



# **Guía instalación de STM32CubeIDE e importación del primer proyecto**

**Gustavo Adolfo Osorio Londoño - Samuel Esteban Reyes -  
Santiago Matta Amador**

*Universidad Nacional de Colombia sede Manizales Email: [gaosoriol@unal.edu.co](mailto:gaosoriol@unal.edu.co)  
[sreyesn@unal.edu.co](mailto:sreyesn@unal.edu.co) [smatta@unal.edu.co](mailto:smatta@unal.edu.co)*

## **Introducción**

La plataforma de desarrollo STM cuenta con una IDE creada por la misma empresa para la programación de las distintas placas que fabrican, la cual genera una parte del código al inicio de cada proyecto según las necesidades que el usuario haya especificado al comenzar el proyecto, sin embargo para los primeros códigos que se desarrollaran en el curso se hará uso de la plantilla Led-ON que se encuentra en el repositorio de la asignatura en Github ([Click aquí](#)).

# Contenido

<b>1.Instalación del STM32CubeIDE .....</b>	<b>3</b>
<b>2.Primer Proyecto .....</b>	<b>4</b>
2.1 Importar proyecto al Workspace .....	4
2.2 Compilar proyecto .....	6
2.3 Debug .....	6
<b>3.STM32L476 MCU.....</b>	<b>8</b>
3.1 Instrucciones en Ensamblador .....	8

## 1 Instalación STM32CubeIDE

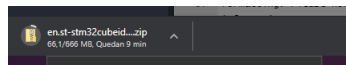
Para comenzar descargamos la IDE para la programación, lo cual se hace ingresando a la pagina:

<https://www.st.com/en/development-tools/stm32cubeide.html>

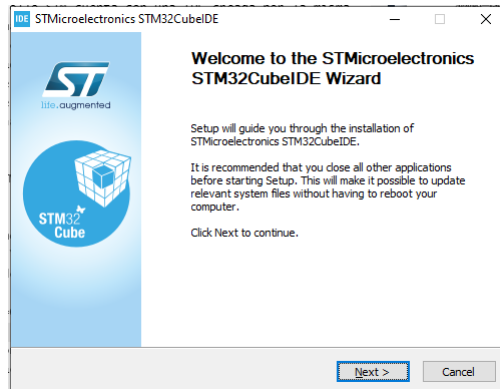
Get Software

Part Number	General Description	Supplier	Download	All versions
STM32CubeIDE-DEB	STM32CubeIDE Debian Linux Installer	ST	Get latest	Select version 1.10.1
STM32CubeIDE-Lnx	STM32CubeIDE Generic Linux Installer	ST	Get latest	Select version 1.10.1
STM32CubeIDE-Mac	STM32CubeIDE macOS Installer	ST	Get latest	Select version 1.10.1
STM32CubeIDE-RPM	STM32CubeIDE RPM Linux Installer	ST	Get latest	Select version 1.10.1
STM32CubeIDE-Win	STM32CubeIDE Windows Installer	ST	Get latest	Select version 1.10.1

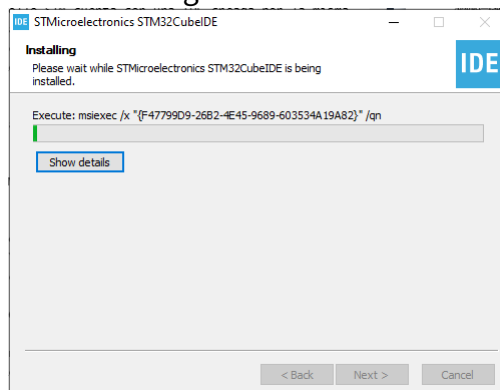
En esta pagina seleccionamos la opción que corresponda con nuestro sistema operativo y hacemos click en Get Software (Descargar la versión mas reciente del sistema). Entonces iniciara la descarga del instalador.



Al terminar la descarga, descomprimos el archivo y lo ejecutamos.



Next → I Agree → Next → Install



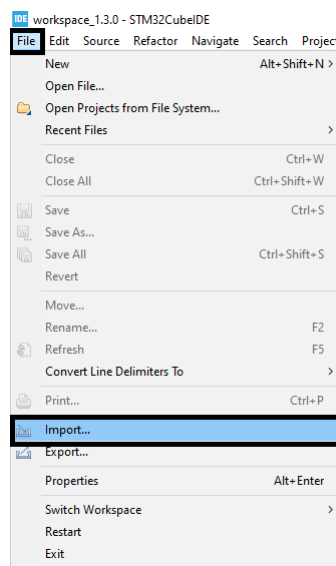
Después de esto dejamos que se termine de instalar el programa.

## 2 PrimerProyecto

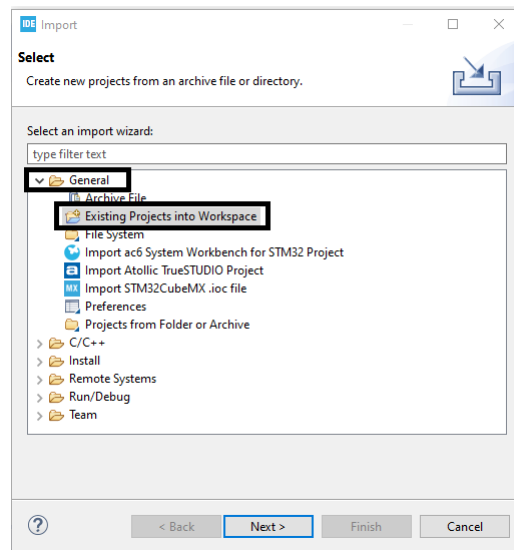
Los proyectos de esta plataforma pueden dejarse guardados en una carpeta Workspace del IDE, antes de comenzar la creación de algún proyecto vamos a iniciar descargando un código para verificar que se realizó bien la instalación y todo funciona correctamente, el código que vamos a descargar es el de Led\_ON que se encuentra en el repositorio de la asignatura en Github ([Click aquí](#)), para descargar solo la carpeta del primer código entramos a DownGit ([Click Aquí](#)) colocamos el link de la carpeta y la descargamos, una vez descargada la carpeta del primer código la movemos al Workspace que usaremos.

### 2.1 Importar Proyecto al Workspace

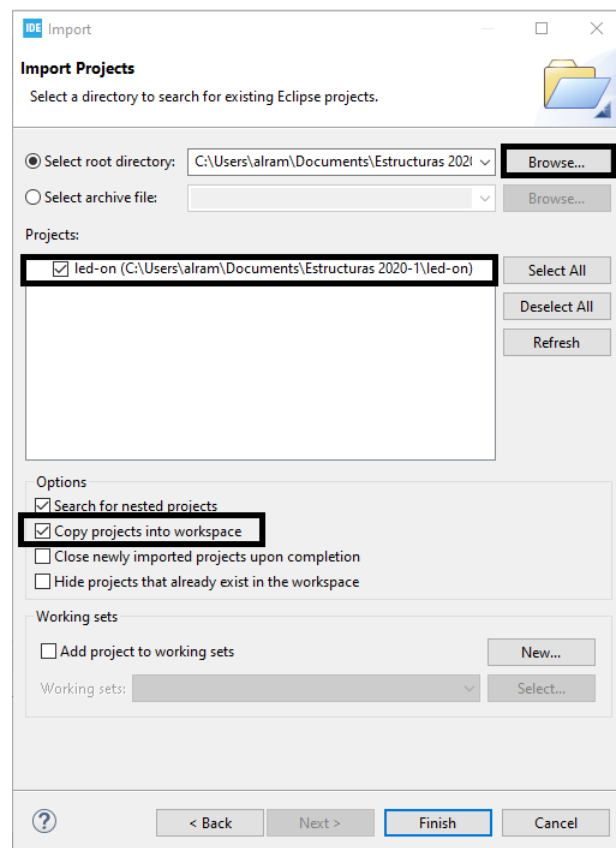
Después de colocar el archivo en el Workspace que usaremos, descomprimos el archivo y nos dirigimos a la aplicación de stm descargada anteriormente, le damos en el botón FILE que se encuentra en la esquina superior izquierda para luego seleccionar la opción de IMPORT:



Entramos a la carpeta General → Existing projects into Workspace y después Next:

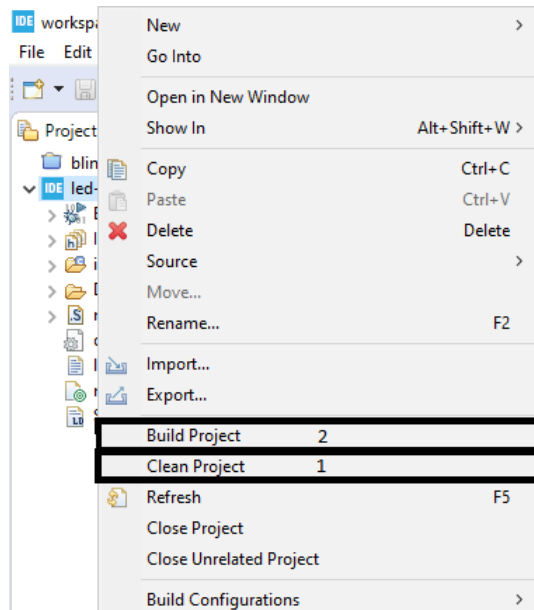


En la siguiente ventana con el botón Browse buscamos la carpeta en la que esta el proyecto y la seleccionamos, al hacerlo en el espacio blanco de abajo debe aparecer el nombre del proyecto y la dirección de la carpeta (si aparece en gris y no permite importarla puede ser debido a que ya se encuentra un proyecto con el mismo nombre en el workspace, en caso de que ya se hubiera importado antes), abilitamos con el chulo la opción de crear una copia en el workspace y despues en Finish.

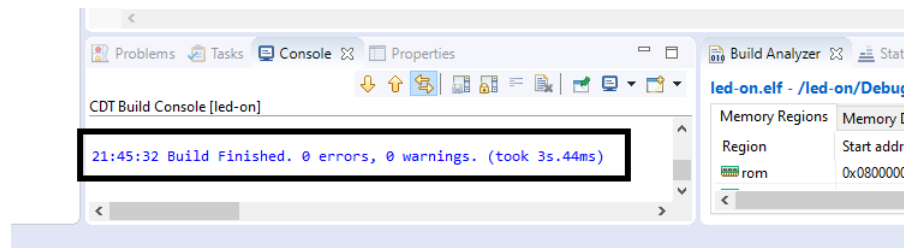


## 2.2 Compilar Proyecto

El proyecto queda importado y podemos acceder a el en la ventana Project Explorer, a continuación cargaremos el proyecto y lo subiremos a la tarjeta. Comenzamos dando click derecho en el nombre del proyecto (led-on) y seleccionamos la opción clean project, la cual limpia los residuos que quedan en el programa para evitar errores, despues volvemos y seleccionamos la opción Build project para que el IDE contruya el proyecto y podamos subirlo, durante esta construcción se pueden encontrar errores en el código los cuales nos va a señalar.

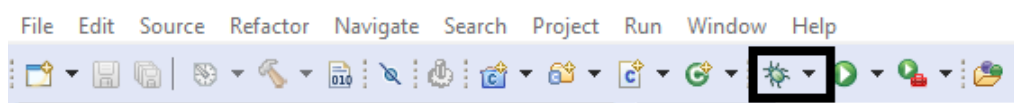


En caso de que no haya errores aparecerá el siguiente mensaje, entonces podremos comenzar con el Debug.



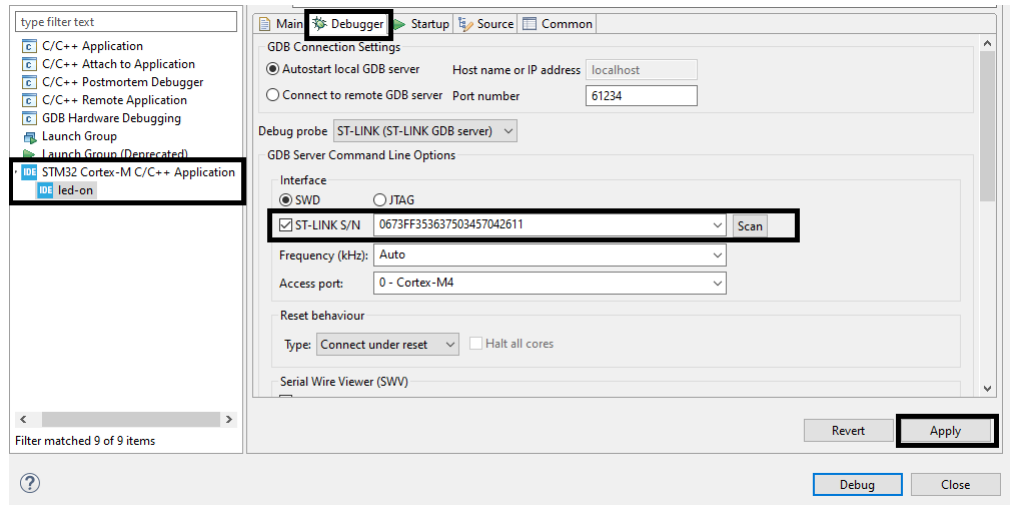
## 2.3 Debug

Para iniciar el debug presionaremos la flecha del botón que tiene un escarabajo verde que se muestra a continuación, la cual nos desplegara un menú de opciones del cual seleccionaremos Debug Configuration:

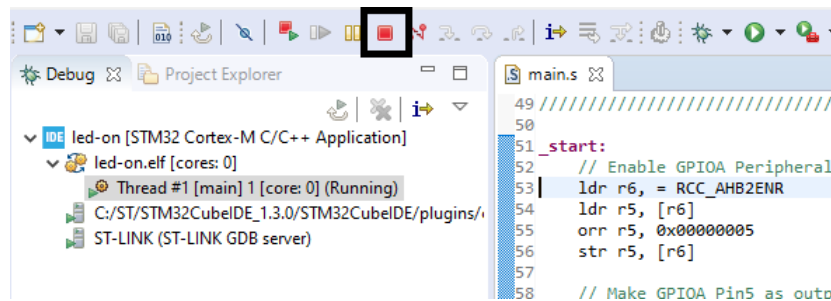


Entonces se abrirá una ventana más grande y en ella entraremos a la pestaña Debugger, donde se encuentra la configuración del Debug y un número de

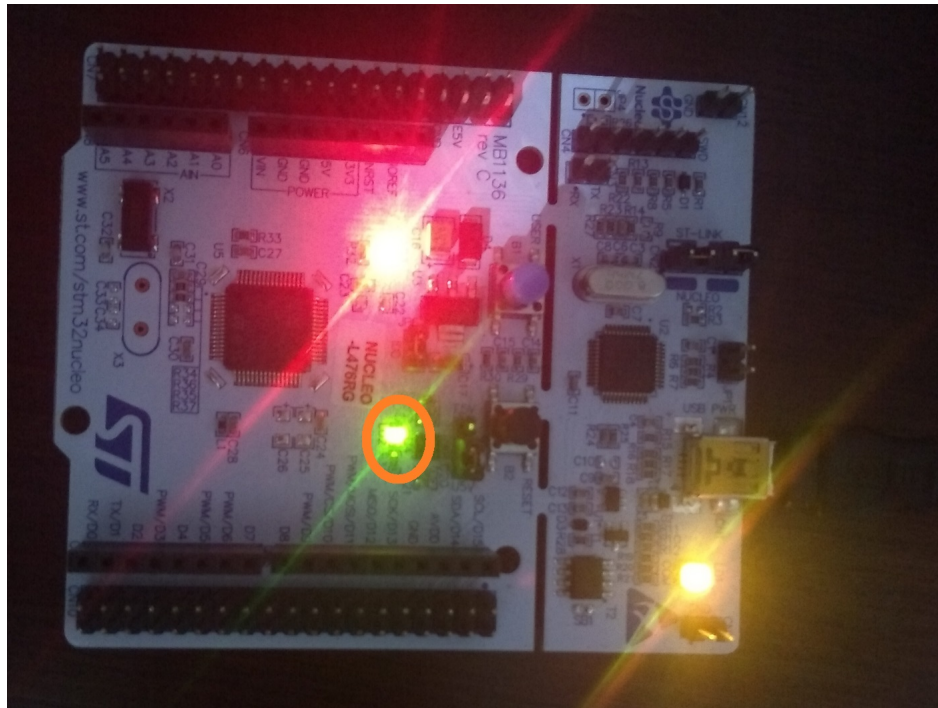
referencia para la tarjeta que estamos manejando, es posible que el espacio de este número este vacío o tenga el número de una tarjeta diferente, por lo cual habilitaremos ST-LINK S/N y presionaremos scan, para que así identifique el nuevo número de la tarjeta que tenemos conectada, volvemos a deshabilitar ST-LINK S/N, hacemos click en Apply y después en Debug.



Suponiendo que la tarjeta esté correctamente conectada al computador, el IDE cargará el programa en esta y abrirá una nueva ventana para manejar del Debug, puede ser que la ventana no se abra inmediatamente sino que primero aparezca una notificación para cambiar a modo Debugger, seleccionaremos la opción switch y entonces se deberá abrir la ventana de Debugger.



El programa se estará ejecutando en la tarjeta, debemos tener en cuenta que antes de desconectar la tarjeta es necesario terminar el Debug presionando el cuadro rojo que se señaló en la imagen anterior, para así evitar daños en la misma, la finalización del debug se reflejara en el Led 1 que se encuentra al lado de la entrada mini-USB de la tarjeta, el cual dejará de parpadear y permanecerá de un solo color.



### 3 STM32L476 MCU

El microcontrolador que se usará durante el curso es el STM32L476 con un procesador Cortex-M4, el cual determina el funcionamiento de los registros y la programación, se debe tener esto en cuenta porque los códigos hechos para otros procesadores no funcionarán (aunque también sean STM) ya que los registros difieren en funciones y valores. Para programar esta tarjeta se hará uso de dos documentos guía que se encuentran en la página de STM, los documentos son PM0214 Programming manual, STM32 Cortex-M4 MCUs and MPUs programming manual y RM0351 Reference manual, STM32L4x5 and STM32L4x6 advanced Arm-based 32-bit MCUs en los cuales encontrarán los registros del micro y comandos de ensamblador para la programación.

#### 3.1 Instrucciones en ensamblador

Conjunto de comandos que permiten cargar valores a los registros de la tarjeta, leer dichos valores y hacer operaciones aritméticas y binarias con ellos.

- **ldr**: Carga un registro con un valor que se encuentra en la memoria, se puede usar para guardar en un registro el valor que se encuentra en otro registro (`ldr r1, r2`).
- **mov**: Copia un valor en un registro (`mov r1, 0x10`) que después puede ser leído o usado en operaciones.
- **cmp**: Realiza una comparación entre el valor en un registro y un valor en otro registro o un valor numérico, el resultado de la comparación se ve reflejado en las banderas N, Z y C (`cmp r1, 0x2`).
- **b**: Se usa para saltar hacia una etiqueta sin ejecutar el código que se encuentre



entre el uso del salto y la etiqueta (b label).

- add, sub: Suman y restan el valor en un registro con el valor de otro registro o un valor numérico, el resultado se puede guardar en otro registro (add r0, r1, r2 que se ejecutaria como  $r0=r1+r2$ ).
- and, orr: Hace un and o un orr logico al valor en el registro con otro registro o un valor numérico, el resultado se guarda en el primer registro o en otro especificado, se usa para hacer enmascaramiento (orr r0, r1, r2 que se ejecutaria como  $r0=r1|r2$ ).