

# 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
  - 1.a Guía de instalación para Windows 10
  - 1.b Guía de instalación para Ubuntu 18.04
  - 1.c Inicialización de CCS
- 2 Instalación de Git
  - 2.a Guía de instalación para Windows 10
  - 2.b Guía de instalación para Ubuntu 18.04
- 3 Puesta a punto del repositorio del curso
  - 3.a Creación del repositorio del curso
  - 3.b Primer push al repositorio del curso
  - 3.c Clonación del repositorio del curso
  - 3.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

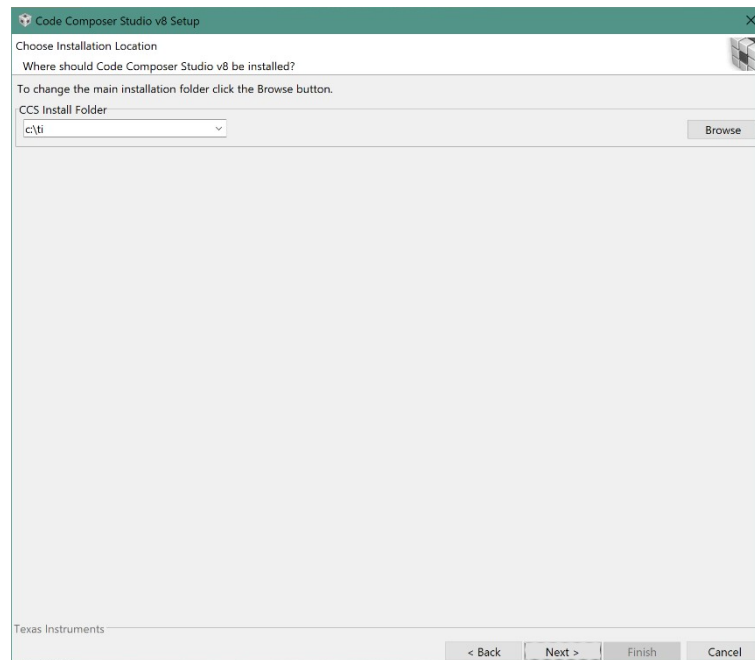
	Mínimo	Recomendado
<b>RAM</b>	2GB	6GB
<b>Espacio en disco</b>	900MB	2 GB para soporte MSP430 y MSP432
<b>Procesador</b>	1.0 GHz x86	Dual core x86

### Requerimientos de SO

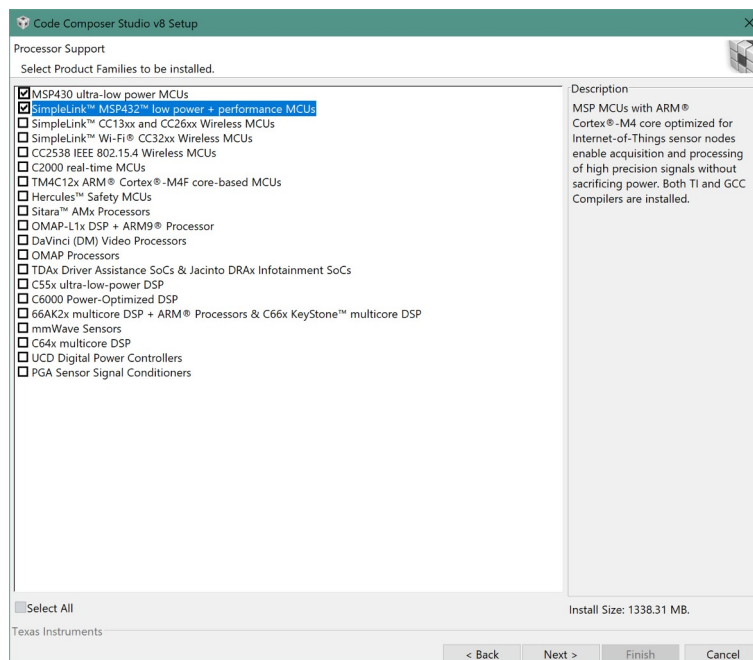
- Windows
  - Requiere Windows 7 SP1 o superior
  - La instalación fue probada en Windows 10 Pro 1809
- Linux
  - Requiere distribuciones de 64 bits
  - La instalación fue probada en Ubuntu 18.04 LTS
  - Se reporta compatibilidad con Ubuntu 17.10, Ubuntu 16.04LTS, Mint 18 y Debian GNU 9
- Mac
  - Funciona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

## Guía de instalación Windows 10

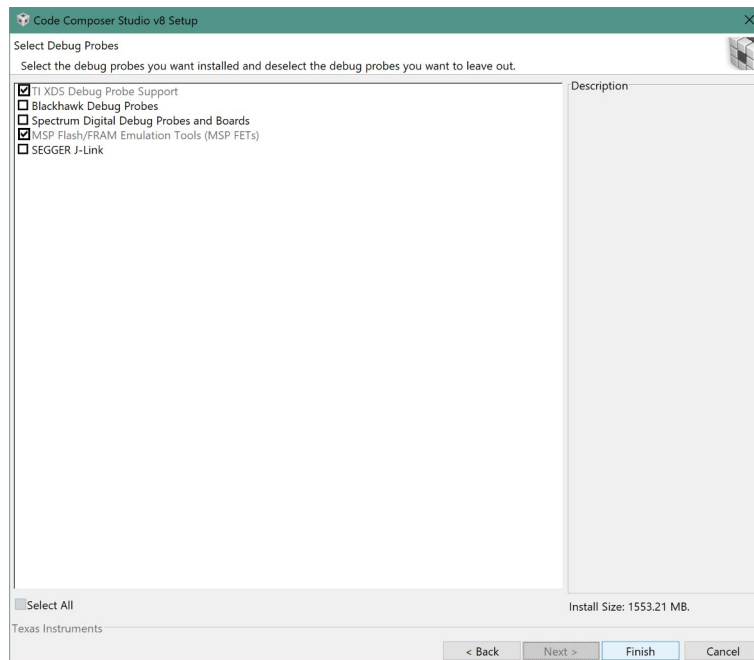
- 1 Descargar el instalador de  
[http://software-dl.ti.com/ccs/esd/CCSv8/CCS\\_8\\_3\\_0/exports/CCS8.3.0.00009\\_win32.zip](http://software-dl.ti.com/ccs/esd/CCSv8/CCS_8_3_0/exports/CCS8.3.0.00009_win32.zip)
- 2 Descomprimir el archivo CCS8.3.0.00009\_win32.zip
- 3 Dentro de la carpeta descomprimida, 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)



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



- 6 Seleccionar soporte únicamente para los debugger por defecto



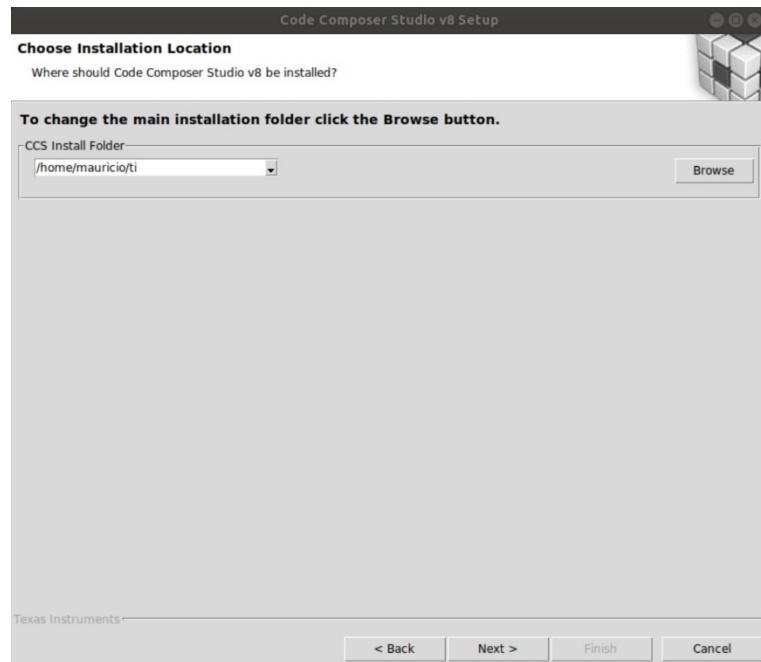
## Guía de instalación Ubuntu 18.04 LTS

- 1 Descargar el instalador de  
[http://software-dl.ti.com/ccs/esd/CCSv8/CCS\\_8\\_3\\_0/exports/CCS8.3.0.00009\\_linux-x64.tar.gz](http://software-dl.ti.com/ccs/esd/CCSv8/CCS_8_3_0/exports/CCS8.3.0.00009_linux-x64.tar.gz)
- 2 Ejecutar  

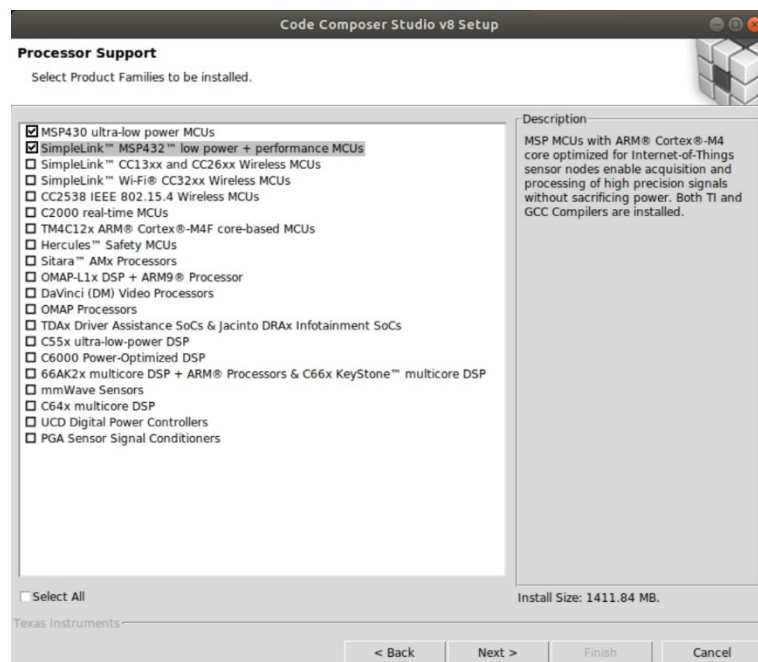
```
sudo apt update  
sudo apt install libc6:i386 libusb-0.1-4 libgconf-2-4 build-essential
```
- 3 Ir al directorio en el que se descargó el instalador y descomprimirlo  

```
tar xzvf file.tar.gz
```
- 4 Dentro de la carpeta del instalador, ejecutar  

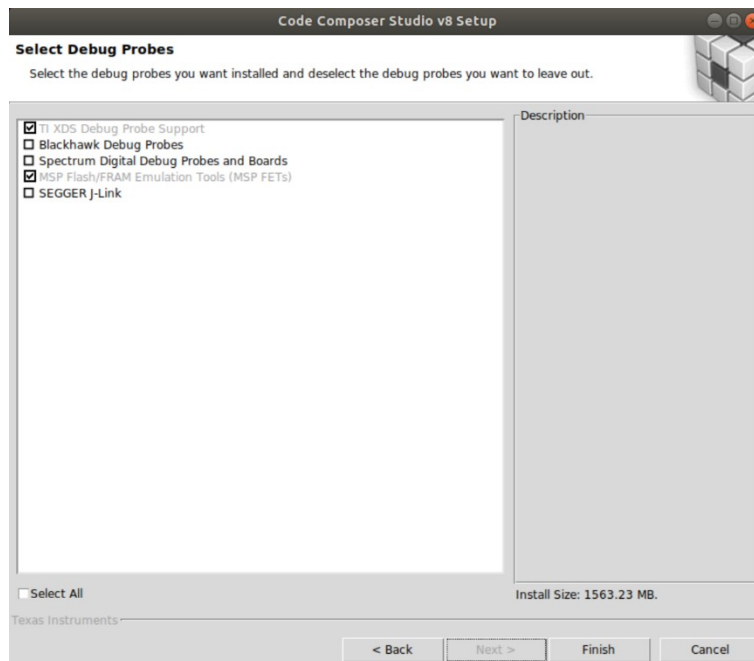
```
./ccs_setup_linux64_8.3.0.00009.bin
```
- 5 Elegir un directorio de instalación (en caso de no tener una preferencia, puede usarse el por defecto)



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



7 Seleccionar soporte únicamente para los debugger por defecto



8 Verificar que su launchpad no esté conectado a su PC

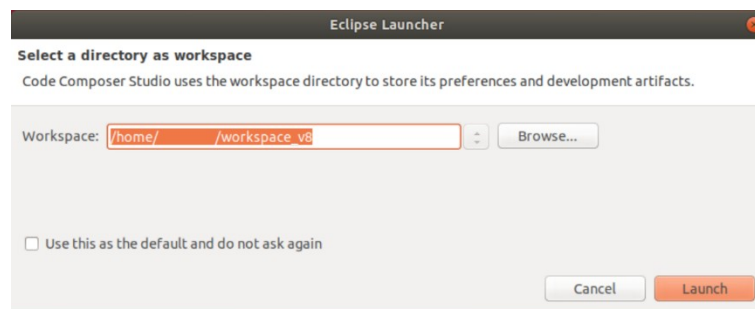
9 Desde un terminal, ejecutar

```
cd <DIRECTORIO DE INSTALACIÓN>/ccsv8/instal_scripts  
sudo ./install_drivers.sh
```

Inicialización de CCS

1 Abrir el programa

2 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”



3 Cerrar CCS

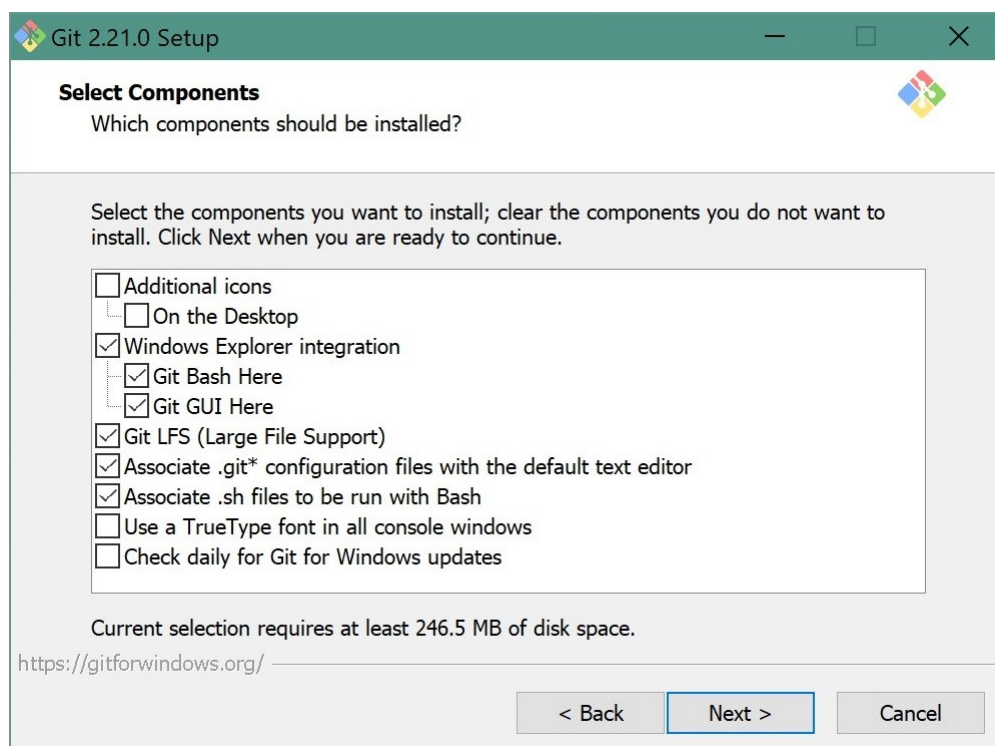
## 2 Instalación de Git

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.

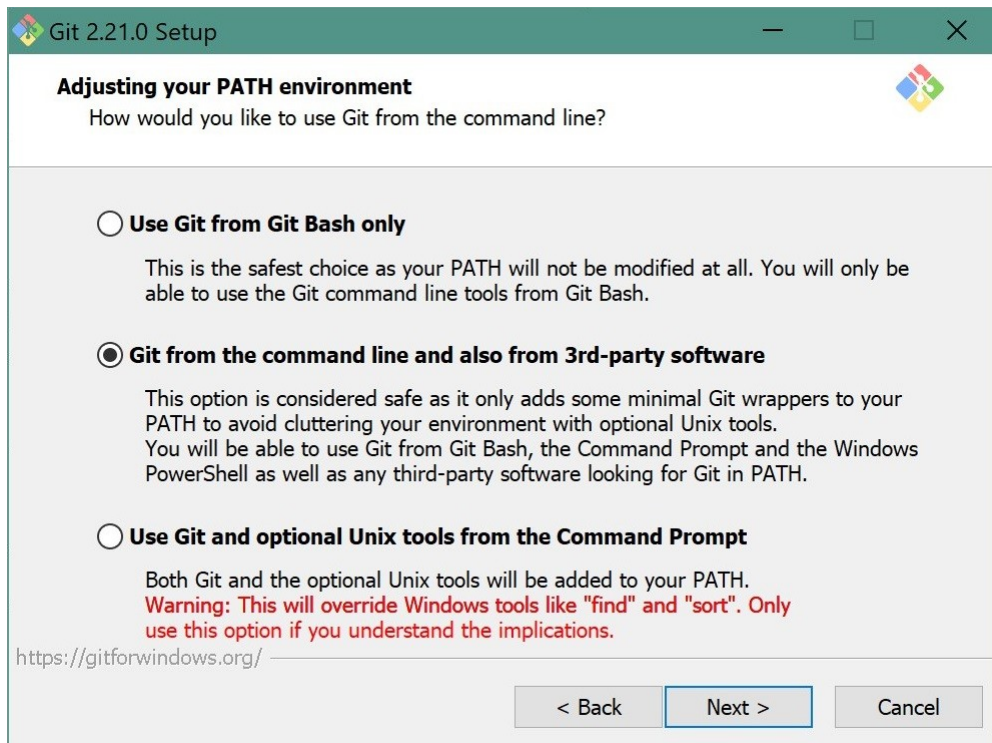
Dependiendo del sistema operativo que use, se sugieren diferentes versiones de la herramienta.

### Guía de instalación e inicialización Windows 10 (Git for Windows)

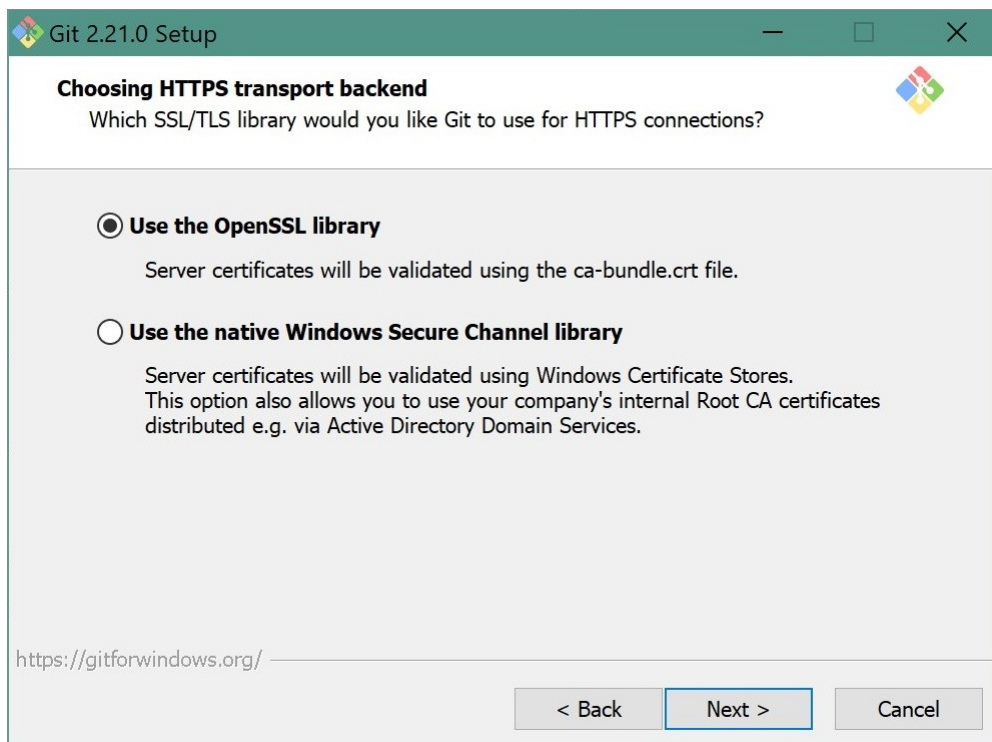
- 1 Descargue la última versión de Git for Windows de <https://gitforwindows.org/>
- 2 Ejecutar el .exe
- 3 Mantenga los componentes de instalación por defecto



- 4 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.

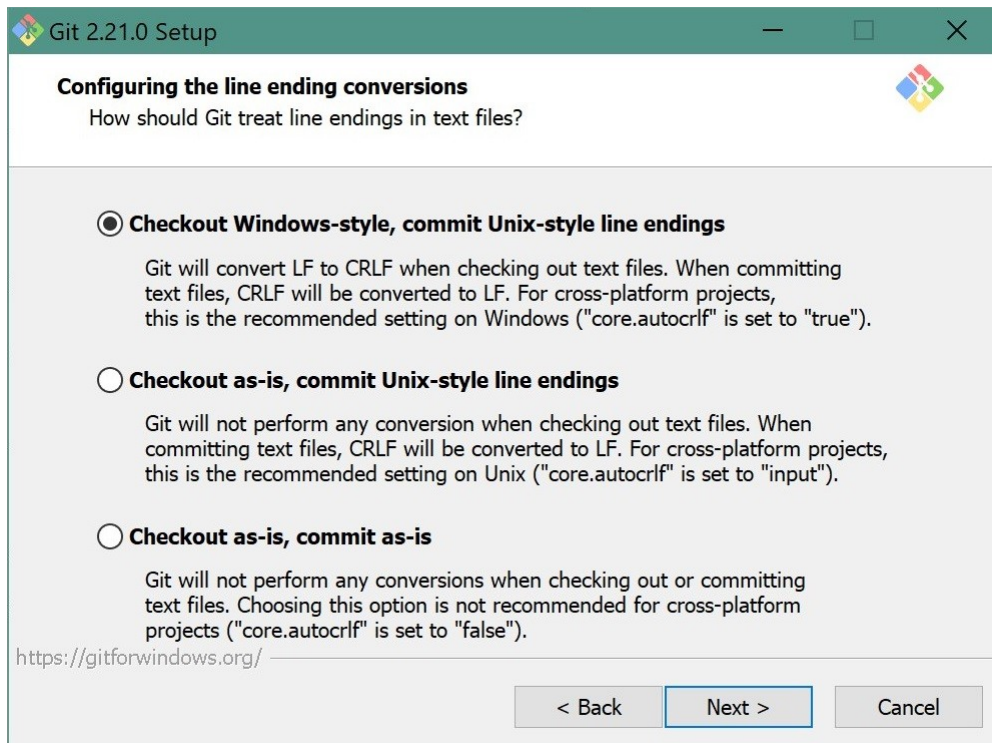


6 Seleccione la opción “Use the OpenSSL library”

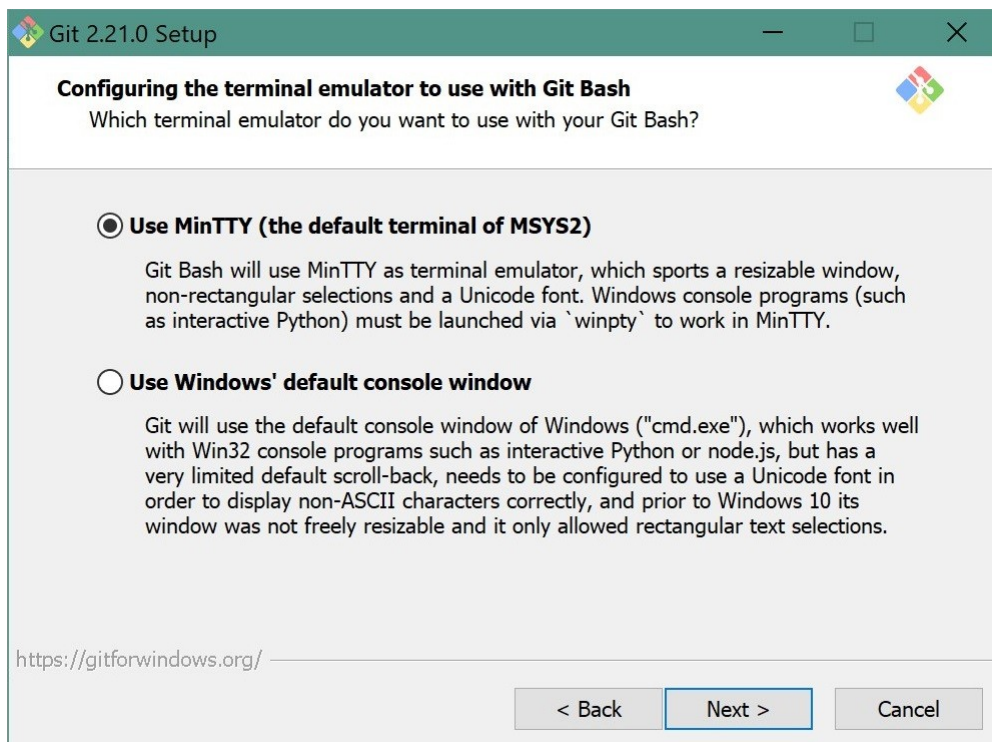


7 Seleccione la primera opción de conversión de fin de línea





## 8 Selecciones "Use MinTTY"



## 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

- 1 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

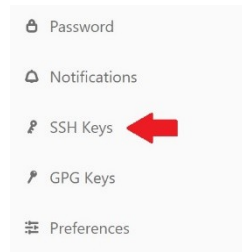
### **3 Puesta a punto del repositorio del curso**

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

- 2 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 <https://gitlab.fing.edu.uy>, 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
<p><b>Project name</b></p> <input type="text" value="Laboratorios Sisem"/>			
<p><b>Project URL</b></p> <input type="text" value="https://gitlab.com/mgonzalezgenta/"/>		<p><b>Project slug</b></p> <input type="text" value="laboratorios-sisem"/>	
<p>Want to house several dependent projects under the same namespace? <a href="#">Create a group</a>.</p>			
<p><b>Project description (optional)</b></p> <div>             Description format           </div>			
<p><b>Visibility Level ?</b></p> <p> <input checked="" type="radio"/> <b>Private</b>              Project access must be granted explicitly to each user.           </p> <p> <input type="radio"/> <b>Internal</b>              The project can be accessed by any logged in user.           </p> <p> <input type="radio"/> <b>Public</b>              The project can be accessed without any authentication.           </p>			
<p> <input type="checkbox"/> <b>Initialize repository with a README</b>              Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.           </p>			
<p>Create project</p>			<p>Cancel</p>

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

- 11 Desde la pestaña “Invite meber” otorgue acceso de “Maintainer” a los docentes del curso: @jsanchez, @mgonzalez, @ccabrera, @vcabrera, @jschandy, @lbarboni, @leo, @juliano
- 12 Desde la pestaña “Invite meber” otorgue acceso de “Maintainer” a sus compañeros de laboratorio.
- 13 Obtenga la URL de su proyecto. En la web del proyecto, seleccione Clonar y copie el acceso SSH o HTTPS según corresponda



- 14 Abra un terminal o Git Bash, según corresponda
- 15 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
- 16 Ejecute el comando  
    `git clone <enlace-al-proyecto>`
- 17 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
- 18 Dentro de la carpeta laboratorios-sisem, cree la siguiente estructura de directorios
  - include
  - src
  - test
- 19 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



- 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
```

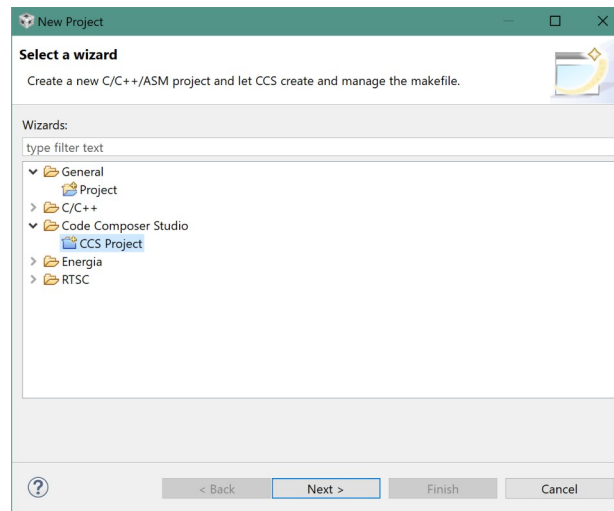
 para Linux
- 4 Ejecute el comando  

```
git clone <enlace-al-proyecto>
```

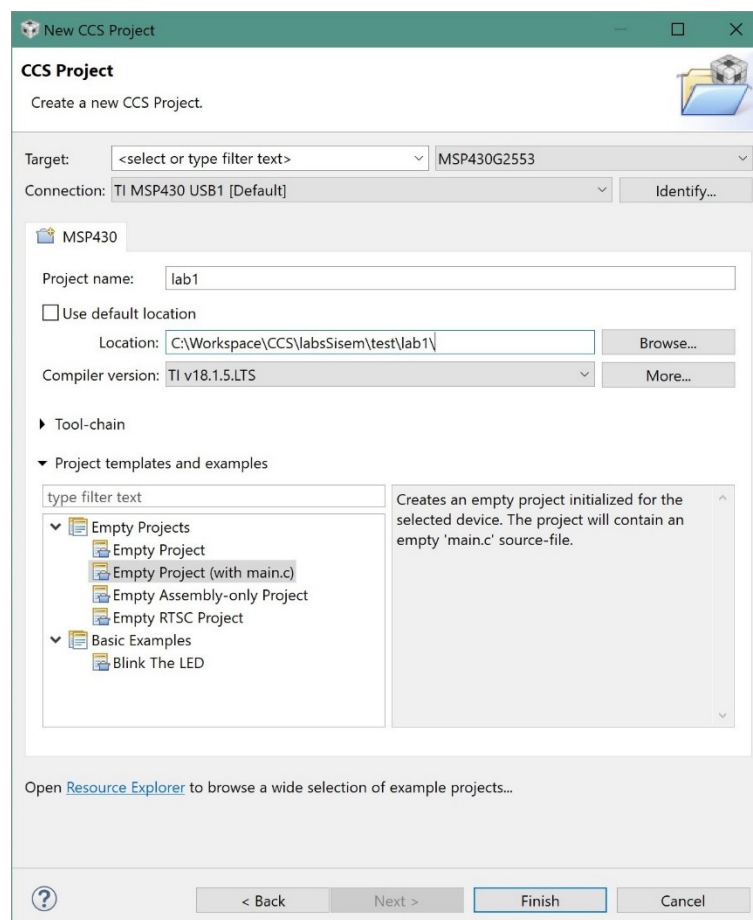
#### Creación de un nuevo proyecto de CCS

Este proceso debe realizarse previo a cada laboratorio.

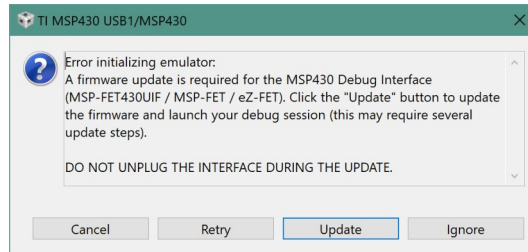
- 1 Abra CCS
- 2 Seleccionar File/New/Project...
- 3 En el cuadro de selección de Wizards, seleccionar Code Composer Studio/CCS Project



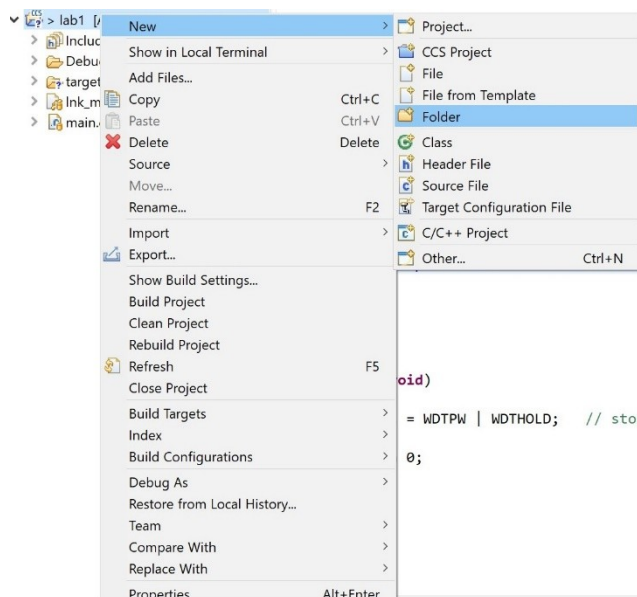
#### 4 Dentro del cuadro CCS Project, realizar las siguientes tareas



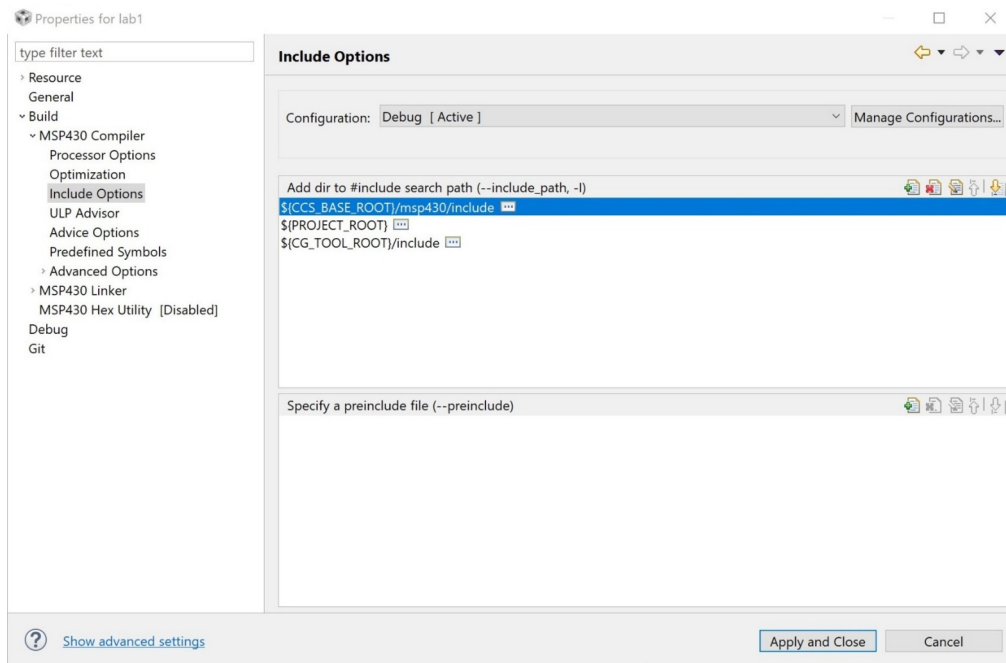
- Usando el filtro de Target, seleccionar el MCU del Launchpad (MSP430G2553)
- Elegir el puerto correcto en Connection
- Hacer clic en Identify...
  - Durante el primer uso del launchpad puede aparecer el siguiente diálogo



- 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



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

Este material didáctico fue elaborado por docentes del Departamento de Electrónica de la Universidad de la República a lo largo a varios años. Se pone a disposición de la comunidad bajo la licencia “Creative Commons Attribution 4.0 International License”.

Ver detalles de la licencia aquí: <https://creativecommons.org/licenses/by/4.0/>



FACULTAD DE  
INGENIERÍA  
UDELAR



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



Co-funded by the  
Erasmus+ Programme  
of the European Union