards, seleccionar Code Composer Studio/CCS Project



4. Dentro del cuadro CCS Project, realizar las siguientes tareas

CS Project Create a new CCS P	roject.		
arget: <selec< td=""><td>t or type filter text></td><td>MSP430G2553</td><td></td></selec<>	t or type filter text>	MSP430G2553	
onnection: TI MSP	430 USB1 [Default]	~	Identify
1 MSP430			
Project name:	lab1		
Use default loc	ation		
Location:	C:\Workspace\CCS\labsSisen	n\test\lab1\	Browse
Compiler version:	TI v18.1.5.LTS	~	More
 Tool-chain Project template type filter text 	es and examples	Creates an empty project initializ	ed for the
 Tool-chain Project template type filter text Empty Pro Empty Pro Empty Empty Empty Empty Sasic Exam Blink Th 	es and examples jects Project Project (with main.c) Assembly-only Project RTSC Project aples ne LED	Creates an empty project initializ selected device. The project will empty 'main.c' source-file.	ed for the contain an

- Usando el filtro de Target, seleccionar el MCU del Launchpad (MSP430G2553)
- Elegir el puerto correcto en Connection
- Hacer clic en Identify...
 - o Durante el primer uso del lauchpad puede aparecer el siguiente diálogo

?	Error initializing A firmware upda (MSP-FET430UII the firmware an update steps).	emulator: ate is required for F / MSP-FET / eZ- d launch your del	the MSP430 Debug In FET). Click the "Update bug session (this may	nterface " button to update require several
	DO NOT UNPLU	IG THE INTERFACE	DURING THE UPDATE	

- En ese caso, seleccionar "Update". No desconectar la placa durante el proceso
- Asignar el nombre "labN" al proyecto, donde N es el número de laboratorio
- Asignarle la ubicación laboratorios-sisem/test/labN/ dentro del Wokspace de CCS
- Elegir la versión más reciente del compilador de TI
- Elegir el template Empty Project (with main.c)
- 5. Hacer clic en Finish
- 6. Hacer clic derecho en el proyecto, seleccionar New -> Folder



7. En la ventana New Folder, seleccionar Advanced, Link to alternate location y especificar la ruta

\${WORKSPACE_LOC}/laboratorios-sisem/include

Nota: CCS resuelve de manera automática la variable \${WORKSPACE_LOC}

- 8. Repetir los pasos 6 y 7 para la ruta \${WORKSPACE_LOC}/laboratorios-sisem/src
- 9. Hacer clic derecho en el proyecto, e ir a Properties
- 10. En la lista de la izquierda, seleccionar Build/MSP430 Compiler/Include Options

type filter text	Include Options	← + ⇒ + +
 Resource General Build MSP430 Compiler Processor Options 	Configuration: Debug [Active]	✓ Manage Configurations
Optimization Include Options ULP Advisor Advice Options Predefined Symbols Advanced Options MSP430 Linker MSP430 Hax Utility [Disabled] Debug Git	Add dir to #include search path (include_path, -I) <u>S(CCS_BASE_ROOT)/msp430/include</u> S(PROJECT_ROOT) [] S(CG_TOOL_ROOT)/include []	ରି ଲି କି ତି । ହ
	Specify a preinclude file (preinclude)	▲ 副 副 告日会
Show advanced settings		Apply and Close Cancel

- 11. Agregar la ruta \${WORKSPACE_LOC}/laboratorios-sisem/include
- 12. Hacer clic en Apply and close